УДК 004.272.2:519.63

О.А. Дмитрієва, д-р техн. наук, проф.,
О.С. Бобилєва, магістрант
Донецький національний технічний університет, м. Покровськ, Україна
dmitrieva.donntu@gmail.com

Паралельна реалізація балістичного методу при розв'язанні крайових задач

Робота присвячена питанням паралельного моделювання складних динамічних об'єктів, які описуються системами звичайних диференційних рівнянь великої розмірності з крайовими умовами. Крайова задача зводиться до задачі Коші, множина розв'язань якої отримується паралельно багатостадійними методами з відстеженням умови припинення розрахунків. Обрані розв'язки використовуються для формування остаточного результату. Для керування кроком застосовуються вкладені вектори. Проведено тестові експерименти та наведено отримані результати залежності часу виконання розрахунків від кількості розрахункових точок, кількості процесорів та заданої точності.

Ключові слова: паралельне моделювання, крайова задача, крайові умови, балістичний метод, задача Коші, стадійний метод.
DOI: 10.31474/1996-1588-2018-2-27-4-10

Вступ

Однією з основних проблем при моделюванні поведінки динамічних об'єктів з зосередженими параметрами є необхідність чисельної реалізації систем звичайних диференціальних рівнянь (СЗДР) великої розмірності, заданих у формі задачі Коші [1]. Велику кількість робіт в цьому напрямку присвячено модифікації або розробці нових чисельних методів [2-3], спрямованих на ефективну реалізацію в паралельних комп'ютерних системах [4], розглянуто питання паралельного управління кроком інтегрування [5], розроблено нові підходи до оцінювання локальної та глобальної похибок [6], що сприяє посиленню об'єктивності отриманих результатів. Складність завдання при моделюванні динамічних процесів значно підсилюється, якщо мова йде про крайові задачі для СЗДР, до яких приводить велика кількість прикладних задач.

Через, як правило, високу розмірність вирішуваних завдань, на перше місце виходить проблема скорочення часу отримання результату. Враховуючи те, що сигнал має кінцеву швидкість поширення, зростання швидкодії процесорів обмежено, тому шлях, заснований на залученні для вирішення поставлених задач найсучасніших архітектурних напрацювань, не вирішує проблему повною мірою. У той же час розробка нових ефективних методів чисельного розв'язання крайової задачі або модифікація існуючих, орієнтованих на реалізацію в паралельних комп'ютерних системах, є перспективним напрямком розвитку [3,6]. З цього виникає необхідність дослідження та знаходження нових можливостей до реалізації та вдосконалення чисельних методів розв'язання крайових задач. Саме тому в даній роботі пропонуються підходи до ефективного розв'язання крайової задачі, засновані на комбінуванні паралельного пошуку розв'язків задачі Коші з застосуванням стадійних методів та балістичного методу (стрільби), на підставі якого формується остаточний результат розв'язання крайової задачі [6]

$$y''(x) + p(x)y'(x) - q(x)y(x) = g(x), x\epsilon[a, b] \quad (1)$$

$$\Gamma_a(y) = \alpha_0 y(\mathrm{a}) + \alpha_1 y'^{(\mathrm{a})} = A, |\alpha_0| + |\alpha_1| \neq 0, \quad (2)$$

$$\Gamma_b(y) = \beta_0 y(b) + \beta_1 y'^{(b)} = B, |\beta_0| + |\beta_1| \neq 0, \quad (3)$$

де $p(x), q(x), g(x)$- задані на відрізку $[a, b]$ функції,

$\alpha_0, \alpha_1, \beta_0, \beta_1, A, B$ - задані постійні.

Мета роботи полягає в розробці та дослідженні алгоритмічних методів розв'язання крайової задачі, орієнтованих на ефективну реалізацію в паралельних комп'ютерних системах.

