computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

СЕКЦІОНУВАННЯ СТРІЧКОВИХ МАТРИЦЬ ВЕЛИКОЇ РОЗМІРНОСТІ

Дмитро Федасюк, Павло Сердюк, Юрій Семчишин

Національний університет "Львівська політехніка"
вул. С. Бандери, 12, Львів, 79013, Україна
e-mail: fedasyuk@lp.edu.ua, serdpavlo@yahoo.com, 7th@ukr.net

Резюме: *Розв'язування систем лінійних алгебраїчних рівнянь великої розмірності використовується при розв'язуванні багатьох задач математичної фізики, зокрема є однією з основних підзадач при розв'язуванні систем рівнянь в частинних похідних. Розподілене розв'язування систем лінійних рівнянь великої розмірності дає змогу зменшити час обчислень, особливо у випадках коли ці матриці неможливо зберігати у оперативній пам'яті одного комп'ютера. Предметом цього дослідження є пошук оптимальних алгоритмів секціонування матриць великої розмірності при розподіленому розв'язуванні систем лінійних алгебраїчних рівнянь.*

Ключові слова: *секціонування, СЛАР, розподілені обчислення, теплове проектування.*

ВСТУП

Велика кількість задач математичної фізики описується за допомогою систем рівнянь в частинних похідних. Після лінеаризації таких систем чисельними методами задачі зводяться до розв'язування систем лінійних алгебраїчних рівнянь (СЛАР). Розмірність таких систем може бути настільки великою, що їх розв'язання ускладнюється або й стає неможливим без використання спеціальних обчислювальних ресурсів та засобів [1, 4].

В свою чергу, у зв'язку зі зростанням потреби в потужних обчислювальних ресурсах та на порядок нижчою вартістю систем розподілення обчислень в порівнянні із спеціалізованими обчислювальними кластерами, останнім часом зросла актуальність використання систем розподілення обчислень [2].

Секціонування стрічкової матриці – розбиття її на такі підматриці – секції, що повністю включають в себе усі ненульові елементи.

Проблема секціонування стрічкових матриць великої розмірності постає, в першу чергу, при потребі розподіленого або розпаралеленого розв'язування СЛАР, особливо за умов, коли загальна структура матриці, якою описується система, нерегулярна, або з тих чи інших причин невідома [3].

Також потреба в секціонуванні матриць виникає при реалізації систем ітераційного розв'язування СЛАР; наприклад, в роботі [4] поміж іншим згадується про розбиття матриці на секції при ітераційному розв'язуванні СЛАР на паралельних системах з розподіленою пам'яттю, проте докладно алгоритм такого розбиття не описується.

Необхідність секціонування матриць виникає також при розв'язуванні низки таких специфічних задач, як, наприклад, проектування систем виявлення аномалій обчислювальних процесів, про що згадується в роботі [5], проте без опису конкретних алгоритмів.

В роботі [6] розглянуто розв'язування задачі знаходження добутку двох дійсних матриць на кластерах з серійних персональних комп'ютерів, перетворених на паралельну систему зі спільною пам'яттю (організованою за рахунок жорсткого диску з мережевим доступом), зокрема, з використанням секціонування матриць-множників.

Також розв'язування секціонованих СЛАР засобами паралельної системи зі спільною пам'яттю розглядається в роботі [7].

В роботі [8] згадується секціонування матриці для обробки частин паралельною системою з розподіленою пам'яттю та підтримкою стандарту MPI.

Проте у всіх розглянутих працях секціонування матриць було не основним предметом досліджень, а лише одним з аспектів, і тому дослідження ефективності алгоритмів секціонування не були представлені.

В цій роботі вперше здійснено постановку задачі та розробку алгоритмів секціонування стрічкових матриць великої розмірності з метою подальшого їх розподіленого розв'язування.

2. ПОСТАНОВКА ЗАДАЧІ СЕКЦІОНУВАННЯ

Задачею роботи є розробка та порівняння ефективних алгоритмів секціонування стрічкових матриць великої розмірності. Розглянемо детальніше поняття секціонування та вимоги до нього.

