УДК 519.613

Параллельный алгоритм обращения матрицы: результаты экспериментов

С. А. Хворов

Лаборатория алгебраических вычислений,
Тамбовский государственный университет,
ул. Интернациональная, д.33, Тамбов, Россия, 392000

Исследуется параллельный алгоритм вычисления обратной матрицы. Параллельный алгоритм основан на использовании китайской теоремы об остатках и последовательном алгоритме, который имеет сложность матричного умножения и реализован в системе компьютерной алгебры Math Partner.

Ключевые слова: присоединенная матрица, определитель, неравенство Адамара, параллельный алгоритм, метод Ньютона.

1. Введение

Мы исследуем параллельный алгоритм вычисления обратной матрицы. Данный алгоритм основан на китайской теореме об остатках. В его основе лежит последовательный алгоритм [1], который вычисляет определитель $\det A$ матрицы A и присоединенную матрицу A^* и имеет сложность матричного умножения. Обратная матрица равна A^{-1} вычисляется как $A^*/\det A$.

2. Основная часть

Данный алгоритм использует китайскую теорему об остатках и алгоритм Ньютона для получения решения в рациональных числах.

Алгоритм Ньютона может быть записан следующим образом:
$c_1 = r_1$; $\bar{c}_1 = c_1 \ mod \ m_2$,
$c_2 = c_1 + (r_2 - \bar{c}_1)m_1 n_{12}$; $\bar{c}_2 = c_2 \ mod \ m_3$ и так далее,
$c_n = c_{n-1} + (r_n - \bar{c}_{n-1})m_1 ... m_n n_{1n} ... n_{n-1n}$,

Результат: $x = c_n \ mod \ M$,

Здесь $m_1, m_2, ..., m_n$ – попарно взаимно простые модули, $r_1, r_2, ..., r_n$ – целые числа, $M = m_1 m_2 ... m_n$, n_{ij} – это обратный к m_i элемент по модулю m_j.

Такой алгоритм имеет два преимущества по сравнению с алгоритмом, в котором все вычисления производятся в целых числах: сложность вычислений в n раз уменьшается и становятся доступны параллельные потоки вычислений. В каждом простом поле вычисления происходят независимо и общее количество арифметических машинных операций в таком алгоритме будет $\sim (mn^{1+\log_2 7} + n^2 m^2)$. Здесь n – размер матрицы, а m – разрядность коэффициентов.

Для оценки абсолютного значения наибольшего числового коэффициента используется неравенство Адамара: $|det(A)| \leqslant \prod_{i=1}^{n} (\sum_{j=1}^{n} a_{i,j}^2)^{1/2}$.

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

$$Had_2 = 2 \prod_{i=1}^{n} v_i, \text{где } v_i = \begin{cases} 1 & \text{если } \sqrt{\sum_{j=1}^{n} a_{i,j}^2} = 0 \\ \sqrt{\sum_{j=1}^{n} a_{i,j}^2} & \text{если } \sqrt{\sum_{j=1}^{n} a_{i,j}^2} > 0 \end{cases}$$

Пусть дан набор всех простых чисел $p_1 < p_2 < \cdots < p_{m-1} < p_m$, размер которых не превышает 32 бит. Будем выбирать простые числа до тех пор пока их произведение не станет удовлетворять неравенству (1):

$$\prod_{i=q}^{m} p_i \geq 2Had_2{}^{\frac{1}{2}} + 2 \tag{1}$$

Дополним полученный список простых чисел так, чтобы их количество было кратно числу процессоров. Тогда все процессоры будут равномерно загружены. В итоге получится следующий набор простых чисел: $p_l, p_{l+1}, \ldots, p_{m-1}, p_m$.

Параллелизм обеспечивается простым двухуровневым деревом алгоритма и равномерным распределением простых модулей по листовым процессорам.

Алгоритм состоит из следующих шагов:

1. Нулевой процессор рассылает исходную матрицу A всем остальным процессорам.
2. Каждый процессор $\mathbf{Pr}_k$ генерирует общий массив модулей, опираясь на оценку Had_2, и выбирает из него подмассив P_k тех простых чисел, с которым он будет вычислять обратную матрицу. Но вычисления запускаются на со всеми этими модулями, а только примерно в корень раз меньше.
3. Каждый процессор k переводит матрицу A из кольца Z в кольцо Z_{p_i}, где $p_i \in P_k$, вычисляет присоединенную матрицу и определитель в кольце Z_{p_i}.
4. Процессоры распределяют и рассылают между собой строки присоединенных матриц так, чтобы каждому нужно было восстановить примерно одинаковое число строк.
5. Процессоры отсылают определители, полученные в кольцах Z_{p_i} нулевому процессору для последующего восстановления;
6. Каждый процессор восстанавливает строчки присоединенной матрицы по методу Ньютона. Если хотя бы одно из чисел было восстановлено неверно (модулей не хватило), то процессор посылает всем остальным сообщение об этом. В массив модулей добавляется некоторое количество модулей, равное числу процессоров в группе, и алгоритм возвращается к шагу[3];
7. Если же все элементы всех строк были восстановлены правильно, процессоры отсылают восстановленные строки нулевому процессору;
8. Нулевой процессор собирает все присланные строки в искомую присоединенную матрицу в кольце Z и вычисляет обратную матрицу.

Таким образом алгоритм использует четыре вида пересылок данных между процессорами:

1. Рассылка исходной матрицы с нулевого процессора всем процессорам.
2. Рассылка строк матриц, которые получены в конечных полях, между всеми процессорами для восстановления.
3. Рассылка сообщения о необходимости продолжить выполнение алгоритма и добавить модули.
4. Пересылка восстановленных строк присоединенной матрицы от каждого процессора нулевому.

3. Заключение

На основе данного подхода были разработаны алгоритмы и программы. Эти алгоритмы находятся в стадии тестирования и экспериментирования. В докладе будут сообщены результаты экспериментов, которые будут проведены с кластером, содержащим сотни процессоров, и имеющим распределенную память.

Благодарности

Работа частично поддержана грантом РФФИ № 16-07-00420.

Литература

1. *Малашонок Г.И.* Матричные методы вычислений в коммутативных кольцах. Монография. Тамбов: Изд-во ТГУ им. Г.Р.Державина, 2002.
2. *Малашонок Г.И.* Параллельная компьютерная алгебра. Тамбов: Изд-во Тамбовского университета, 2011

UDC 519.613

Parallel Algorithm for Matrix Inversion: Results of Experiments

S.A. Khvorov

Laboratory for algebraic computations
Tambov State University
Internatsionalnaya,33, Tambov, Russia, 392000

We study parallel algorithm for computing the inverse matrix. This parallel algorithm is based on the use of the Chinese remainder theorem and sequential algorithm, which has a complexity of matrix multiplication and is implemented in the system of computer algebra Math Partner.

Key words and phrases: adjoint matrix, determinant, Chinese remainder theorem, inequality of Hadamard, parallel algorithm, Newton method .