Завдання дослідження:

– провести порівняльний аналіз сучасних чисельних методів розв'язання крайової задачі з оцінюванням можливості паралельної реалізації;

– проаналізувати проблеми розпаралелювання розв'язань крайової задачі;

– розробити паралельну модифікацію метода стрільби;

– розробити програмну систему з паралельною реалізацією методу для розв'язання крайової задачі;

– протестувати розроблену програмну систему на відомих тестових завданнях з визначенням залежностей часу реалізації від кількості залучених процесорів, розмірності розв'язуваних систем, точності розрахунків.

Огляд сучасних методів розв’язання крайової задачі

Розв’язання крайової задачі (1-3) передбачає знаходження часткового розв’язання системи звичайних диференційних рівнянь з додатковими умовами, що накладаються на значення функцій не менше ніж у двох точках відрізка $[a, b]$. Отже, крайова задача для звичайних диференціальних рівнянь ставиться для системи диференціальних рівнянь порядку не менше другого [7]. Лінійна крайова двоточкова задача полягає в знаходженні функції $y(x)$, що задовольняє лінійному звичайному диференціальному рівнянню (1) і лінійним двоточковим крайовим (граничним) умовам (2) та (3). В роботі розглядаються випадки, коли додаткові умови задачі задані на кінцях відрізка, але запропоновані підходи можуть застосовуватися і для завдань, у яких додаткові умови можуть задаватися й у внутрішніх точках відрізка (внутрішні крайові умови). Крім того, додаткові умови можуть пов'язувати між собою значення кількох функцій, похідних функцій або комбінацій функцій і похідних в одній або декількох точках відрізка, де шукається розв’язок [7].

З цього можна зробити висновок, що знаходження точного розв’язку викликає більше труднощів, ніж розв’язання задачі Коші. Звідси – підвищений інтерес і велика різноманітність наближених методів розв’язання таких завдань [6].

Аналітичні методи існують лише для вузького класу рівнянь. Зокрема, добре розвинений цей апарат для вирішення лінійних диференційних рівнянь другого порядку з постійними коефіцієнтами, які широко використовуються в дослідженні різних фізичних процесів (наприклад в теорії коливань, динаміці твердого тіла).

Наближені методи розроблялися ще задовго до появи комп'ютерів, але вони і досі не втратили свого значення. До цих методів можна віднести: методи колокацій, найменших квадратів, метод Гальоркіна і його модифікації та інші, засновані на мінімізації нев'язків рівнянь.

Сучасні чисельні методи можна розділити на 2 групи: зведення розв’язання крайової задачі до послідовності розв’язань задач Коші й безпосереднє застосування кінцево-різницевих методів [8].

За ідейною основою наближені методи можна класифікувати наступним чином:

– методи зведення до задачі Коші;
– метод кінцевих різниць;
– метод балансів або інтегро-інтерполяційний метод;
– метод колокації;
– проекційні методи (моментів, Гальоркіна);
– методи зведення до інтегральних рівнянь Фредгольма і інші [6-8].

Проблемі розпаралелювання чисельного розв’язання задачі Коші присвячено роботи [9-10], де розглядається чистий розпад часу для паралельного розв’язання еволюційних проблем. Роботи [11-12] присвячено питанням розпаралелювання крайових задач за часом. Одним з поширених методів є метод стрільби [6-8, 13-15], схему роботи якого наведено на рис. 1. Він полягає у тому, щоб звести крайову задачу до багаторазового розв’язання задачі Коші. А для вирішення задачі Коші існує множина методів, що і дозволить в кінцевому результаті знайти розв’язок. В роботі [12] наведено варіант паралельної реалізації балістичного методу, який пропонує початковий відрізок $[a, b]$ розділити на деяку кількість інтервалів, для кожного з яких обрати початкові умови, знайти розв’язання на кожному частковому інтервалі, а потім в кінцевих точках провести погодження умов. За такою реалізацією, по-перше, значно збільшується розмірність системи рівнянь, по-друге, виникає проблема вибору і погодження початкових і кінцевих умов для кожного інтервалу.