Нехай задана СЛАР вигляду:

$$A \cdot \vec{x} = \vec{b}\,, \tag{1}$$

де $A \in R^{n \times n}$ – дійсна квадратна матриця розмірності $n \times n$ стрічкового вигляду:

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & 0 & 0 \\ A_{21} & A_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{(n-1)(n-1)} & A_{(n-1)n} \\ 0 & 0 & \cdots & A_{n(n-1)} & A_{nn} \end{pmatrix}, \tag{2}$$

де A_{ij} – прямокутні матриці довільного розміру.

Секціонуваням, або розбиттям на секції, називатимемо знаходження для матриці A такого натурального вектора $\vec{s} \in N^m$ розмірності m, який би задовільняв одночасно дві умови:

- сума розмірів секцій повинна дорівнювати розмірності матриці A:

$$\sum_{i=1}^{m} \vec{s}_i = n\,; \tag{3}$$

- будь-який ненульовий елемент матриці A повинен знаходитись всередині квадратної області, утвореної двома послідовними секціями:

$$\forall i, j \in \overline{1,n} : a_{ij} \neq 0 \Rightarrow \exists k \in \overline{2,m} : \\ \left(\sum_{i=1}^{k-1} \vec{s}_i < i \leq \sum_{i=1}^{k} \vec{s}_i \right) \wedge \left(\sum_{i=1}^{k-1} \vec{s}_i < j \leq \sum_{i=1}^{k} \vec{s}_i \right). \tag{4}$$

3. ОЦІНКА ЯКОСТІ СЕКЦІОНУВАННЯ

Розглянемо приклад. Нехай дано матрицю $A \in R^{4 \times 4}$:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & 0 \\ A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & A_{33} & 0 \\ 0 & 0 & A_{43} & A_{44} \end{pmatrix}. \tag{5}$$

Для матриці A можливим розбиттям на секції буде

$$\vec{s}_1 = \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}, \tag{6}$$

при якому на головній діагоналі матриці A будуть розміщені три квадратні секції з розмірностями, відповідно, 1×1, 2×2 та 1×1:

$$A = \left(\begin{array}{c:cc:c} A_{11} & A_{12} & A_{13} & 0 \\ \hdashline A_{21} & A_{22} & 0 & 0 \\ 0 & 0 & A_{33} & 0 \\ \hdashline 0 & 0 & A_{43} & A_{44} \end{array}\right); \tag{7}$$

важливо, що за межами квадратних областей, утворених парами послідовних секцій (дві крайні позиції на побічній діагоналі), немає ненульових елементів.

Очевидно, що розбиття, яке задовільняє поставлені вимоги, є не єдиним; наприклад, для тієї ж матриці A можливим є також секціонування вигляду:

$$\vec{s}_2 = \begin{pmatrix} 2 & 1 & 1 \end{pmatrix}, \tag{8}$$

при якому матриця A буде розбита на секції наступним чином:

$$A = \left(\begin{array}{cc:c:c} A_{11} & A_{12} & A_{13} & 0 \\ A_{21} & A_{22} & 0 & 0 \\ \hdashline 0 & 0 & A_{33} & 0 \\ \hdashline 0 & 0 & A_{43} & A_{44} \end{array}\right). \tag{9}$$

При можливості багатьох альтернативних розбиттів матриці на секції постає необхідність порівняння їх між собою. Авторами пропонується використання наступних трьох критеріїв оцінювання якості секціонування матриць:

1) Дрібність секціонування. Отримання якнайбільшої кількості дрібних секцій дуже важливе для забезпечення гнучкості розподіленого розв'язування СЛАР, якнайдоцільнішого забезпечення завданнями усіх систем-виконавців та

