ISSN 1810-0198. Вестник Тамбовского университета. Серия Естественные и технические науки
Том 23, № 121 2018

УДК 519.61, 519.688
DOI: 10.20310/1810-0198-2018-23-121-100-108

ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ОБРАЩЕНИЯ ЦЕЛОЧИСЛЕННОЙ МАТРИЦЫ: РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

© С. А. Хворов

ФГБОУ ВО «Тамбовский государственный университет им. Г.Р. Державина»
392000, Российская Федерация, г. Тамбов, ул. Интернациональная, 33
E-mail: derbist27@gmail.com

Аннотация. Описаны параллельный алгоритм нахождения обратной матрицы с помощью присоединенной матрицы и определителя, его программная реализация и приведены результаты экспериментов, проведенных на кластере МВС-10П. Параллельный алгоритм основан на использовании Китайской теоремы об остатках и последовательном алгоритме, программно реализованном в системе компьютерной алгебры MathPartner. Граф описываемого алгоритма имеет двухуровневую структуру, достигнуто равномерное распределение данных между процессорами.
Ключевые слова: параллельный алгоритм; присоединенная матрица; определитель; система MathPartner; КТО; неравенство Адамара; метод Ньютона

1. Описание параллельного алгоритма

Пусть дана целочисленная квадратная матрица A порядка n. Матрицу A^*, транспонированную к матрице (A_{ij}) алгебраических дополнений, называют присоединенной. Если матрица A – невырожденная, то есть ее определитель не равен 0, то обратная к ней имеет вид: $A^{-1} = \frac{1}{detA} A^*$, где $detA$ – определитель матрицы A.

Таким образом, чтобы для квадратной матрицы порядка n найти обратную матрицу, надо вычислить один определитель порядка n и составить присоединенную матрицу, то есть вычислить n^2 определителей порядка $n-1$. Последовательный метод присоединенной матрицы имеет сложность n^5a^2, где $a = \log m$, m – наибольший по абсолютной величине коэффициент в матрице, модулярно же он имеет сложность n^4a^2. Для нахождения матрицы A^{-1} предлагается параллельный алгоритм, использующий китайскую теорему об остатках. Она позволяет найти обратную матрицу в простых полях, а затем получить ответ в рациональных числах. При этом, сложность вычислений в n раз уменьшается и становятся доступны параллельные потоки вычислений: в каждом простом поле вычисления происходят независимо. Общее количество арифметических машинных операций в таком алгоритме будет $\sim (mn^{1+\log_2 7} + n^2m^2)$. Если

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 16-07-00420)

число процессоров порядка $n(m/M)$, где M – разрядность машинного слова, то общее время вычислений на кластере будет оцениваться величиной $\sim n^{\log_2 7}$.

Теорема 1.1. *(Китайская теорема об остатках). Пусть числа* $m_1, m_2 ... m_k$ *– попарно взаимно простые,* $b_1, b_2, ..., b_k$ *– целые числа и* $M = m_1 * m_2 * ... m_k$. *Тогда система*

$$\begin{cases} x \equiv b_1 (mod\ m_1), \\ x \equiv b_2 (mod\ m_2), \\ \quad ...., \\ x \equiv b_k (mod\ m_k). \end{cases} \tag{1.1}$$

имеет единственное решение по модулю M, *и это решение может быть представлено в виде:*
$x = x_1 (mod\ M)$, *при* $x_1 = \frac{M}{m_1} b_1 y_1 + \frac{M}{m_2} b_2 y_2 + ... + \frac{M}{m_k} b_k y_k$ *и* y_i *обозначающим число, обратное числу* $\frac{M}{m_i}$ *относительно умножения по модулю* m_i *:*
$\frac{M}{m_i} y_i \equiv (mod\ m_i)$.