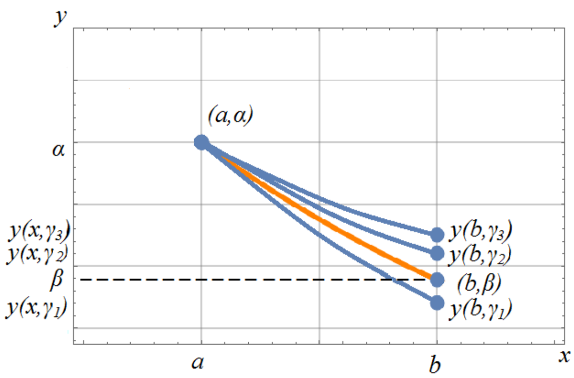

Рисунок 1 – Схема метода стрільби

Враховуючи те, що рівняння містить одне власне значення і має другий порядок, завдання вимагає трьох додаткових умов. Але третя умова задається неявно, оскільки розв’язання лінійне та однорідне і визначено з точністю до множника. Третю умову зручно задати у формі $y'(\text{a}) = \gamma$.Метод стрільби простий, внаслідок цього його можна використовувати для лінійних та нелінійних задач. Також цей метод дозволяє використовувати при чисельному інтегруванні схеми стадійного типу (або інші) високого порядку точності.

Даний метод зручно використовувати, коли мовиться про однопараметричну стрільбу, якщо ця вимога не виконується, то алгоритм значно ускладнюється, внаслідок чого стає менш надійним. У цьому випадку треба використовувати інший метод, наприклад, метод кінцевих різниць.

Також виникають труднощі тоді, коли крайова задача добре обумовлена, а відповідна їй задача Коші обумовлена погано. При цьому чисельне інтегрування задачі Коші визначає функцію з великою похибкою, що ускладнює організацію ітерацій.

В цьому випадку намагаються поставити початкові умови на іншому кінці відрізка $x = b$, тобто інтегрувати завдання Коші справа наліво. Але це можна робити, якщо число параметрів пристрілки не збільшується. Нерідко можна помітити, що стійкість поліпшується. Але, якщо зміна напрямку інтегрування не допомагає, то таку крайову задачу вирішують іншим методом [15]. Велика кількість робіт, що спрямовані на реалізацію розв'язання крайових задач, свідчить про значущість і актуальність поставленої проблеми.

Паралельна алгоритмічна реалізація балістичного методу

Реалізація крайової задачі в роботі здійснюється шляхом зведення до задачі Коші, паралельні чисельні розв'язання якої виконуються з залученням стадійних методів (4)

$$\begin{array}{c|c} c & Q \\ \hline y_{n+1} & s^t \end{array} \quad (4)$$

в яких для
управління кроком інтегрування може бути додатково використаний вкладений вектор (5).

$$\begin{array}{c|c} c & Q \\ \hline y_{n+1} & s^t \\ \hline \hat{y}_{n+1} & \hat{s}^t \end{array} \quad (5)$$

В послідовному варіанті без залучення процедури управління кроком для кожного наступного вузла обчислення проводяться за допомогою коефіцієнтів, що визначаються матрицею *Q* у стадійних точках, розташування яких визначається елементами вектора *c*. Вектор s^t містить коефіцієнти для формування результату в наступному розрахунковому вузлі. Для проведення обчислювальних експериментів і в послідовному і в паралельному варіантах реалізації використовувалося середовище розробки Microsoft Visual Studio та мова високого рівня C++.

При програмній реалізації використовувалися наступні припущення. Нехай має місце двоточкова крайова задача (1)

$$y'' = f(x, y, y') \quad (6)$$

з граничними умовами

$$y(a) = \alpha, y(b) = \beta, \quad (7)$$

яку можна вирішити за умови, що буде знайдено таке значення $y'(a)$ похідної розв'язання у початковій точці a, яке призведе до виконання умови $y(b) = \beta$. Обравши в якості початкових умов задане значення $y(a) = \alpha$ та довільне значення $y'(a)$ для рівняння (6), можна отримати розв'язок на усьому відрізку $[a, b]$, але при цьому числовий результат $y(x)$ у точці b буде відрізнятися від β.

