Proposed algorithms can work with data in the form of texts, audio and video files, files with various extensions such as .avi, .tif, .pdf and etc. Algorithms can generate digests of already encrypted files, this option gives a possibility to check the integrity of files without their decryption. Suggested methods of digest generation have a stream nature, the speed for constant m is linearly dependent on variable n. Growth of n increases the cryptographic stability. The implementation in the form of block by block compression is possible but it has a lack of motivation because the size of the block restricts the number of variables in the system of nonlinear equations.

The necessity of a further research and technological solutions on the constructions of key dependent hash functions is caused by cybersecurity calls, the increase of global information space, expectations of quantum computers appearance and development of bitcoins technology, which requires hashing of data of arbitrary size with its transformation into sequences of bits which form digests of the so called blockchains. Proposed algorithms of generation of sensitive for changes of documents digests will be used for cyberattacks detection and for the auditing of all files after registered intrusion.

Key words: *cybersecurity, hash functions, message authentication codes, homomorphism of compression, highly nonlinear multivariate cryptography, noncommutative cryptography.*

Одержано 27.01.2019

УДК 519.6
DOI: 10.32626/2308-5878.2019-19.180-187

О. М. Хіміч, член-кореспондент НАН України, д-р фіз.-мат. наук,
В. А. Сидорук, канд. фіз.-мат. наук

Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

ВИКОРИСТАННЯ МІШАНОЇ РОЗРЯДНОСТІ У МАТЕМАТИЧНОМУ МОДЕЛЮВАННІ

У роботі пропонується методика, за допомогою якої можна прискорити час математичного моделювання складних систем, використовуючи мішану розрядність в обчисленнях. Мішана розрядність дозволяє підвищити продуктивність обчислень, а також зекономити пам'ять. Показано застосування такого підходу при побудові алгоритму розв'язання систем лінійних алгебраїчних рівнянь з розрідженими матрицями.

Ключові слова: *математичне моделювання, мішана розрядність, паралельні алгоритми, розріджені матриці.*

Вступ. Проведення обчислень на довільній розрядності — один з основних інструментів підвищення ефективності програм і ідентифікації лінійних систем у комп'ютерному середовищі [1], зокрема,

виявлення поганообумовлених чи некоректних задач, задач з близькими чи кратним власними значеннями. Обчислення на довільній розрядності реалізуються програмно, а час виконання обчислень експоненціально залежить від розрядності числа з плаваючою комою, що використовується. Програмні бібліотеки такі як GNU Mpfr [2], GNU GMP [3] служать для реалізації обчислень на довільній розрядності. У бібліотеці mpfr, реалізовано ряд blas функцій.

В обчислювальних системах різних архітектур апаратно реалізовані половинна, одинарна, подвійна та розширена точність обчислень. У наукових розрахунках переважна більшість обчислень проводиться з подвійною точністю, тому в багатьох високопродуктивних процесорах одинична точність була усунена на користь емуляції операцій з одинарною точністю з використанням схеми подвійної точності. Водночас, використання даних у форматі з одинарною точністю дозволяє зберігати вдвічі більше даних на кожному рівні ієрархії пам'яті, включаючи кеш-пам'ять та регістрову пам'ять. До того ж, обробка значення одинарної точності вимагає використання меншої ширини смуги пропускання між різними рівнями пам'яті і зменшує кількість необхідного кешу і величину TLB промахів.

Враховуючи наведене вище, можна зробити висновок, що одним з шляхів підвищення ефективності обчислень і скорочення часу математичного моделювання є побудова паралельних алгоритмів, які б в процесі роботи виконували б обчислення на різній розрядності.

Розглянемо застосування такого підходу при побудові алгоритму розв'язання систем лінійних алгебраїчних рівнянь з розрідженими матрицями.

Постановка задачі. Розглянемо задачу

$$Ax = b \tag{1}$$

з симетричною додатно-визначеною розрідженою матрицею порядку n.

Однією з передумов розв'язання задачі (1) на комп'ютерах MIMD-архітектури з багатоядерними процесорами (CPU), (зокрема, з процесорами Intel Xeon Phi) є приведення матриці до такого виду, який дозволяє працювати з більш щільними групами ненульових елементів. Перетворення матриці зумовлене особливостями ієрархії пам'яті для систем з процесорами Intel Xeon Phi.