Для выяснения достаточного числа модулей необходимо знать, какое наибольшее число может появиться в результате. Обозначим евклидову норму вектора, образованного строкой i матрицы A через a_i :

$$||a_i|| = \sum_{i=0}^{i=n} \sqrt{a_{i,j}^2}. \tag{1.2}$$

Так как присоединенная матрица образована минорами исходной матрицы, то достаточно знать оценку для определителя матрицы. Для этого используется неравенство Адамара (Hadamard), оно дает достижимую верхнюю оценку для определителя

$$|det(A)| \leqslant \prod_{i=1}^{n} ||a_i||.$$

Так как может быть произвольный знак у определителя, то будем вычислять удвоенное выражение, стоящее справа. Учитывая, что некоторые строки могут оказаться нулевыми, мы будем вычислять такое выражение:

$$Had_2 = 2 \prod_{i=1}^{n} v_i, \text{где } v_i = \begin{cases} 1 & \text{если } ||a_i|| = 0 \\ ||a_i|| & \text{если } ||a_i|| > 0 \end{cases}. \tag{1.3}$$

Пусть дан набор всех простых чисел $p_1 < p_2 < \cdots < p_{k-1} < p_k$, размер которых не превышает 32 бит. Выбираются простые числа, начиная с конца, до тех пор пока их произведение не станет удовлетворять неравенству (1.4).

$$\prod_{i=q}^{k} p_i \geq Had_2. \tag{1.4}$$

Если количество ядер, выделенных на задачу, будет превышать количество полученных простых модулей, то список пополняется модулями до тех пор, пока их количество не станет кратным числу процессоров.

В итоге получится следующий набор простых чисел: $p_k, p_{k+1}, \ldots, p_{m-1}, p_m$.

Для восстановления элементов строк присоединенной матрицы и определителя, полученных в конечных полях, используется метод Ньютона.

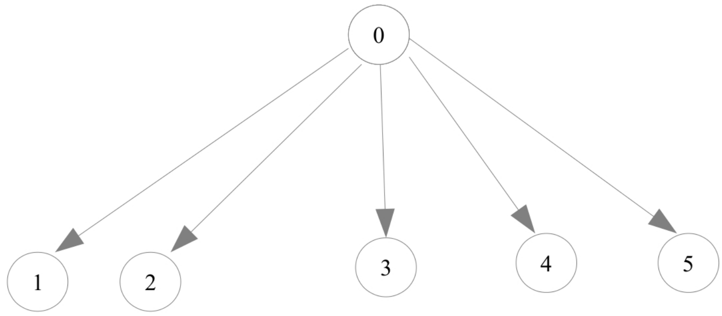
Метод Ньютона. *Пусть* n_{ij} *- это элемент, обратный к* m_i *по модулю* m_j. *Следовательно,* $n_{ij} * m_i = 1 \ mod \ m_j$.
Пусть $c_1 = b_1$; $\overline{c_1} = c_1 \ mod \ m_2$;
$c_2 = c_1 + (b_2 - \overline{c_1}) * m_1 * n_{1,2}$; $\overline{c_2} = c_2 \ mod \ m_3$;
$c_3 = c_2 + (b_3 - \overline{c_2}) m_1 * m_2 * n_{1,3} * n_{2,3}$; $\overline{c_3} = c_3 \ mod \ m_4$;
....
$c_n = c_{n-1} + (b_n - \overline{c_{n-1}}) m_1 * m_2 * ... * m_n * n_{1,n} * n_{2,n} * .... * n_{n-1,n}$

Тогда искомое число $x = c_n \ mod \ M$ *, где* $M = m_1 * m_2 * ... m_n$ *– произведение всех простых модулей.*