мінімізації сумарного часу простою розподіленої системи:

$$cnt(\vec{s}) = m \to \max . \qquad (10)$$

2) Акуратність секціонування. Оскільки обчислювальна складність алгоритмів діагоналізації насиченої матриці розмірності $n \times n$, якою і є кожна з секцій, в загальному випадку становить $O(n^3)$, то, очевидно, що для максимальної швидкодії розподіленої системи сума кубів розмірів секцій повинна бути мінімальною:

$$sqr(\vec{s}) = \sum_{i=1}^{m} \vec{s}_i^{\,3} \to \min . \qquad (11)$$

3) Точність результату. Результати розв'язування однієї й тієї ж СЛАР, розбитої на секції кількома альтернативними способами, можуть відрізнятися. Якщо відомим є еталонний розв'язок СЛАР $\vec{x}$, то точність результату, отриманого завдяки секціонуванню матриці $\vec{x}'$ можна визначити як середнє квадратичне відхилення (стандартну девіацію) вектора $\vec{x}'$ відносно вектора $\vec{x}$:

$$dev(\vec{s}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\vec{x}_i - \vec{x}'_i)^2} \to \min . \qquad (12)$$

На думку авторів, запропоновані критерії оцінки якості секціонування матриць дозволять порівнювати між собою не лише конкретні варіанти розбиття на секції матриць, але й алгоритми секціонування загалом.

4. АЛГОРИТМИ СЕКЦІОНУВАННЯ

В ході дослідження можливих підходів до секціонування стрічкових матриць великої розмірності авторами було розроблено три різних алгоритми, які названо, відповідно до принципу дії, однонапрямленим, двонапрямленим та пристосовним.

Однонапрямлений (One-Directional, OD) алгоритм секціонування матриць складається з наступних кроків:

[OD0] Ініціалізація: виділити пам'ять, заповнити список розмірів секцій одиничними елементами.

[OD1] Обчислити для кожного елемента, що знаходиться на головній діагоналі матриці, дві величини: 1) відстані до найвіддаленішого ненульового елемента по горизонталі вправо або по вертикалі вниз; 2) відстані до найвіддаленішого ненульового елемента по горизонталі вліво або по вертикалі вгору.

[OD2] Переглянути всі пари послідовних секцій від першої і до останньої; у разі, якщо на рядках чи стовпцях, які належать секціям, є ненульові елементи, що не потрапляють всередину квадратної області, утвореної цими секціями, – об'єднати їх.

[OD3] Якщо на попередньому кроці виконано принаймі одне злиття – перейти до кроку OD2.

[OD4] Фіналізація: конвертувати список розмірів секцій в масив, повернути результат.

Двонапрямлений (Bidirectional, BD) алгоритм відрізняється від однонапрямленого переглядом секцій як в прямому, так і в зворотньому напрямку, що дозволяє зробити розбиття на секції рівномірнішим. Двонапрямлений алгоритм секціонування матриць складається з наступних кроків:

[BD0] Ініціалізація: виділити пам'ять, заповнити список розмірів секцій одиничними елементами.

[BD1] Обчислити для кожного елемента, що знаходиться на головній діагоналі матриці, дві величини: 1) відстані до найвіддаленішого ненульового елемента по горизонталі вправо або по вертикалі вниз; 2) відстані до найвіддаленішого ненульового елемента по горизонталі вліво або по вертикалі вгору.

[BD2] Переглянути всі пари послідовних секцій від першої до останньої в прямому порядку; у разі, якщо на рядках чи стовпцях, які належать секціям, є ненульові елементи, що не потрапляють всередину квадратної області, утвореної цими секціями, – об'єднати їх.

[BD3] Переглянути всі пари послідовних секцій від останньої до першої в зворотньому порядку; у разі, якщо на рядках чи стовпцях, які належать секціям, є ненульові елементи, що не потрапляють всередину квадратної області, утвореної цими секціями, – об'єднати їх.

[BD4] Якщо на одному з двох попередніх кроків виконано принаймі одне злиття – перейти до кроку BD2.

[BD5] Фіналізація: конвертувати список розмірів секцій в масив, повернути результат.

Пристосовний (Adjustable, AD) алгоритм серед кількох можливих об'єднань вибирає те, яке призведе до меншого розміру новоутвореної секції, що дозволяє зробити розбиття на секції дрібнішим. Пристосовний алгоритм секціонування матриць складається з наступних кроків:

[AD0] Ініціалізація: виділити пам'ять, заповнити список розмірів секцій одиничними елементами.

