

вычислений позволяет выполнить вычислительный процесс в 1, 2, 3 раза быстрее по сравнению с однопоточным последовательным процессом расчета благодаря почти 100% загрузке всех ядер компьютера. Применение технологии OpenMP целесообразно для решения задач на сетках со значительным числом узлов. Алгоритм решения можно применить для решения трехмерных задач.

Список использованной литературы:

1. Тихонов А. Н. Уравнения математической физики / А. Н. Тихонов, А. А. Самарский. — М. : Наука, 1977. — 735 с.
2. Самарский А. А. Методы решения сеточных уравнений / А. А. Самарский, Е. С. Николаев. — М. : Наука, 1978. — 590 с.
3. К. С. Гэтлин, П. Айсенс. OpenMP и C++. — Режим доступа: <http://www.microsoft.com/Rus/MSDN/Magazine/2005/10/OpenMP.mspx>.

Results of application of OpenMP technology to parallelize numerical solving of heterogeneous 2D heat equation using alternating directions implicit (ADI) method are reviewed.

Key words: partial differential equations, OpenMP technology, alternating directions implicit method.

Отримано 25.09.2010

УДК 519.64

Д. А. Верлань, студент

КНУ ім. Тараса Шевченка, факультет кібернетики, м. Київ

АЛГОРИТМИ РЕАЛІЗАЦІЇ МЕТОДУ ВИРОДЖЕНИХ ЯДЕР ПРИ РОЗВ'ЯЗАННІ ІНТЕГРАЛЬНИХ РІВНЯНЬ ВОЛЬТЕРРИ

У статті розглянуто чисельний алгоритм розв'язку інтегрального рівняння Вольтерри II роду з ядром довільного вигляду. При цьому довільне ядро попередньо приводиться до виродженого вигляду за допомогою ітераційно-варіаційного методу. Шляхом обчислювальних експериментів оцінено даний метод.

Ключові слова: інтегральне рівняння Вольтерри, вироджене ядро, апроксимація, функція двох змінних.

Вступ. Інтегральні рівняння Вольтерри II роду

$$y(t) + \int_a^t K(t, \tau) y(\tau) d\tau = f(t), \quad (1)$$

де $y(t)$ — шукана функція, ядро $K(t, \tau)$ та $f(t)$ — відомі функції, отримують все більше розповсюдження в якості динамічних моделей

об'єктів різноманітної фізичної природи і призначення [1]. Важливу роль дані рівняння відіграють при дослідженні систем управління об'єктами зі зосередженими та розподіленими параметрами.

При цьому особливе значення мають задачі формування та реалізації моделей, які призначені для синтезу керуючих систем в тих випадках, коли керуюча частина системи (регулятор) будується у вигляді програмних додатків, котрі організовані за заданою динамічною моделлю. В даному випадку до алгоритмів та програм реалізації динамічних моделей пред'являються високі вимоги щодо відносної швидкодії, тобто програми, які розроблюються, мають забезпечувати функціонування системи в реальному часі. Це свідчить про те, що суттєве значення отримує вибір чисельного метода для розв'язання відповідних рівнянь.

Обчислювальні схеми. Відомо [2], що найбільш розповсюджені квадратурні методи розв'язання інтегрального рівняння (1), маючи ряд переваг (простота алгоритму, висока стійкість розрахункового процесу), у випадку ядер довільного вигляду задовольняють властивості, яка полягає в рості кількості операцій на кожному новому кроці розрахунків. Квадратурний алгоритм розв'язання такого рівняння [1] має вигляд:

$$\left. \begin{aligned} y_1 &= f_1, \\ y_2 &= \frac{f_2 + \frac{h_2}{2} K(t_2, t_1) \tilde{y}(t_1)}{1 + \frac{h_2}{2} K(t_2, t_2)}, \\ y_i &= \frac{f_i + \frac{h_i}{2} K(t_i, t_1) \tilde{y}(t_1) + \sum_{j=2}^{i-1} \left(\frac{t_{j+1} - t_{j-1}}{2} \right) K(t_i, t_j) \tilde{y}(t_j)}{1 + \frac{h_i}{2} K(t_i, t_i)}, \\ & \quad i = \overline{3, n}, \end{aligned} \right\} \quad (2)$$