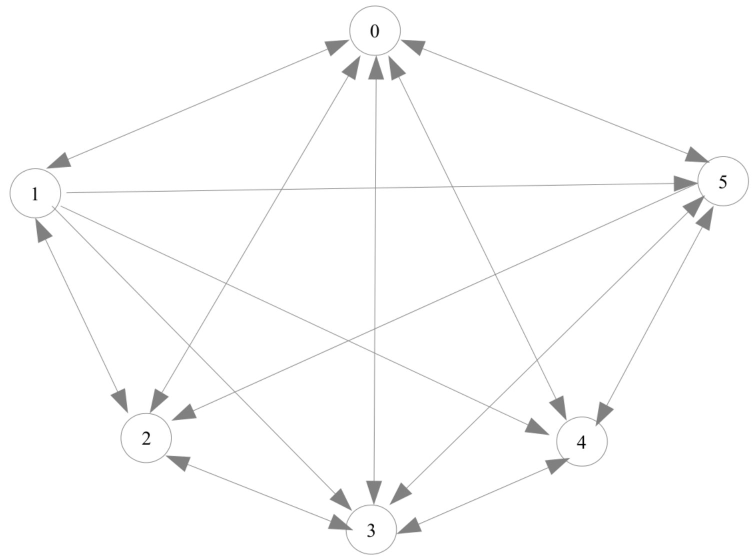
Параллелизм обеспечивается простым двухуровневым деревом алгоритма «корень – листья» и равномерным распределением простых модулей по листовым процессорам. Для восстановления присоединенной матрицы производится обмен строк модулярных матриц между всеми процессорами. Нулевой процессор собирает определители, полученные в конечных полях. Затем каждый процессор восстанавливает несколько строк присоединеной матрицы, помимо этого нулевой процессор восстанавливает определитель. После завершения процесса восстановления, все процессоры пересылают восстановленные строки нулевому для сборки искомой присоединенной матрицы. Выполняется деление присоединенной матрицы на определитель и в результате получается обратная матрица к исходной матрице A.

Таким образом в алгоритме используется три вида пересылок данных между процессорами:

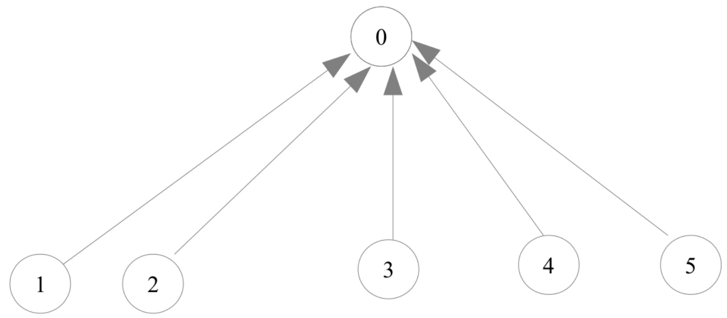
1. Рассылка исходной матрицы с нулевого всем процессорам.



2. Рассылка строк матриц, которые получены в конечных полях, между всеми процессорами для последующего восстановления.



3. Рассылка определителей и восстановленных строк присоединенной матрицы, полученных в конечных полях, от каждого процессора нулевому.



2. Результаты экспериментов

На основе разработанного алгоритма был написан программный код на языке java в системе комепьютерной алгебры MathPartner. Все эксперименты проводились на кластере МВС-10П, который располагается в МСЦ РАН.

Данный кластер имеет 207 вычислительных узлов. Каждый из узлов имеет по 2 процессора, оснащенных 8 ядрами каждый. Общее количество оперативной памяти на одном узле равно 64 гигабайтам.

В таблице указано время вычисления присоединенной матрицы и определителя, p (по горизонтали) – число ядер, n (по вертикали) – размер матрицы, S – программная реализация последовательного алгоритма вычисления присоединенной матрицы и определителя, реализованная в MathPartner, время измеряется в секундах. Исходные матрицы имели 100% плотность и 8-битные коэффициенты.