Якщо представити знайдене розв'язання задачі Коші $y(x)$ деякою функцією від $y'(a)$:

$$y(x) = \varphi\big(y'(a)\big),$$

то можна звести початкову задачу до розв'язання нелінійного рівняння

$$F\big(y'(a)\big) = \big(\varphi\big(y'(a)\big) - \beta\big) = 0.$$

Знаходження значення $F\big(y'(a)\big)$ при кожному довільному $y'(a) = \gamma_k$ супроводжується розв'язанням задачі Коші

$$y'' = f(x, y, y'),$$
$$y(a) = \alpha, y'(a) = \gamma_k, k = 1,2, \dots \quad (8)$$

Організація ітераційного процесу дозволить отримати шуканий розв'язок $y(x)$ крайової задачі (6-7) за умови виконання наступного співвідношення

$$\lim_{k\to\infty} y\,(b, \gamma_k) = y(b) = \beta, \quad (9)$$

де $y(b, \gamma_k)$ - розв'язання задачі Коші (8) з $y'(a) = \gamma_k$.

Починаючи з деякого γ_1, що є початковим значенням для $y'(a)$, якщо $y(b, \gamma_1)$ не знаходиться близько до β, значення змінюється на нове γ_k доти розв'язання не буде знаходитися близько до β з обраною точністю.

Існують різні методи, що дозволяють підбирати за деякою схемою значення γ_k, щоб знайти розв'язання з достатньою точністю, це, наприклад, метод Ньютона, метод поділу відрізка навпіл, метод хорд. Але ці методи не підходять для розпаралелювання, бо проблема полягає в тому, що розрахункові схеми, які визначають наступне значення, повинні виконуватися тільки послідовно, тому було знайдено інший спосіб завдання γ_k.

Ідея алгоритму засновується на тому, що на початку встановлюємо ліву та праву межі симетричні відносно α, які дорівнюють $\alpha - \Delta$ та $\alpha + \Delta$. В залежності від кількості процесів n, за допомогою яких і буде виконуватися розпаралелювання, встановлюється кількість додаткових значень $\gamma_k, k = 1,2, \dots n$.

Початковий відрізок поділяється на кількість процесів, так щоб у кожного процесора був свій під відрізок. Всі підвідрізки однакового розміру. Значення γ_k дорівнює випадково обраному значенню з підвідрізку, що належить процесору. Після того, як усі процесори обрали свої значення, виконується розв'язання задачі Коші за допомогою класичного багатостадійного методу. Після цього усі отримані розв'язки з кожного процесору пересилаються до головного, де і виконується перевірка на коректність отриманого розв'язку (10) крайової задачі (6-7)

$$|y(b,\gamma_k)-\beta|\le\varepsilon, \qquad (10)$$

де ε – задана похибка.

При виконанні умови (10) вважається, що розв'язання було знайдено з заданою точністю, якщо умову не виконано, то розрізняються наступні ситуації:

– ліва та права межи завеликі, треба їх зменшити;

– ліва та права межи замалі, треба їх збільшити;

– відстань, що регламентована умовою (10), порушена, необхідно скоректувати відповідну границю.

На кожний випадок передбачена відповідна дія, що дозволяє коректувати процес обчислень. Також слід зазначити, що розв'язання усіх цих ситуацій виконується за допомогою встановлення нових границь. Для першого та другого випадку встановлюються межі того підвідрізка, який найближче знаходиться до значення β. Для останнього випадку нові межі встановлюються тільки для межових підвідрізків відповідно зменшуючи та збільшуючи їх у відповідності до отриманих результатів. Перевірку на кожному отриманому розв'язанні виконання умови (10) та коригування межі варіації додаткової початкової умови $y'(a)$ можна виконувати паралельно.

Комп'ютерні експерименти

Для тестування роботи запропонованого алгоритму була використана крайова задача [16] (11) на інтервалі $x\in[1;3]$

$$y''=\frac{1}{8}(32+2x^3-yy'), \qquad (11)$$

з граничними умовами

$$y(1)=17, y(3)=\frac{43}{3} \qquad (12)$$

та відомим точним розв'язком

$$y(x)=x^2+\frac{16}{x}. \qquad (13)$$

Реалізація запропонованого алгоритму здійснювалася за допомогою програмного інтерфейсу передачі інформації MPI (Message Passing Interface). Визначення експериментального часу виконання алгоритму проводилося з залученням можливостей бібліотеки MPI.