де $\frac{h_i}{2} K(t_i, t_i) \neq 1$. Для постійного кроку $h_i = h = const$ при застосуванні формули трапеції має місце наступний алгоритм:

$$\left. \begin{aligned} y_1 &= f_1, \\ y_i &= \frac{f_i + h \sum_{j=1}^{i-1} A_j K(t_i, t_j) \tilde{y}(t_j)}{1 + \frac{h}{2} K(t_i, t_i)}, \quad i = \overline{2, n}, \end{aligned} \right\} \quad (3)$$

де $l = 2, 3, \dots, n$; $t_l = a + h(i-1)$; $A_j = \begin{cases} 0,5 & \text{при } j = 1, \\ 1 & \text{при } j > 1. \end{cases}$

Ефективним підходом до подолання даного ускладнення може бути застосування методу вироджених ядер, що дозволяє побудувати алгоритм розв'язання, в якому кількість операцій на кожному кроці не залежить від номеру кроку.

Дійсно, інтегральне рівняння (1) з виродженим ядром

$$K(t, \tau) = \sum_{i=1}^N \alpha_i(t) \beta_i(\tau), \quad (4)$$

де $\alpha_i(t)$ та $\beta_i(\tau)$ — задані функції, приймає вигляд

$$y(t) + \sum_{i=1}^N \alpha_i(t) \int_a^t \beta_i(\tau) y(\tau) = f(t). \quad (5)$$

Переведемо рівняння (5) в дискретну форму та отримаємо

$$y(t_j) + \sum_{i=1}^N \alpha_i(t_j) \int_a^t \beta_i(\tau) y(\tau) = f(t_j), \quad j = \overline{1, n}, \quad (6)$$

що дозволяє застосувати для розв'язання інтегрального рівняння якусь із квадратурних формул та отримати наступний рекурентний вираз:

$$\left. \begin{aligned} y_1 &= f_1 \\ y_l &= \frac{1}{1 - \sum_{i=1}^N \alpha_i(t_l) \beta_i(t_l)} \left(f_l + \sum_{i=1}^N \alpha_i(t_l) \sum_{j=1}^{l-1} \beta_i(t_j) A_j y_j \right), \quad l = \overline{2, n}, \end{aligned} \right\} \quad (7)$$

з якого видно, що кількість обчислень на кожному кроці залишається незмінною, оскільки для обчислення кожної з N сум $\sum_{j=1}^{l-1} \beta_i(t_j) A_j y_j$ можна скористатися обчисленнями на попередньому кроці, тобто $\sum_{j=1}^{l-1} \beta_i(t_j) A_j y_j = \sum_{j=1}^{l-2} \beta_i(t_j) A_j y_j + \beta_i(t_{j-1}) A_{j-1} y_{j-1}$, $l = \overline{3, n}$. Саме тому доцільно використовувати при розв'язанні інтегральних рівнянь особливість роздільності ядер.

Відмітимо, що застосування загальної формули з постійним кроком $h = h_i = \text{const}$ дозволяє також отримати розрахункові формули у вигляді:

$$\left. \begin{aligned} \tilde{y}_1(a) &= f_1(a), \\ y_l &= \frac{1}{1 - \frac{h}{2} \sum_{i=1}^N \alpha_i(t_l) \beta_i(t_l)} \left(f_l + h \sum_{i=1}^N \alpha_i(t_l) \sum_{j=1}^{l-1} \beta_i(t_j) A_j \tilde{y}_j \right), \quad l = \overline{2, n}, \end{aligned} \right\} \quad (8)$$