[AD1] Обчислити для кожного елемента, що знаходиться на головній діагоналі матриці, дві величини: 1) відстані до найвіддаленішого ненульового елемента по горизонталі вправо або по вертикалі вниз; 2) відстані до найвіддаленішого ненульового елемента по горизонталі вліво або по вертикалі вгору.

[AD2] Переглянути всі пари послідовних секцій від першої і до останньої; у разі, якщо на рядках чи стовпцях, які належать секціям, є ненульові елементи, що не потрапляють всередину квадратної області, утвореної цими секціями, – об'єднати ті дві з чотирьох послідовних секцій, об'єднання яких дозволить отримати найменший розмір новоутворення.

[AD3] Якщо на попередньому кроці виконано принаймі одне злиття – перейти до кроку AD2.

[AD4] Фіналізація: конвертувати список розмірів секцій в масив, повернути результат.

Кожен з трьох розроблених алгоритмів на кожному кроці свого виконання здійснює об'єднання двох секцій. Для кожного об'єднання виконується перегляд усіх пар послідовних секцій, що залишаються на поточному кроці. Оскільки початково секцій є n, то буде здійснено не більше ніж $(n-1)$ злиттів, для кожного з яких буде переглянуто в середньому $\frac{1}{2}(n-1)$ пар секцій. Відтак, обчислювальна складність розроблених алгоритмів становить $O((n-1)\cdot\frac{1}{2}(n-1)) = O\left(\frac{1}{2}(n-1)^2\right) < O(n^2)$. Тому ми можемо стверджувати, що усі розроблені алгоритми мають обчислювальну складність не вищу за $O(n^2)$.

Також варто зазначити, що, в разі потреби, першим кроком для кожного з описаних алгоритмів може бути впорядкування рядків та стовпців матриці за допомогою алгоритму Катхілла-Маккі, що дозволить зменшити ширину стрічки матриці, а, відтак, покращити дрібність та акуратність її секціонування.

5. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

Тестування розроблених алгоритмів проводилося на шести різних тестових СЛАР з розрідженими стрічковими матрицями, подібних до СЛАР, що розв'язуються в реальних задачах теплового проектування електронних пристроїв. Матриці тестових СЛАР зображено на рис. 1. Тестові СЛАР складалися, відповідно, з 134 (див. рис. 1а), 1095 (див. рис. 1б), 1648 (див. рис. 1в), 3870 (див. рис. 1г), 6240 (див. рис. 1д) та 8192 (див. рис. 1е) рівнянь.