На рис. 2 наведено результати експериментів, які встановлюють залежність часу виконання алгоритму від кількості процесорів та розрахункових вузлів при фіксованій припустимій похибці обчислень, що складала 10^{-5}.

На рис.3 наведено результати експерименту, що відслідковував залежність часу розрахунків від заданої точності. Експеримент проводився для фіксованої кількості розрахункових точок при варіації заданої точності ε від 10^{-1} до 10^{-10} на 2 та 4 процесорах. Слід зазначити, що на 1 процесорі була змога отримати дані тільки для похибки 10^{-5}. Результати, які були отримано при проведенні такого експерименту є прогнозованими. Вони показують, що час виконання зменшується при залученні більшої кількості процесорів.

Рисунок 2 – Залежність часу виконання алгоритму від кількості точок для 2 та 4 процесорів

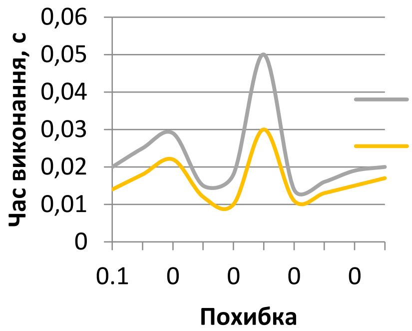

Рисунок 3 – Залежність часу виконання алгоритму на 2 та 4 процесорах від заданої похибки

На рис. 4 – 6 наведено результати чисельних і точних розв'язків тестової крайової задачі (11-12), виконаних на 4 процесорах з похибкою 10^{-5} для 20, 50 та 100 точок. Крапки показують чисельний розв'язок, отриманий багатостадійним методом, суцільна лінія – точний розв'язок (13), отриманий за допомогою теоретичної формули.

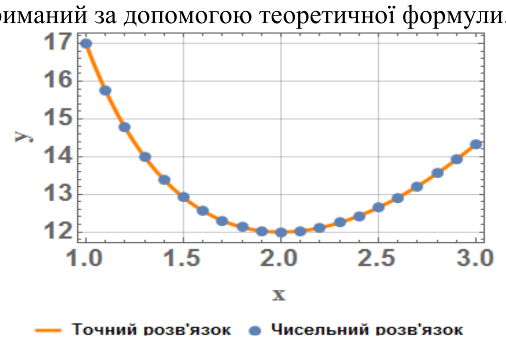

Рисунок 4 – Зображення графіку для тестової задачі на 4 процесорах для 20 точок

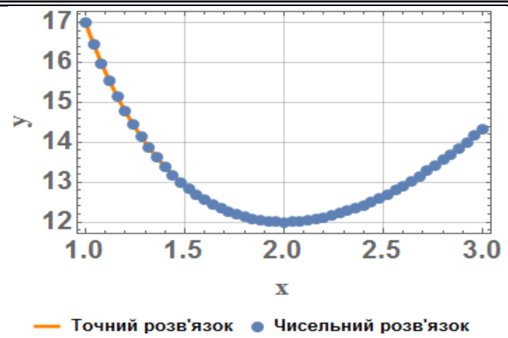

Рисунок 5 – Зображення графіку для тестової задачі на 4 процесорах для 50 точок

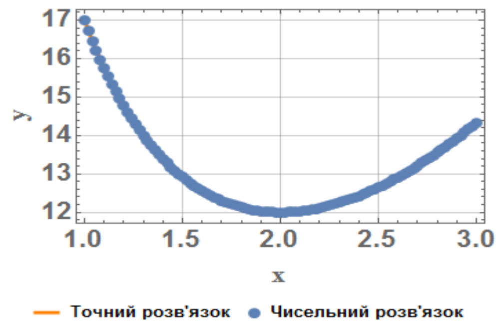

Рисунок 6 – Зображення графіку для тестової задачі на 4 процесорах для 100 точок

За отриманими результатами виявлено, що з ростом кількості процесорів, можна отримати значний виграш у часі виконання. Відсоткове відношення різниці у часі між 2 та 4 процесорами зображено на рис. 7.