де $l = 2, 3, \dots, n$; $t_l = a + h(i-1)$; $A_j = \begin{cases} 0,5 & \text{при } j = 1, \\ 1 & \text{при } j > 1. \end{cases}$

Таким чином, для загального випадку розв'язання рівняння (1), коли ядро має довільний вигляд, застосування методу вироджених ядер потребує розв'язання задачі попередньої апроксимації ядра виродженим. При цьому метод вироджених ядер має містити в собі додаткові обчислення в вигляді операцій реалізації методу апроксимації ядра. Ця обставина не є перешкодою для застосування методу вироджених ядер як в дослідницьких задачах, не зв'язаних з вимогою до швидкості алгоритмів, так і при синтезі систем керування, оскільки ці задачі розв'язуються на етапі проектування системи.

В якості методу апроксимації ядра рівняння (1) з метою отримання виродженого ядра вигляду (3) доцільно використовувати варіаційно-ітераційний алгоритм, відповідно до [2].

Апроксимація ядра. Схема алгоритму полягає в тому, що при заданому ядрі $K(t, \tau)$ задача апроксимації зводиться до розв'язування задачі мінімізації функціоналу

$$\Phi = \iint_{a a}^{b b} \left[K(t, \tau) - \sum_{i=1}^N \alpha_i(t) \beta_i(\tau) \right]^2 dt d\tau, \quad (9)$$

де $\alpha_i(t)$, $\beta_i(\tau)$ — шукані функції.

Спочатку знаходиться перше наближення заданої функції у вигляді одного доданку $\alpha_1(t) \beta_1(\tau)$. Цей перший доданок формуємо наступним чином: задаємо $\beta_1^{(0)}(\tau)$ — початкове наближення функції $\beta_1(\tau)$ та з умов мінімуму функціонала

$$\Phi_1^{(0,0)} = \iint_{a a}^{b b} \left[K(t, \tau) - \alpha_1^{(0)}(t) \beta_1^{(0)}(\tau) \right]^2 dt d\tau \quad (10)$$

знаходимо $\alpha_1^{(0)}(t)$ — початкове наближення $\alpha_1(t)$.

Відомим варіаційним методом знаходимо, що для виконання умови мінімуму функціонала, визначеного виразом (8), необхідно, щоб

$$\alpha_1^{(0)}(t) = \frac{\int_a^b K(t, \tau) \beta_1^{(0)}(\tau) d\tau}{\int_a^b (\beta_1^{(0)}(\tau))^2 d\tau}. \quad (11)$$

Потім з умови мінімуму функціоналу

$$\Phi_1^{(0,1)} = \iint_{a a}^{b b} \left[K(t, \tau) - \alpha_1^{(0)}(t) \beta_1^{(1)}(\tau) \right]^2 dt d\tau, \quad (12)$$

відшуковуємо $\beta_1^{(1)}(\tau)$ — перше наближення функції $\beta_1(\tau)$:

$$\beta_1^{(1)}(\tau) = \frac{\int_a^b K(t, \tau) \alpha_1^{(0)}(t) dt}{\int_a^b (\alpha_1^{(0)}(t))^2 dt} \quad (13)$$

і так далі. Процес припиняється, як тільки

$$\left. \begin{aligned} \int_a^b (\beta_1^{(n+1)}(\tau) - \beta_1^{(n)}(\tau))^2 \leq \varepsilon, \\ \int_a^b (\alpha_1^{(n+1)}(t) - \alpha_1^{(n)}(t))^2 \leq \varepsilon, \end{aligned} \right\} \quad (14)$$

де ε — показник заданої точності обчислення функцій $\beta_1(\tau)$ та $\alpha_1(t)$.