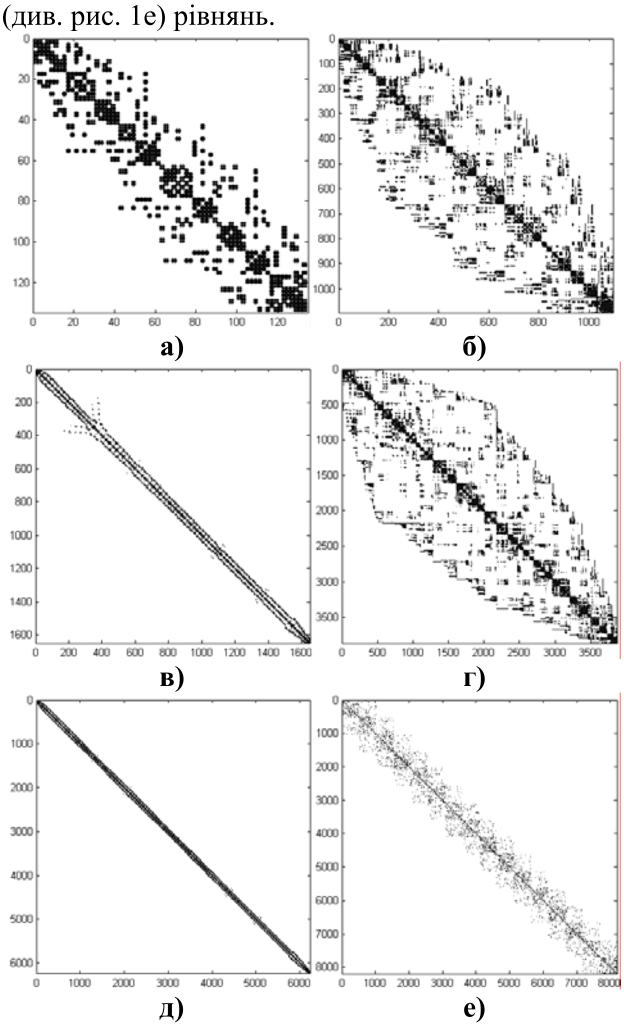


Рис. 1 – Матриці тестових СЛАР

В ході тестування було виконано розбиття на секції кожної з шести тестових матриць кожним з трьох розроблених алгоритмів секціонування.

Після цього матриці було розв'язано локально та розподілено, з використанням отриманого секціонування. Для кожного з отриманих розбиттів на секції було обчислено критерії якості секціонування *cnt*, *sqr* та *dev* а також визначено сумарний час розподіленої системи, витрачений на розв'язування СЛАР. Отримані результати наведено в табл. 1.

Таблиця 1. Результати експериментів

Матриця		Метод	Критерії якості			Час, с
№	*n*		*cnt*	*sqr*	*dev*	
1	134	OD	4	327 896	2.252682E-12	0.109
		BD	4	265 448	2.231834E-12	0.109
		AD	4	186 776	2.240844E-12	0.078
2	1 095	OD	3	268 793 367	1.651711E-12	45.781
		BD	3	206 520 705	1.551169E-12	52.328
		AD	3	295 266 231	1.660775E-12	48.328
3	1 648	OD	24	22 319 104	2.881726E-06	9.234
		BD	24	15 373 636	2.881726E-06	10.531
		AD	21	44 339 200	2.881726E-06	9.500
4	3 870	OD	2	14 638 398 840	1.677139E-14	2 096.781
		BD	3	10 463 074 488	1.677842E-14	2 081.641
		AD	3	6 945 711 480	1.677039E-14	1 896.891
5	6 240	OD	50	111 968 256	2.902765E-06	245.188
		BD	52	98 430 912	2.902765E-06	259.703
		AD	50	100 958 208	2.902766E-06	252.109
6	8 192	OD	8	11 365 478 054	5.177261E-10	72.734
		BD	8	18 610 300 226	5.177261E-10	108.359
		AD	8	10 717 585 418	5.177261E-10	42.000

Співвідношення дрібності секціонування (критерій *cnt*) для трьох різних алгоритмів секціонування подано в графічному вигляді на рис. 2.

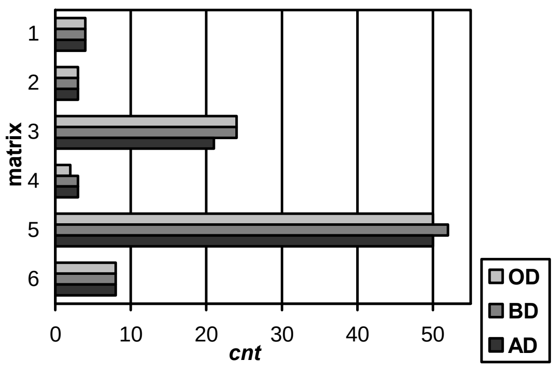


Рис. 2 – Дрібність секціонування (критерій *cnt*)

Можна стверджувати, що дрібність розбиттів на секції, отриманих трьома різними методами секціонування матриць, відрізняється незначно, проте двонапрямлений (BD) алгоритм, на відміну від двох інших, в усіх випадках забезпечив найвищу дрібність розбиття.

Співвідношення акуратності секціонування (критерій *sqr*) різних алгоритмів секціонування графічно представлено на рис. 3.