Таблица 1: Результаты тестирования программной реализации алгоритма

n/p	S	4	8	16	32	64	128	256	512	1024
8	0.1	0.2	0.1							
16	0.1	0.3	0.2	0.1						
32	0.2	0.4	0.3	0.3	0.1					
64	1	0.9	0.8	0.7	0.5	0.2				
128	20	4.5	4	2.3	2	1.3	1			
256	557	50	28	16	10	7	6	3.7		
512	$4{*}10^3$	768	415	213	110	59	36	24	17	
1024	10^4	$4{*}10^3$	$3{*}10^3$	$1.5 * 10^3$	10^3	868	445	275	189	137

Исходя из результатов, представленных в таблице, можно сделать вывод, что для матриц порядка, не превышающего 32, время выполнения параллельного алгоритма вычисления обратной матрицы больше, чем время выполнения последовательного алгоритма, программно реализованного в системе компьютерной алгебры MathParnter. Для матриц порядка от 32 до 64 включительно, время выполнения примерно одинаково. Для матриц порядка выше 64, параллельный алгоритм работает значительно быстрее последовательного.

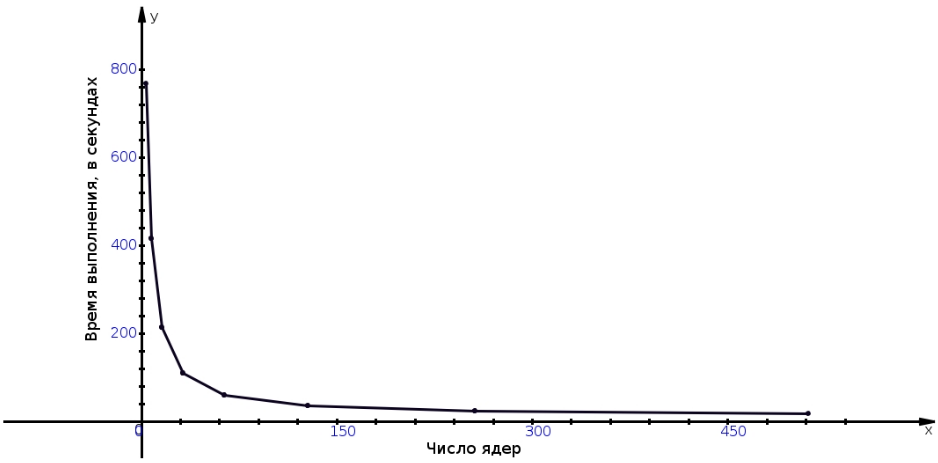


График 1. Зависимость времени выполнения параллельного алгоритма от количества ядер для плотной матрицы, размер которой равен 512x512.

Данный график демонстрирует, что с увеличением числа процессоров при работе с матрицей одного и того же порядка, время выполнения программы значительно снижается.

Для сравнения использовалось время выполнения параллельного алгоритма на 4 ядрах. p (по горизонтали) – число ядер, n (по вертикали) – размер матрицы, оценка масштабируемости измеряется в процентах. Исходные матрицы имели 100% плотность и 8-битные коэффициенты.

Из данной таблицы следует, что параллельный алгоритм наиболее эффективен для матриц порядка 128 и выше. Для матрицы с небольшими размерами эффективность снижается, с увеличением числа процессоров.

Таблица 2: Масштабируемость параллельного алгоритма

n/p	8	16	32	64	128	256	512	1024
8	100							
16	75	50						
32	67	33	50					
64	56	32	23	28				
128	56	49	28	22	14			
256	89	78	62	45	26	21		
512	92	90	87	81	67	50	35	
1024	67	67	50	29	28	23	16	11

