References

[1] Revised 1996 IPCC Guidelines for National Greenhouse Gas Inventories: Reporting instructions.- Vol. 1.- IPCC, 1996.

[2] Інформаційні технології інвентаризації парникових газів та прогнозування вуглецевого балансу України / Р.А.Бунь, М.І.Густі, В.С.Дачук та ін.; За ред. Р.А.Буня.- Львів: УАД, 2004.- 376 с.

[3] Гамаль Х. Геоінформаційний підхід до інвентаризації парникових газів на Львівщині / Х. Гамаль, Н. Терлецька // Комп'ютерні науки та інженерія : Матеріали 1-ї Міжнар. конф. молодих науковців (CSE-2006). – Львів : НУ «ЛП», 2006. – С. 88–90

[4] Бунь Р. А. Математичні моделі для просторової інвентаризації парникових газів в енергетичній галузі Львівщини / Р. А. Бунь, Х. В. Гамаль // Моделювання та інформаційні технології. − Вип. 40.− 2007.− С. 167−175.

[5] Просторова база даних України масштабу 1:500 000. Версія 1.5 / Товариство з обмеженою відповідальністю "Інтелектуальні Системи ГЕО" (ТОВ ІСГЕО).

[6] Офіційний Інтернет сайт Верховної Ради України. Режим доступу: http://zakon.rada.gov.ua/

[7] Паливно−енергетичні ресурси Волинської області та їх використання : Статистичний збірник. − – Луцьк: Головне управління статистики у Волинській обл., 2008. –76 с

[8] Статистичний щорічник Волинської області за 2006 рік : Статистичний збірник. – Луцьк : Головне управління статистики у Волинській обл., 2008. – 540с.

Інкрементні ітераційні методи для розв'язання СЛАР

Гомозов Олег

Кафедра прикладної математики та інформатики, Донецький національний технічний університет, УКРАЇНА, м.Донецьк, вул. Артема, 58, E-mail: blackswanny@gmail.com

Abstract – This paper describes special class of incremental iterative methods to solve linear equations. Also there described their advantages compared to other iterative methods and analysis of perspecitve FPGA based implementation of these methods.

Ключові слова – linear equations, solver, increment, iterative methods, matrix.

I. Вступ

У багатьох наукових сферах використовуються системи лінійних алгебраїчних рівнянь (СЛАР). Ними описуються складні математичні моделі, фізичні процеси, динамічні системи і інші аналітичні задачі. Проте часто необхідні системи можуть досягати великих порядків. При цьому виникає проблема швидкості таких задач і коректності отриманих результатів. Обидві проблеми можуть розв`язати обчислювальні машини. У зв'язку з цим були створені спеціальні алгоритми, засновані на класичних ітераційних методах розв`язання СЛАР.

II. Опис методів

Ітераційні методи засновані на тому, що розв`язання СЛАР вигляду:

$$A \cdot \vec{x} = \vec{b}, \qquad (1)$$

де A - матриця коефіцієнтів, b - вектор вільних членів, x - вектор невідомих, відбувається шляхом послідовних наближень $\vec{x}^{(n)}$ при $n \to \infty$, де n - номер ітерації, умовою закінчення часто є виконання нерівності $\left\| x^{(n)} - x^{(n-1)} \right\| < \varepsilon$, де $\varepsilon>0$ –погрішність обчислення. Найвідоміші методи Якобі, Зейделя, Ланцоша, верхніх релаксацій.

Недоліком ітераційних методів є лише їх можлива незбіжність, але їх перевагами перед прямими методами є:

- менша кількість обчислень за рахунок раціонального використовування структури матриці і відсутності надмірності інформації;
- велика швидкість розв`язання;
- ефективність реалізації для обчислювальних машин, зокрема, менші апаратні витрати.

На основі цих методів були створені алгоритми, орієнтовані на апаратну реалізацію. У цих методах арифметичні операції перетворяться в зручний вигляд для використовування в цифрових обчислювальних пристроях. При цьому використовуються проміжні змінні, такі як, нев'язність, що визначає значення і напрям відхилення поточного значення невідомої від істинного. У цих алгоритмах розв`язання СЛАР (1) представляється у вигляді інтеграції еквівалентної системи лінійних диференціальних рівнянь (СЛДР) [1], тоді розв`язання визначається за формулою:

$$\frac{dx_i}{dt} = b_i - \sum_{i,j=1}^{n} a_{ij} x_i,\ i, j = 1,2 \ldots n,\ n - \textit{порядок СЛАР} \qquad (2)$$

Позначимо ліву частину системи через нев'язність $\varepsilon_i(x)$, тоді $\varepsilon_i(x) = b_i - \sum_{i,j=1}^{n} a_{ij} x_i$. Розв`язання СЛДР з початковими умовами $\vec{x}_0$ виконується методом Ейлера, тоді розв`язання обчислюється за наступними формулами:

$$x_i^k = x_i^{k-1} + \Delta x_i^k; \quad \Delta x_i^k = H \cdot \varepsilon_i^{k-1}(x);$$
$$x_i^{(0)} = x_{0i},\ i = 1,2...n, \tag{3}$$

де $k -$ номер ітерації, $H -$ крок інтегрування.

Оскільки в обчислювальних пристроях використовується двійкова система числення, для спрощення обчислень крок і приріст невідомого можуть бути виражені через неї:

$H = 2^{-m}, \Delta x = \pm 2^{-m}$, де $m \le p, p$ - розрядність даних.