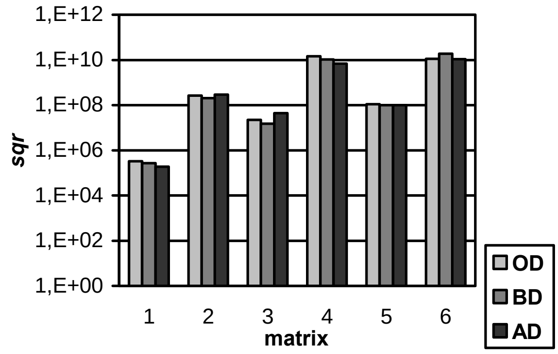


Рис. 3 – Акуратність секціонування (критерій *sqr*)

Значення акуратності секціонування відрізняються для різних алгоритмів, проте в значній мірі залежить також від структури тестової матриці, тому для різних матриць найефективнішими за акуратністю можуть виявитися різні алгоритми. Загалом, доцільним можна вважати використання двонапрямленого (BD) та пристосовного (AD) алгоритмів секціонування.

Значення середніх квадратичних відхилень розв'язку СЛАР, отриманого розподілено з використанням секціонування від еталонного розв'язку, отриманого локально, загалом достатньо низькі, тому можна стверджувати, що всі три алгоритми секціонування забезпечують достатню точність розв'язку СЛАР.

Усі три запропонованих алгоритми секціонування матриць підтвердили свою ефективність. Оскільки усі вони мають обчислювальну складність не вищу за $O(n^2)$, доцільним може бути використання одночасно декількох методів секціонування та вибір того результату, який буде найзручнішим для даного конкретного випадку. У випадку необхідності вибору єдиного алгоритму секціонування найперспективнішим видається двонапрямлений (BD).

6. ВИСНОВКИ

В роботі було сформульовано задачу секціонування стрічкових матриць великої розмірності при розподіленому розв'язуванні СЛАР та обґрунтовано її актуальність.

Запропоновані критерії якості секціонування – дрібність (критерій *cnt*), акуратність (критерій *sqr*) та точність результату (критерій *dev*) – дозволяють як оцінку якості конкретного розбиття на секції, так і порівняння самих алгоритмів секціонування матриць.

Розроблені та досліджені алгоритми секціонування матриць, а саме однонапрямлений (OD), двонапрямлений (BD) та пристосовний (AD), дозволяють виконати розбиття матриць на секції та мають обчислювальну складність, що не перевищує $O(n^2)$.

Результати серії експериментів, проведеної на тестових СЛАР, підтвердили ефективність усіх запропонованих алгоритмів секціонування матриць, зокрема двонапрямленого (BD).

7. Список літератури

[1] М. Ю. Баландин, Э. П. Шурина. *Методы решения СЛАУ большой размерности*. Новосибирск: Изд-во НГТУ, 2000. 70 с.

[2] Д. В. Федасюк, П. В. Сердюк, Ю. Б. Семчишин. Математичне та програмне забезпечення для розподіленого розв'язування параметричних задач математичної фізики. *Вісник НУ "Львівська політехніка" "Комп'ютерні системи проектування: теорія і практика"*. № 626. Львів: Вид-во НУ "Львівська політехніка". 2008. С. 94-102.

[3] D. Fedasyuk, P. Serdyuk, Y. Semchyshyn. Hierarchical distribution of high dimensional block-banded SLE solving. *Proceedings of the Xth International Conference CADSM 2009*. Lviv: Publishing House Vezha & Co. 2009. pp. 292-295.

[4] Р. Вайс, И. Подгаецкая, Х. Хёфнер, В. Шонауер. Итерационные методы решения систем линейных уравнений, от прошлого к будущему. *Математическое моделирование*. т. 13. № 2. 2001. С. 39-50.

[5] А. Беляев, С. Петренко. Системы обнаружения аномалий: новые идеи в защите информации. *Экспресс-Электроника*. № 2. 2004.

[6] А. Н. Заворин. Параллельное решение линейных систем при моделировании электрических цепейю *Математическое моделирование*. т. 3. № 3. 1991. С. 91-96.

[7] С. В. Востокин. Графический метод проектирования параллельных программ с использованием асинхронной событийной модели вычислений. *Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки*. № 30. 2004. С. 178-183.

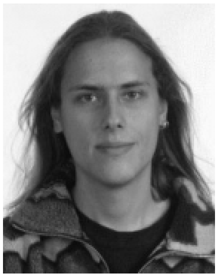
[8] А. В. Малышев, В. В. Шайдуров. Параллельные вычисления на кластерах из персональных компьютеров. *Труды Международной конференции RDAMM–2001*. т. 6, ч. 2. 2001. С. 287-293.