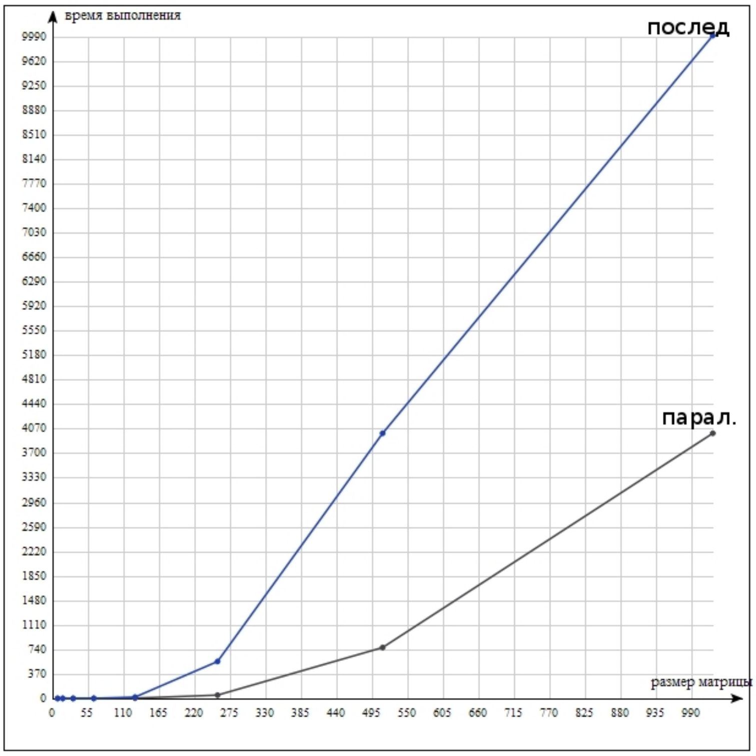


График 2. Рост времени вычисления присоединенной матрицы и определителя с увеличением порядка матрицы.

Первый кривая представляет собой программную реализацию последовательного алгоритма, реализованного в проекте MathParner, второй кривая отвечает за параллельную реализацию, запущенную на 4 процессорах. Как видно из графика, рост времени выполнения последовательного алгоритма и паралльного аглгоритмов для матриц небольшого порядка практически идентичен. Но при порядке 100 и выше, время выполнения последовательного растет значительно быстрее, чем у параллельного. Если рассмотреть эксперименты в целом, то можно сделать вывод, что параллельный алгоритм вычисления присоединенной матрицы и определителя в конечных полях, а затем их восстановления, в несколько раз эффективнее, нежели, чем прямой последовательный алгоритм.

СПИСОК ЛИТЕРАТУРЫ

1. *Малашонок Г.И.* Дискретная математика с элементами компьютерной алгебры. Тамбов: Изд-во ТГУ им. Г.Р. Державина, 2005.

2. *Малашонок Г.И.* Матричные методы вычислений в коммутативных кольцах. Тамбов: Изд-во Тамб. ун-та, 2002.

3. *Малашонок Г.И.* О вычислении ядра оператора, действующего в модуле // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2008. Т. 13. Вып. 1. С. 129–131.

4. *Хворов С.А.* Параллельный алгоритм обращения матрицы: результаты экспериментов // Компьютерная алгебра: материалы Междунар. конф. М., 2016. С. 63–65.

5. *Хворов С.А.* Параллельный алгоритм обращения целочисленной матрицы: эксперименты на кластере МВС-10П // International Conference on Mathematical Partnership, Parallel Computing and Computer Algebra: MathParCA-2017. Ierusalim, 2017. P. 43–48.

6. *Малашонок Г.И.* Компьютерная математика для вычислительной сети // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2010. Т. 15. Вып. 1. С. 322–327.

7. *Малашонок Г.И.* Управление параллельным вычислительным процессом // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2009. Т. 14. Вып. 1. С. 269–274.

Поступила в редакцию 23 января 2018 г.
Прошла рецензирование 14 февраля 2018 г.
Принята в печать 20 февраля 2018 г.