На даний момент існує велика кількість алгоритмів перевпорядкування елементів матриць: метод мінімальної степені, метод вкладених перерізів, метод паралельних перерізів і т. д. Кожен з цих алгоритмів дозволяє привести довільну розріджену структуру до більш регулярного вигляду. Найбільш зручну для паралельної обробки структуру матриці отримується після застосування до матриці методу вкладених, або паралельних, перерізів.

$$\tilde{A} = P^T AP = \begin{pmatrix} A_{11} & 0 & 0 & A_{1p} \\ 0 & A_{22} & 0 & A_{2p} \\ 0 & 0 & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{pmatrix},$$

де P — матриця перестановок, p — кількість діагональних блоків у матриці, блоки A_{pp}, A_{ip}, A_{pi}, A_{ii}, $i = \overline{1, p-1}$ зберігають розріджену структуру.

Таким чином, задача розв'язання (1) зводиться до розв'язування еквівалентної системи

$$\tilde{A}\tilde{x} = \tilde{b}, \tag{2}$$

де $\tilde{x} = P^T x$, $\tilde{b} = P^T b$.

Найбільш ефективним прямим методом розв'язання (2) є метод Холецького [4–6]. В статті буде розглянуто паралельний алгоритм який відповідає саме етапу факторизації матриці.

Паралельний алгоритм. Розіб'ємо матрицю A на блоки розмірністю $c \times c$. Далі для факторизації блочно-діагональної матриці застосуємо алгоритм запропонований в [7] для щільних матриць.

Для факторизації матриці на k-му кроці використаємо наступне співвідношення:

$$A^k = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{pmatrix}, \tag{3}$$

де розмірності блоків A_{11} — $c \times c$, A_{12} — $(n-kc)c$, A_{22} — $(n-kc)(n-kc)$, блоки A_{12} та A_{22} враховують структуру діагональних блоків та блоків обрамлення.

Звідси отримаємо алгоритм, за яким проводиться розвинення на k кроці:

$$A_{11} = L_{11} * L_{11}^T; \tag{4}$$

$$L_{21} = A_{21} * \left(L_{11}^T\right)^{-1}; \tag{5}$$

$$\tilde{A}_{22} = A_{22} - L_{21} * L_{21}^T. \tag{6}$$

Зазначимо, що реалізація (4)–(6) на кожному кроці модифікує тільки блоки D_{ii}, C_{pi}, $i = \overline{1, p-1}$, D_{pp}.

Нехай для розв'язування задачі на комп'ютері MIMD-архітектури маємо CPU з p процесорними ядрами. Для роботи алго-

ритму на *к* кроці реалізовується наступна декомпозиція даних: у пам'яті CPU (*i*) зберігається плитка A_{11} та плитки необхідні для модифікації підматриці A_{22}. $A_{pp}^{(i)}$ — набір плиток для модифікації діагонального блоку D_{pp}. На рис. 1. показано блочний розподіл даних на *к*-му кроці факторизації блочно-діагональної матриці з обрамленням, враховуючи запропоновану вище декомпозицію.

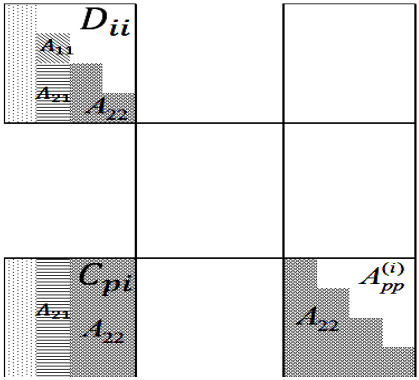

Рис. 1. *Декомпозиція даних в CPU на к-му кроці факторизації*

Враховуючи декомпозицію даних, приведену вище, плитковий алгоритм факторизації записується у наступній формі:

- над всіма діагональними блоками крім D_{pp} та відповідними блоками обрамлення послідовно виконуються такі операції:
  - на CPU факторизуємо A_{11} $A_{11} = L_{11} * L_{11}^T$;
  - паралельно в різних потоках модифікуємо стовпчик блоків L_{21} $L_{21} = A_{21}\left(L_{11}^T\right)^{-1}$;
  - незалежно в кількох потоках модифікуємо блоки матриці A_{22} за формулою:
$$\tilde{A}_{22} = A_{22} - L_{21}L_{21}^T;$$
- використовуючи операцію мультизбирання модифікуємо блок D_{pp} $\tilde{D}_{pp} = D_{pp} - \sum_{i=1}^{p-1} A_{pp}^{(i)}$;
- факторизуємо блок $\tilde{D}_{pp}$, тим самим завершуючи процес факторизації матриці A.