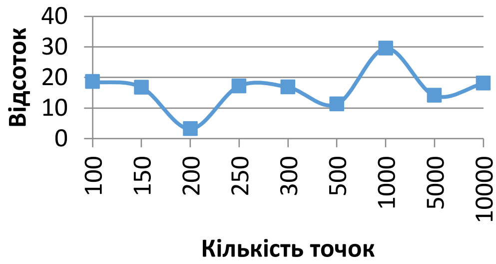

Рисунок 7 – Залежність відсоткового відношення часу виконання на 2 процесорах до 4 для різної кількості точок

Можна зробити висновок, що у середньому отримали пришвидшення виконання алгоритму на 20%.

Висновки

Робота присвячена питанням паралельного моделювання складних динамічних об'єктів, які описуються системами звичайних диференційних рівнянь великої розмірності з крайовими умовами. За результатами аналізу сучасних розробок і публікацій було обрано балістичний метод, при якому крайова задача зводиться до задачі Коші, множина розв'язань якої може бути отримана паралельно багатостадійними методами з відстеженням умови припинення розрахунків.

В роботі запропонований паралельний алгоритм балістичного методу, для розробки якого використовувалося середовище Microsoft Visual Studio та мова високого рівня C++. Реалізація запропонованого алгоритму здійснювалася за допомогою програмного інтерфейсу передачі інформації MPI. Визначення експериментального часу виконання алгоритму проводилося з залученням можливостей бібліотеки MPI.

Для перевірки результатів в якості еталонних використовувалися тестові задачі з відомими точними розв'язками. Тестування розробленої програмної системи виконано на багатоядерному комп'ютері, за результатами експериментів отримано залежності часу виконання алгоритму від кількості процесорів та розрахункових вузлів при фіксованій припустимій похибці обчислень, а також залежності часу виконання заданої похибки при фіксованій кількості процесорів та розрахункових вузлів.

Наукова новизна полягає у розробці та удосконалені паралельного алгоритму реалізації чисельних методів розв'язання крайової задачі та розв'язання задачі Коші. Було проведено дослідження розробленого методу, проаналізовано результати та відзначено закономірності.

Практична цінність полягає в розробці програмної системи, що містить в собі реалізацію чисельного методу розв'язання крайової задачі шляхом зведення її до задачі Коші та подальшої паралельної реалізації багатостадійним методом. Надалі система може застосовуватися в інших розробках, як у вигляді окремого модуля, так і в якості вбудованої з модифікацією її коду.

Список літератури

1. Хайрер Э. Решение обыкновенных дифференциальных уравнений. Нежесткие задачи / Хайрер Э., Нерсетт С., Ваннер Г. – М.: Мир, 1990. – 512с.
2. Якимов А.С. Аналитический метод решения краевых задач / А.С. Якимов. – Томск: Том. ун-та, 2011. – 199 с.
3. Дмитриева О.А. Параллельные численные методы моделирования динамических объектов: монография / О.А. Дмитриева. – Красноармейск: ГВУЗ «ДонНТУ», 2016. – 384 с.
4. Дмитриева О.А. Разработка коллокационных схем параллельного управления шагом в эволюционных уравнениях / О.А. Дмитриева, Н.Г. Гуськова // Вестник НТУ "ХПИ". Серия: Информатика и моделирование. – 2018. – № 24(1300). – С. 25 – 36.