Хворов Сергей Александрович, Тамбовский государственный университет им. Г. Р. Державина, г. Тамбов, Российская Федерация, аспирант, кафедра функционального анализа, e-mail: derbist27@gmail.com

Для цитирования: *Хворов С.А.* Параллельный алгоритм обращения целочисленной матрицы: результаты экспериментов // Вестник Тамбовского университета. Серия Естественные и технические науки. Тамбов, 2018. Т. 23. № 121. С. 100–108. DOI: 10.20310/1810-0198-2018-23-121-100-108

DOI: 10.20310/1810-0198-2018-23-121-100-108

PARALLEL INVERSION OF INTEGER MATRIX: THE RESULTS OF THE EXPERIMENTS

© **S. A. Khvorov**

Tambov State University named after G.R. Derzhavin
33 Internatsionalnaya St., Tambov 392000, Russian Federation
E-mail: derbist27@gmail.com

Abstract. We focuses on results of experiments of the parallel algorithm for finding the inverse matrix through the adjoint matrix and determinant. A parallel algorithm based on the use of the Chinese remainder theorem and sequential algorithms implemented in the computer algebra system MathPartner. Graph of algorithm has a two-tier structure, achieved a uniform distribution between processors.
Keywords: parallel algorithm; adjoint matrix; determinant; system Math Partner; CRT; inequality of Hadamard; method of Newton

REFERENCES

1. Malaschonok G.I. *Diskretnaya matematika s elementami komp'yuternoy algebry* [Discrete Mathematics with Elements of Computer Algebra]. Tambov, Publishing of Tambov State University, 2005. (In Russian).

2. Malaschonok G.I. *Matrichnye metody vychisleniy v kommutativnykh kol'tsakh* [Matrix Methods of Computations in Commutative Rings]. Tambov, Publishing House of Tambov State University, 2002. (In Russian).

3. Malaschonok G.I. O vychislenii yadra operatora, deystvuyushchego v module [On the calculation of the kernel of the operator acting in the module]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Reports. Series: Natural and Technical Sciences*, 2008, vol. 13, no. 1, pp. 129–131. (In Russian).

4. Khvorov S.A. Parallel'nyy algoritm obrashcheniya matritsy: rezul'taty eksperimentov [Parallel algorithm of matrix inversion: the results of the experiments]. *Materialy Mezhdunarodnoy konferentsii: «Komp'yuternaya algebra»* [Proceedings of International Conference: «Computer Algebra»]. Moscow, 2016, pp. 63–65. (In Russian).

5. Khvorov S.A. Parallel'nyy algoritm obrashcheniya tselochislennoy matritsy: eksperimenty na klastere MVS-10P [Parallel algorithm for inversion of an integer matrix: the experiments on the MVS-10P cluster]. *International Conference on Mathematical Partnership, Parallel Computing and Computer Algebra: MathParCA-2017.* Ierusalim, 2017, pp. 43–48. (In Russian).

6. Malaschonok G.I. Komp'yuternaya matematika dlya vychislitel'noy seti [Computer mathematics for the computer network]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Reports. Series: Natural and Technical Sciences*, 2010, vol. 15, no. 1, pp. 322–327. (In Russian).

The work is partially supported by the Russian Fund for Basic Research (project № 16-07-00420)

7. Malaschonok G.I. Upravlenie parallel’nym vychislitel’nym protsessom [Management of parallel computing process]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Reports. Series: Natural and Technical Sciences*, 2009, vol. 14, no. 1, pp. 269–274. (In Russian).

Received 23 January 2018
Reviewed 14 February 2018
Accepted for press 20 February 2018

Khvorov Sergey Aleksandrovich, Tambov State University named after G.R. Derzhavin, Tambov, the Russian Federation, Post-Graduate Student, Functional Analysis Department, e-mail: derbist27@gmail.com

For citation: Khvorov S.A. Parallel’ny algoritm obrashcheniya tselochislennoy matritsy: rezul’taty eksperimentov [Parallel inversion of integer matrix: the results of the experiments]. *Vestnik Tambovskogo universiteta. Seriya Estestvennye i tekhnicheskie nauki – Tambov University Reports. Series: Natural and Technical Sciences*, 2018, vol. 23, no. 121, pp. 100–108. DOI: 10.20310/1810-0198-2018-23-121-100-108 (In Russian, Abstr. in Engl.).