Дмитро Федасюк *народився 1955 року. Закінчив радіотехнічний факультет Національного університету "Львівська політехніка". Здобув ступінь кандидата технічних наук 1985 року, ступінь доктора технічних наук – 2000 року. Автор близько 140 наукових публікацій, в тому числі 2 монографій.*

Науковими інтересами є теплове проектування мікроелектронної та радіоелектронної апаратури, системи автоматизованого проектування та комп'ютерні інформаційні технології прийняття рішень.

Павло Сердюк *закінчив Факультет прикладної математики Львівського національного університету ім. І. Франка 2003 року. У 2007 році здобув ступінь кандидата технічних наук в Національному університеті "Львівська політехніка".*

Науковими інтересами є САПР, теплове проектування та чисельні методи математичної фізики.



Юрій Семчишин *закінчив Інститут комп'ютерних наук та інформаційних технологій Національного університету "Львівська політехніка" 2006 року. Станом на сьогодні навчається в аспірантурі.*
Науковими інтересами є розподілені обчислення та теплове проектування.

CiSj

computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

SECTIONING OF HIGH DIMENSIONAL BANDED MATRICES

Dmytro Fedasyuk, Pavlo Serdyuk, Yuriy Semchyshyn

Lviv Polytechnic National University
12 S. Bandery Street, Lviv, 79013, Ukraine
e-mail: fedasyuk@lp.edu.ua, serdpavlo@yahoo.com, 7th@ukr.net

Abstract: *Solving high dimensional systems of linear algebraic equations is of use to many problems of mathematical physics, in particular, it is one of the main subgoals at solving systems of equations in partial derivatives. Distributed solving of high dimensional systems of linear equations allows to reduce computing time, especially in cases when these matrices can not be kept in one computer's RAM. The subject of this study is the search of optimal high dimensional matrices sectioning algorithms for distributed solving systems of linear algebraic equations.*

Keywords: *Sectioning, SLE, Distributed Computing, Thermal Design.*

1. INTRODUCTION

Many problems of mathematical physics can be described by a system of equations in partial derivatives. After linearizing such systems with numerical methods, problems come to solving the systems of linear algebraic equations (SLE). The dimensions of such systems can be so high that their solution becomes difficult or impossible without the use of special computing resources and tools [1, 4].

In turn, due to growing of need for powerful computational resources and lowering cost of distributed computing systems in comparison with specialized computing clusters, the relevance of the distributed computing systems has recently strongly increased [2].

Sectioning banded matrix – splitting it into such submatrices – sections, that fully include all nonzero elements.

The problem of sectioning high dimensional banded matrices appears, in the first place, if distributed or parallelized SLE solving is necessary, especially under conditions when the structure of matrix, which describes the system, is irregular, or unknown for some reasons [3].

Also need for matrices sectioning occurs in the implementation of iterative SLE solving, such as in paper [4] among others is mentioned splitting the matrix into sections when iteratively solving SLE on parallel systems with distributed memory, but details of the sectioning algorithm are not described.

Need for matrices sectioning occurs also in solving a number of specific tasks, such as designing computational processes anomalies detection systems, as mentioned in the paper [5], but no descriptions of specific algorithms are provided.

In the paper [6] considered the solution of the problem of finding the product of two real matrices on clusters of the serial personal computers, converted to a parallel system with shared memory (organized by the hard disk with network access), in particular, by using sectioning matrix multiplier.

Also solving of sectioned SLE by parallel system with shared memory is considered in the paper [7].

In the paper [8] mentioned sectioning matrix for parallel processing by the system with distributed memory and support of MPI standard.

However, in all the works matrices sectioning was not the main subject of research, but only one of the aspects, and therefore investigation of sectioning algorithms effectiveness was not presented.

The problem was stated and algorithms for high dimensional banded matrices sectioning in order to further distributed solving was developed for the first time in this work.

2. SECTIONING PROBLEM STATEMENT

Objective is to develop and compare algorithms for high dimensional banded matrices sectioning. The concept of sectioning and requirements to it should be considered in more detail.