Оцінка прискорення. Для оцінки якості паралельних алгоритмів будемо використовувати коефіцієнт прискорення S_p, що обчислюється за формулою

$$S_p = T_1 / T_p,$$

де T_1 — час розв'язування задачі на комп'ютері з одним CPU, T_p — час розв'язування тієї ж задачі на комп'ютері з багатоядерним CPU.

Будемо вважати, що порядки всіх діагональних блоків приблизно рівні

$$q_i \approx q = \frac{n-s}{p-1},$$

де s — порядок останнього діагонального блоку.

Оскільки (4)–(6) виконуються паралельно і незалежно у всіх $p-1$ CPU і максимальна кількість операцій припадає на етап (6), то оцінка кількості операцій виконуваних на одному CPU визначається саме складністю етапу (6).

Кількість операцій необхідних для виконання (6) можна оцінити величиною

$$N_p \approx \frac{q^3}{3} + sq^2 = \frac{q^2}{3}(q+3s).$$

Обчислимо значення T_1 і T_p, використовуючи значення N_p знайдене вище.

$$T_1 \approx (p-1) N_p t \ , \ T_p \approx N_p t \ + \frac{(p-1)s^2}{2} t_{opp},$$

де t — час виконання однієї арифметичної операції, t_{opp} — час обміну між двома процесами.

Коефіцієнт прискорення для алгоритму оцінюється величиною

$$S_p \approx (p-1)\left(1 + \frac{3}{q^2(q+3s)}\left(\frac{(p-1)s^2}{2}\tau_{opp}\right)\right)^{-1}, \qquad (7)$$

де $\tau_{opp} = \dfrac{t_{opp}}{t}$.

Програмна реалізація та чисельні експерименти. При програмній реалізації алгоритму на комп'ютерах гібридної архітектури доцільно використовувати функції оптимізованих програмних бібліотек. Зокрема, до таких бібліотек відносяться Intel MKL [8]. Реалізація обмінів між процесорами відбувається за допомогою функцій MPI [9].

Розглянемо функції, що використовувались при програмній реалізації алгоритму:

- MPI_Reduce — функція глобальної редукції зі збереженням результату у вказаному процесорі;
- LAPACKE_dlange — повертає значення 1-норми, норми Фробеніуса, норми нескінченності або найбільшого абсолютного значення будь-якого елемента прямокутної матриці;
- dpotrf — знаходження LL^{T} розвинення щільної матриці;
- cblas_strsm, cblas_dtrsm — розв'язання трикутної системи з багатьма правими частинами на одинарній та подвійній точності, відповідно;
- cblas_sgemm, cblas_dgemm — знаходження добутку двох матриць на одинарній та подвійній точності, відповідно.

Розрядність на якій проводяться обчислення визначається наступним чином:

- вибір розрядності для обчислення (4) базується на числі обумовленості матриці відповідного блоку;
- у (5), (6) розрядність вибирається на основі евклідових норм відповідних блоків.

Обчислення проводились на вузлах суперкомп'ютера СКІТ [10], що мають наступні характеристики: 2 чотирьох ядерних Intel Xeon 5345 з тактовою частотою 2,2 ГГЦ, 16 ГБ оперативної пам'яті.

Для проведення чисельних експериментів вибрано ряд тестових матриць з колекції розріджених матриць університету Флориди [11]. Характеристики матриць приводяться в таблиці.

Таблиця

Набір тестових матриць

№ п./п.	Назва	Проблемна область	Порядок	Кількість ненульових елементів
1.	ecology2	2D/3D problem	999999	4,995,991
2.	apache2	structural problem	715 176	4 817 870
3.	thermomech_dM	thermal problem	204,316	1,423,116
4.	G2_circuit	circuit simulation problem	150 102	726 624
5.	Dubcova3	2D/3D problem	146,689	3,636,643
6.	cvxbqp1	optimization problem	50 000	349 968
7.	minsurfo	optimization problem	40 806	203 622

На рис. 2. показано залежність продуктивності від розміру плитки та при використанні різної розрядності. Результати приведені для матриці apache2 при використанні 6 потоків.