Якщо досягнута задана точність визначення $\alpha_1(t)$ та $\beta_1(\tau)$, то аналогічним чином наближаємо функцію $K(t, \tau) - \alpha_1(t)\beta_1(\tau)$ добутком $\alpha_2(t)\beta_2(\tau)$, виходячи з того ж критерію оптимальності (7), і т.д. поки не буде виконана умова

$$\int_a^b \int_a^b \left[K(t, \tau) - \sum_{i=1}^N \alpha_i(t)\beta_i(\tau) \right]^2 dt d\tau < \varepsilon_{анп}, \quad (15)$$

де $\varepsilon_{анп}$ — задана точність апроксимації.

Таким чином, отримуємо ряд $\sum_{i=1}^N \alpha_i(t)\beta_i(\tau)$, який з заданою точністю апроксимує вихідну функцію $K(t, \tau)$.

Обчислювальний експеримент. Розглянемо розв'язок рівняння

$$y(t) + \int_0^t \sin(t\tau) y(\tau) d\tau = \frac{\cos(t^2) - 1}{t} + 1, \quad (16)$$

наведеним вище методом за допомогою програм, розроблених в системі Matlab. На першому (попередньому) етапі виконується апроксимація ядра $\sin(t\tau)$ виродженням за допомогою алгоритму (9)—(15).

При виконанні умови $\varepsilon_{анп} = 0.1^5$ отримуємо в чисельному вигляді необхідну кількість апроксимуючих функцій $\alpha_i(t)$, $\beta_i(\tau)$, що забезпечує можливість реалізації алгоритму (8). В табл. 1 наведені значення отриманого розв'язку рівняння (16) для вибірових значень аргументу t на інтервалі $[0, 1]$ з кроком $h = 0.1^3$, l — номер точки розбиття. На отримання такого розв'язку знадобилося 0.0565 секунди, максимальна абсолютна похибка не перевищує $\Delta = 0.1^7$.

Таблиця 1

l	1	100	200
$y(t_i)$	1	0.99999999998884	0.999999999984276
l	300	400	500
$y(t_i)$	0.99999999899046	0.99999999578345	0.99999998696389
l	600	700	800
$y(t_i)$	0.99999996722903	0.999999992847085	0.999999985919471
l	900	1000	
$y(t_i)$	0.99999974425213	0.99999956470761	

Для порівняння був отриманий розв'язок рівняння (16) за допомогою алгоритму (3), відображений у табл. 2. Розв'язок отримано за 5.3873 секунд з максимальною похибкою $\Delta = 0.1^3$, l – номер точки розбиття.

Таблиця 2

l	1	100	200
$y(t_i)$	1	0.999995100766807	0.999980243696106
l	300	400	500
$y(t_i)$	0.999955658051147	0.999922000653228	0.999879752134584
l	600	700	800
$y(t_i)$	0.999833975943129	0.999785646638863	0.999740608692681
l	900	1000	
$y(t_i)$	0.999704778742659	0.999683975421228	

Висновок. Отриманий алгоритм розв'язання інтегрального рівняння Вольтерри II роду (1) на основі реалізації методу вироджених ядер має суттєві часові переваги перед традиційним квадратурним алгоритмом і забезпечує високу точність отриманого розв'язку. Тим самим забезпечується можливість використання інтегральних динамічних моделей в системах реального часу.

Список використаних джерел:

1. Верлань А. Ф. Интегральные уравнения: методы, алгоритмы, программы. Справочное пособие / А. Ф. Верлань, В. С. Сизиков. — К. : Наук. думка, 1986. — 544 с.
2. Верлань Д. А. Ітераційні алгоритми апроксимації функцій двох змінних / Д. А. Верлань // Математичне та комп'ютерне моделювання. Серія: Технічні науки. Зб. наук. праць. — Кам'янець-Подільський: Кам'янець-Подільський національний університет, 2009. — Вип. 2. — С. 24—32.

The article deals with the numerical algorithm of solution of integral equations Volterra second kind with kernel of arbitrary form. This arbitrary kernel previously reduced to the degenerate-looking iterative-variational method. Through experiments evaluated this method.

Key words: *Volterra integral equation, degenerate kernel, approximation, function of two variables, quadratic residual.*

Отримано 06.10. 2010