При підстановці в (3) одержимо основний інкрементний алгоритм [1,2] розв`язання СЛАР:

$$x_i^{k+1} = x_i^k + \Delta x_i^{k+1}; \quad \Delta x_i^{k+1} = 2^{-p} \cdot sign(\varepsilon_i^k),$$
$$\varepsilon_i^{k+1} = \varepsilon_i^k + \Delta\varepsilon_i^{k+1}; \tag{4}$$
$$\Delta\varepsilon_i^{k+1} = -\sum_{j=1}^{n} a_{ij}\Delta x_i^{k+1},\ x_i^0 = 0, \varepsilon_i^0 = b_i,$$

Для реалізації алгоритму можуть бути використані реверсивні лічильники, суматори, схеми зсуву і виділення знаку. Обчислення приросту невідомих можна проводити іншим способом:

$\Delta x^{k+1} = 2^{-\frac{p}{2S}} \cdot sign(\varepsilon^k)$, де S - величина, що визначає вагу приросту.

Умовою припинення обчислень служить нерівність:

$\sum_{i=1}^{n} (\varepsilon_i^k)^2 \le E = 2^{-\frac{p}{2S}} \cdot \delta$, де δ – погрішність.

Величина S може визначатися різними способами, наприклад:

$\left|\Delta x^{k+1}\right| = \left|\frac{\Delta x^k}{2}\right|$, тоді формула (2) прийме вигляд:

$$\Delta x^{k+1} = 2^{-r} \cdot sign(\varepsilon^k), \text{де } r = 1,2...p-1, p \tag{5}$$

Розв`язання можливе знайти за p ітерацій, але при цьому структура обчислювального пристрою ускладнюється через появу складних логічних операцій.

Найпростіший з погляду апаратної реалізації спосіб визначення приросту вектора невідомих приведений нижче:

$$\Delta x^{k+1} = 2^{-z} \cdot U_i \cdot sign(\varepsilon_i^k); \quad z = \min_{i=1..n} m_i;$$
$$2^{-m_i} \le \varepsilon_i^k < 2^{-m_i+1}; \quad U_i = \begin{cases} 1, \varepsilon_i^k \ge 2^{-z} \\ 0, \varepsilon_i^k < 2^{-z} \end{cases} \tag{6}$$

У цьому алгоритмі є додаткові операції, але він має кращу збіжність. Можлива також його модифікація при використовуванні всіх старших розрядів нев'язності ε_i^k:

$$\Delta x^{k+1} = 2^{-m_i} \cdot sign\ (\varepsilon_i^k),$$
$$2^{-m_i} \le \varepsilon_i^k < 2^{-m_i+1}, \tag{7}$$

схема реалізації стає простіше, але кількість ітерацій декілька зростає в порівнянні з (6).

Вищенаведені алгоритми (4-7) розрізняються за способом обчислення приросту, що визначається кількість ітерацій і відповідно швидкість знаходження розв`язання, а також складність обчислювального пристрою[2].

При однаковій погрішності алгоритми (4) і (5) мають дуже багато ітерацій і є достатньо повільними. Оптимальними є способи (6) і (7), їх основні переваги:

- висока швидкість розв`язання через невелику кількість ітерацій;
- ефективність апаратної реалізації;
- відсутність обмежень, застосування для багатьох класів задач.

III. Аналіз FPGA реалізації

Одним з перспективних напрямків для реалізації цих алгоритмів є інтегральні схеми FPGA. Вони складаються з різних модулів, основними з них є програмовані логічні і блоки комутації.

Мікросхеми FPGA можуть бути запрограмовані в різній конфігурації, так що вони дозволяють проводити розпаралелювання процесу обчислення у зв'язку з наявністю окремих логічних блоків, які можна об'єднати шинами в будь-якій послідовності, що ще більше прискорить роботу алгоритмів, приведених вище. Крім того можлива організація будь-якого зовнішнього інтерфейсу, що дозволить включити пристрої, розв`язуючі СЛАР, в кластери, де виникає проблема розв`язання відповідних прикладних задач[3]. При цьому FPGA мають перевагу в ціні перед універсальними процесорами і спеціалізованими мікроконтролерами.

У сфері сучасних інформаційних технологій FPGA швидко розвиваються у зв'язку з актуальністю цих мікросхем, постійно з'являються нові можливості, зокрема в середовищах програмування за допомогою мов опису апаратури (HDL) і прикладних пакетах для моделювання і тестування, таких як Active-HDL, що дозволяють створювати програми на мовах VHDL і Verilog.

Ці мови є міжнародним стандартом і використовуються, перш за все, для проектування надшвидких ВІС і мають багато переваг по програмуванню FPGA.

Висновок

Описані вище засоби визначають переваги при проектуванні цільових пристроїв на FPGA, а також їх тестування і моделювання, що значно поліпшує процес створення пристроїв розв`язання СЛАР на базі цих мікросхем.

[1] Malinovsky B.N., Boun V.P., Kozlov L.G., "Algorithms to solve linear system equations, oriented on structural realization," "Operating systems and machines". vol. №5, pp.79-84, 1977.

[2] Boun V.P., Kozlov L.G., Malinovsky B.N., Tretyakov S.I., "Devices for solving linear system equations," Cert. № 543943 – BI, 1977.

[3] Maxfield K., "Design of PLD. Architecture, tools, methods," Dodeka – XXI, pp. 408, 2007.