5. Dmitrieva O. Parallel Step Control. Development of parallel algorithms of the step variation for simulation of stiff dynamic systems / O. Dmitrieva, L. Feldman. – Lambert Academic Publishing. – 2013. – 72p.
6. Фельдман Л.П. Чисельні методи в інформатиці / Л.П. Фельдман, А.І. Петренко, О.А. Дмитрієва. – К.: Видавнича група BHV, 2006. – 480 с.
7. Вержбицкий В.М. Основы численных методов: учебник для вузов/ В.М. Вержбицкий. – М.: Высш. шк., 2002. – 840 с.
8. Калиткин Н.Н. Численные методы: учеб. пособие / Н.Н. Калиткин – СПб.: БХВ-Петербург, 2011. – 592 с.
9. Nievergelt J. Parallel methods for integrating ordinary differential equations/ J. Nievergelt // Comm. ACM, 7, 1964. – 731–733 p.
10. Holsapple R. A Modified Simple Shooting Method for Solving Two-Point Boundary-Value Problems/ R. Holsapple, R. Venkataraman, D. Doman //USA, 2003 - 8 p.
11. Adam B. Shooting method in solving Boundary Value Problem / B. Adam, M. H. A. Hashim // IJRRAS . – 21 (1) . – 2014 - 23 p.
12. Deuflhard P. Recent advances in multiple shooting techniques / P. Deuflhard// Computation techniques for ordinary differential equations. – Academic Press, 1980 – p. 217-272.
13. Stoer, J. Introduction to Numerical Analysis /J. Stoer, R. Bulirsch// New York: Springer, 2002 – 609 p.
14. Ascher U. M. Numerical Solution of Boundary Value Problems for Ordinary Differential Equations/U.M. Ascher, R.M. Mattheij R.D. Russell// Philadelphia: SIAM, 1995 – 593 p.
15. Keller H.B. Numerical Solution for Two-Point Boundary-Value Problems, New York: Dover Publications Inc., 1992 – 59 p.
16. Ha S.N. A Nonlinear Shooting Method for Two-Point Boundary Value Problems /S. N. Ha // Computers and Mathematics with Applications 42, Oxford: Pergamon Press .- 2001. – p. 1411 - 1420

References

1. Hairer, E., etc., (1990), "Solving Ordinary Differential Equations I. Nonstiff Problems" [Reshenie obiknovennih differencialnih uravnenii. Negestkie zadachi], M.: Mir, 512 p.
2. Jakimov, A., (2011), "Analytical method for solving boundary value problems" [Analiticheskii metod reshenia kraevih zadach], Tomsk: Tom. un-ty, 199 p.
3. Dmitrieva, O. (2016), "Parallel numerical methods for modeling dynamic objects", [Parallel'nye chislennye metody modelirovanija dinamicheskih ob"ektov], Pokrovsk: "DonNTU", 384 p.
4. Dmitrieva O., Guskova N., (2018), "Development of collocation schemes for parallel control of step in evolutionary equations" [Razrabotka kollokacionnih shem parallelnogo upravlenia shagom v evolucionnih uravneniyah], Vestnik NTU "HPI". Seriya: Informatika i modelirovanie . – Harkiv: NTU: "HPI", № 24(1300), – p. 25 – 36.
5. Dmitrieva O., Feldman L., (2013), "Parallel Step Control. Development of parallel algorithms of the step variation for simulation of stiff dynamic systems", Lambert Academic Publishing, 72p.
6. Feldman L., etc., (2006), "Numerical methods in computer science" [Chiselni metodi v informatyci], K.: Vidavnycha grupa BHV, p. 480.
7. Vergbickii, V., (2002), "The basics of numerical methods" [Osnovi chislennih metodov], M.: Vish. sh., 840 p.
8. Kalitkin, N., (2011), "Numerical methods" [Chislennie metodi], SPb.:BHV-Peterburg, 592 p.
9. Nievergelt, J., (1964), "Parallel methods for integrating ordinary differential equations", Comm. ACM, 7, 731–733 p.
10. Holsapple, R., etc., (2003), "A Modified Simple Shooting Method for Solving Two-Point Boundary-Value Problems", USA, 8 p.
11. Adam, B., Hashim M., (2014), "Shooting method in solving Boundary Value Problem", IJRRAS 21 (1), 23 p.
12. Deuflhard P., (1980), "Recent advances in multiple shooting techniques", Computation techniques for ordinary differential equations, Academic Press, 217-272 p.
13. Stoer, J., Burlisch, R., (2002), "Introduction to Numerical Analysis", New York: Springer, 609 p.
14. Ascher, U., etc., (1995), "Numerical Solution of Boundary Value Problems for Ordinary Differential Equations", Philadelphia: SIAM, 593 p.
15. Keller, H., (1992) "Numerical Solution for Two-Point Boundary-Value Problems", New York: Dover Publications Inc., 59 p.
16. Ha S., (2001), "A Nonlinear Shooting Method for Two-Point Boundary Value Problems" /S. N. Ha // Computers and Mathematics with Applications 42, Oxford: Pergamon Press, p. 1411 – 1420.