Рис. 2. *Залежність продуктивності від розрядності і розміру плитки*

Висновки. В статті розглянуто проблеми, що виникають при обчисленнях на довільній розрядності. Запропоновано алгоритм факторизації розрідженої матриці, який у ході розвинення матриці проводить обчислення на мішаній розрядності. Запропоновано критерії вибору розрядності, які в даному алгоритмі дозволяють зберігати точність обчислень. Використання запропонованого підходу при розв'язанні практичних задач показало значний приріст продуктивності, в порівнянні з обчисленнями на подвійній точності. Також на практиці використання мішаної розрядності в обчисленнях, дозволило значно економити пам'ять системи.

Список використаних джерел:

1. Nikolaevskaya E., Khimich A., Chystyakova T. Programming with Multiple Precision. Springer-Verlag Berlin Heidelberg. 2012. 234 p.
2. URL: https://www.mpfr.org.
3. URL: https://gmplib.org.
4. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. М. : Мир, 1984. 334 с.
5. Химич А. Н., Попов А. В., Полянко В. В. Алгоритмы параллельных вычислений для задач линейной алгебры с матрицами нерегулярной структуры. *Кібернетика і системний аналіз*. 2011. Вип. 47, № 6. С. 159–174.
6. Хіміч О. М., Сидорук В. А. Гібридний алгоритм розв'язування лінійних систем з розрідженими матрицями на основі блочного LL^T методу. *Комп'ютерна математика*. 2015. Вип. 1. С. 67–74.
7. Buttari A., Langou J., Kurzak J., Dongarra J. A Class of Parallel Tiled Linear Algebra Algorithms for Multicore Architectures. *Parallel Computing.* 2009. Vol. 35, Issue 1. P. 38–53.
8. Intel® Math Kernel Library (Intel® MKL). URL: https://software.intel.com/-en-us/intel-mkl.
9. Gropp W., Lusk E. and Thakur R. Using MPI-2: Advanced Features of the Message-Passing Interface. Cambridge : MIT Press. 1999. 382 p.

10. URL: http://icybcluster.org.ua.
11. URL: https://sparse.tamu.edu.

USE OF MIXED PRECISION IN MATHEMATICAL MODELING

The work proposes a method by which it is possible to accelerate the time of mathematical modeling of complex systems, using a mixed precision in calculations. The mixed precision allows you to improve the computing performance and save memory. Showing this approach in constructing an algorithm for solving systems of linear algebraic equations with sparse matrices.

Key words: *mathematical modeling, mixed precision, parallel algorithms, sparse matrices.*

Одержано 15.02.2019

УДК 519.6
DOI: 10.32626/2308-5878.2019-19.187-192

О. В. Чистяков, канд. фіз.-мат. наук

Інститут кібернетики імені В. М. Глушкова НАН України, м. Київ

ГІБРИДНИЙ ІТЕРАЦІЙНИЙ АЛГОРИТМ ДЛЯ РОЗВ'ЯЗУВАННЯ ЧАСТКОВОЇ ПРОБЛЕМИ ВЛАСНИХ ЗНАЧЕНЬ

Розглядається паралельний алгоритм поперемінно-трикутного методу для розв'язування на багатоядерному комп'ютері з графічними прискорювачами часткової узагальненої алгебраїчної проблеми власних значень розріджених симетричних матриць на основі їх зведення до блочно-діагонального виду з обрамленням.

Ключові слова: *гібридний комп'ютер, поперемінно-трикутний метод, розріджена матриця, алгебраїчна проблема власних значень.*

Вступ. Багато прикладних задач зводяться до розв'язування часткової узагальненої алгебраїчної проблеми власних значень (АПВЗ) для розріджених матриць великих розмірів. Наприклад, такі задачі виникають в електричних та механічних системах, в яких власні значення відповідають власним частотам коливань, а власні вектори характеризують відповідні форми (моди) коливань [1]. Визначення власних значень та векторів дають можливість аналізувати процеси та управляти ними.

З метою підвищення ефективності розв'язування задач на власні значення розріджених матриць великих розмірів у гібридному алгоритмі поперемінно-трикутного методу, який пропонується, використано ідею попереднього зведення вихідної розрідженої матриці A задачі виду