The sectioning, or partitioning to sections, is said to be the finding for matrix A such natural vector $\vec{s} \in N^m$ of dimension m, which would simultaneously satisfy both conditions:

- sum of sizes of sections should equal the dimension of matrix A;

- any nonzero matrix A element must be inside the square area formed by two consecutive sections.

3. SECTIONING QUALITY CRITERIA

Since there may be many alternative sectionings of the matrix, it is important to compare them among themselves. The authors were invited to use these three criteria for evaluating the quality of matrix sectioning:

1) Sectioning smallness. Getting the most amount of small sections is very important to ensure flexibility of a SLE distributed solving, provide the most expedient tasks for performing systems and minimize total time losses in distributed system.
2) Sectioning accuracy. Since the computational complexity of full matrix of dimensions $n \times n$, which is, in the general case, each of the sections, diagonalization algorithms is $O(n^3)$, it is apparent that for maximum performance of distributed system sum of cubes of sizes of sections should be minimal.
3) Results precision. The results of solving the same SLE divided into the sections by several alternative methods may vary. If a reference SLE solution $\vec{x}$ is known, the precision of the results obtained by matrix sectioning $\vec{x}'$ can be defined as the mean square deviation (standard deviation) of vector $\vec{x}'$ from vector $\vec{x}$.

According to the proposed criteria for evaluating quality of matrix sectionings allow comparison of not only the alternative matrix sectionings, but comparison of the entire sectioning algorithms too.

4. SECTIONING ALGORITHMS

During the investigation of possible approaches to the high dimensional banded matrices sectioning the authors have developed three different algorithms, which are named according to the principle of action One-Directional (OD), Bidirectional (BD) and Adjustable (AD).

Each of three algorithms developed at each its execution step combines two sections into one. For each combining, all pairs of consecutive sections remaining at the current step are iterated through. Since the original sections count is n, then no more than $(n-1)$ combinings will be performed, for each of which, on average, $\frac{1}{2}(n-1)$ pairs of sections will be iterated through. Therefore, the computational complexity of algorithms developed is $O\left((n-1)\cdot\frac{1}{2}(n-1)\right) = O\left(\frac{1}{2}(n-1)^2\right) < O(n^2)$. And so we can say that all algorithms developed have the computational complexity that not exceeds $O(n^2)$.

It should also be noted that, if necessary, first step for each of these algorithms may be extended with matrix rows and columns ordering using the Cathill-McKee algorithm that will reduce the width of the matrix band and, hence, improve sectioning smallness and accuracy.

5. EXPERIMENTAL RESULTS

The testing of the developed algorithm was performed with six different SLE tests with sparse and banded matrices, such as SLE that arises in the real problems of thermal design of electronic devices. SLE tests consisted of 134, 1095, 1648, 3870, 6240 and 8192 equations.

During the test, sectioning for each of six test matrices was obtained using each of three developed sectioning algorithms. Then matrices were solved locally and distributedly, using obtained sectionings. For each of sectionings criteria *cnt*, *sqr* and *dev* was calculated and total time spent for SLE solving were determined.

All three matrices sectioning algorithms proposed confirmed their effectiveness. Since they all have computational complexity that not exceeds $O(n^2)$, it may be expedient to use several sectioning methods at a time and then select results that will be most convenient for this case. If it is necessary to choose a single sectioning algorithm, then Bidirectional (BD) seems to be most promising.

6. CONCLUSION

We have stated and proved relevance of the problem of high dimensional banded matrices sectioning when solving SLE distributedly.

The proposed sectioning quality criteria – sectioning smallness (criterion *cnt*), sectioning accuracy (criterion *sqr*) and results precision (criterion *dev*) – allow comparison of alternative matrix sectionings, just as comparison of entire sectioning algorithms.

One-Directional (OD), Bidirectional (BD) and Adjustable (AD) algorithms for matrices sectioning under research and development can perform partitioning matrices to sections, and has computing complexity that does not exceed $O(n^2)$.

The experimental results proved the effectiveness of all three developed matrices sectioning algorithms, especially of Bidirectional (BD).