Надійшла до редакції 10.12.2018

О.А. ДМИТРИЕВА, А.С. БОБЫЛЕВА
Донецкий национальный технический университет, г. Покровск, Украина

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ БАЛЛИСТИЧЕСКОГО МЕТОДА ПРИ РЕШЕНИИ КРАЕВЫХ ЗАДАЧ

Работа посвящена вопросам параллельного моделирования сложных динамических объектов, которые описываются системами обыкновенных дифференциальных уравнений большой размерности с краевыми условиями. Краевая задача сводится к задаче Коши, множество решений которой получается параллельно многостадийными методами с отслеживанием условия прекращения расчетов. Избранные решения используются для формирования окончательного результата. Для управления шагом применяются вложенные векторы. Проведены тестовые эксперименты и приведены полученные результаты зависимости времени выполнения расчетов от количества расчетных точек, количества процессоров и заданной точности.

Ключевые слова: параллельное программирование, краевая задача, краевые условия, баллистический метод, задача Коши, стадийный метод.

O. DMYTRIYEVA, **O. BOBYLIEVA**
Donetsk National Technical University, Pokrovsk, Ukraine

PARALLEL IMPLEMENTATION OF BALLISTIC METHOD FOR SOLVING THE BOUNDARY VALUE PROBLEMS

The work is devoted to the issues of parallel modeling of complex dynamic objects, which are described by systems of ordinary differential equations of large dimension with boundary conditions. The boundary-value problem is reduced to the Cauchy problem, the set of solutions of which is obtained in parallel by multistage methods with tracking of the condition for termination of calculations. Selected solutions are used to form the final result. Nested vectors are used to control the step. Test experiments were carried out and the obtained results were given depending on the time taken to complete the calculations on the number of calculation points, the number of processors and the specified accuracy.

Due to, as a rule, the high dimension of the problems being solved, the problem of reducing the time for obtaining the result comes to the first place. Considering that the signal has a finite speed of propagation, the growth of processor speed is limited, so the path based on attracting modern architectural developments to solve the tasks set does not fully solve the problem. At the same time, the development of new efficient methods for the numerical solution of a regional problem or the modification of existing, implementation-oriented parallel computer systems is a promising direction of development. This necessitates the study and finding of new opportunities for the implementation and finding of numerical methods for solving boundary value problems. That is why this paper proposes approaches to effectively solving a boundary value problem based on combining a parallel search for solutions to the Cauchy problem using stepwise methods and a ballistic method (shooting), on the basis of which the final result of solving the boundary value problem is formed.

The paper proposes a parallel algorithm for the ballistic method, for which the Microsoft Visual Studio environment and high level C ++ language were used. The implementation of the proposed algorithm was carried out with the help of the standardized and portable message-passing standard MPI. Determination of the experimental time of execution of the algorithm was carried out with the possible use of the MPI library.

To test the results as benchmarks, test tasks with known exact solutions were used. The testing of the developed program system was performed on a multi-core computer in carrying out numerical experiments.

Based on the results, it is found that with increasing number of processors, you can get a significant gain in the implementation time. One can conclude that the average speed of the algorithm was 20%. According to the results of the experiments, the dependences of the execution time of the algorithm on the number of processors and computational nodes were obtained for a fixed allowable calculation error, as well as the dependence of the execution time of a given error for a fixed number of processors and computational nodes.

Keywords: parallel programming, boundary value problem, boundary conditions, ballistic method, Cauchy problem, stage method.