УДК 519.688

КОНСТРУКТИВНАЯ МАТЕМАТИКА И ПРИНЦИП БЛИЗКОДЕЙСТВИЯ

© Г. И. Малашонок

Ключевые слова: конструктивная математика, принцип близкодействия, параллельный алгоритм, кластерные вычисления.

Обсуждаются современные направления развития конструктивной математики, которые связаны с появлением многопроцессорных вычислительных систем. Обсуждаются требования к алгоритмам и структурам данных, которые предназначены для многопроцессорных систем.

1 Введение

Причины развития отдельных научных областей – это предмет философии науки. Когда говорят о причинах развития математики последних веков, считается общепризнанным, что внешние побудительные мотивы развития находились в области естествознания и, главным образом, в области решения физических задач, а во второй половине XX в. развитие вычислительной техники стало сильно сказываться на развитии конструктивной математики.

Первые поколения компьютеров, представляющие собой, по-сути, большие арифмометры, привели к развитию такого направления, как численные методы. В большинстве разделов математики появились дополнительные главы, посвященные численным методам. Главная проблема всех численных методов – это устойчивость вычислительных методов, борьба с погрешностью вычислений. С ростом масштабов вычислений детерминистские численные методы почти повсеместно проиграли итерационным численным методам. Появились такие разделы, как теория алгоритмов, теория сложности вычислений.

Следующие поколения компьютеров, позволяющие проводить сложную символьную обработку текстов, привели к появлению символьной компьютерной математики, которую обычно называют компьютерной алгеброй. Выкладки, которые приходится делать студенту, который изучает математику или применяет математический аппарат в прикладных задачах, могут быть осуществлены с помощью компьютера. При этом, с одной стороны, можно избежать механических ошибок, а с другой стороны, можно оперировать очень большими аналитическими выражениями, недоступными без компьютера.

Это привело к развитию многих конструктивных направлений в математике, которые не могли быть интересны в эпоху расцвета численных методов.

Однако аналитические вычисления характеризуются взрывным ростом вычислительной сложности. Поэтому и на персональных компьютерах, и на крупных рабочих станциях удается решать только относительно небольшие задачи, скорее учебного, чем производственного характера.

Современный компьютерный парк представляют вычислительные комплексы, составленные из десятков тысяч параллельно работающих процессоров, имеющих собственную память и средства коммуникации друг с другом и с ведущим процессором.

Такое оборудование потенциально способно решать гигантские аналитические и численно-аналитические задачи. Однако отсутствует теория параллельных алгоритмов.

Параллельные алгоритмы принципиально отличаются от последовательных алгоритмов наличием, кроме собственно вычислительной, еще и коммуникативной составляющей: нужно не только вычислять промежуточные результаты, но и пересылать их от одного процессора к другому согласованным образом. Самые замечательные алгоритмы, которые имеют низкую вычислительную сложность, могут быть совершенно не пригодны для параллельных вычислений.

Рассмотрим два класса алгоритмов, которые играют роль «крайних случаев» для параллельных вычислительных алгоритмов.

Самый замечательный класс параллельных алгоритмов это переборные алгоритмы. Это алгоритмы, которые сводятся к поиску решения путем перебора конечного множества претендентов на решение. Например, путем подстановки в некоторую функцию, которая возвращает результат «истина», если претендент является искомым ответом, и «ложь», если претендент не подходит.

Например, задача поиска минимума функции в некоторой области может решаться таким переборным алгоритмом. Нужно разбить область поиска на много подобластей и искать минимум в каждой подобласти, а затем сравнить полученные значения и выбрать минимальный.

Самый неудобный класс алгоритмов – итерационные алгоритмы. Это алгоритмы, у которых следующая итерация вычислений не может начаться, пока не завершится предыдущая. Например, алгоритм, который вычисляет 10000-й член последовательности, у которой задано 10 первых членов и каждый последующий член определяется предыдущими десятью членами.

2 Матричные алгоритмы и принцип близкодействия

Важный класс алгоритмов составляют конструктивные матричные алгоритмы. Это алгоритмы вычисления матричных функций, когда матрицы имеют большой размер, а элементами матриц являются символьные или численно-символьные выражения.

Классический подход в виде вычисления LU-разложения или применение метода Гаусса не позволяют получить эффективную параллельную программу. Во-первых, на каждом шаге требуется пересылать данные из текущей строки во все остальные строки. Во-вторых, всякий раз, когда необходимо выбирать ведущий элемент, а без этого, как известно, алгоритм Гаусса просто остановится, например, на нулевом ведущем элементе, нужно запускать очень «дорогой» коммуникативный процесс. Результатом такого процесса должен быть выбор нового ведущего элемента, перестроение вычислительного процесса и возобновление процесса вычислений.

Для такой организации требуется некоторое централизованное управление вычислительным процессом. Этапы вычислений сменяются этапами перестройки вычислительного процесса, и чем больше процессоров, тем «дороже» перестройка вычислительного процесса.

Принципом «близкодействия» будем называть такую организацию вычислительного процесса, при которой в вычислительный процесс вовлекаются прежде всего близко расположенные элементы, т. е. элементы блоков. Запрещено обращаться к отдельным далеко расположенным элементам. Вместо этого используется взаимодействие крупных блоков.

При такой организации вычислений нет лишних затрат на коммуникативные процессы, на централизованное управление и на перестройку вычислительного процесса.

Принцип близкодействия благотворно сказывается даже на последовательном вычислительном процессе, протекающем в одном процессоре. Как известно, сегодня повсеместно используется многоуровневая память. Около процессора располагается самый скоростной кэш первого уровня, за ним располагается кэш второго уровня, более медленный, затем отдельный чип с оперативной памятью. В процессе вычислений данные из ОЗУ перегружаются блоками в кэш второго уровня, а из него в кэш первого уровня, с которым обменивается данными процессор с самой высокой скоростью.

Поэтому алгоритмы, учитывающие принцип «близкодействия», будут всегда в выигрышной ситуации по сравнению с алгоритмами, которые вынуждены часто обращаться в далекие друг от друга области ОЗУ. Хорошим примером тут может послужить алгоритм умножения матриц. Если матрицы хранятся как вектор из строк, то известный алгоритм умножения матриц «строка-на-столбец» будет проигрывать по сравнению с любым другим алгоритмом, у которого не требуется передвигаться по столбцу, перемещаясь по далеким друг от друга элементам с одной строки на другую.

Альтернативным является блочный матричный алгоритм. Кроме того, можно предложить алгоритм, в котором элемент $a(i,j)$ первого сомножителя после умножения на строку j второго сомножителя подсуммируется к строке i произведения.

Отметим, что рекурсивный локально-блочный подход оказался плодотворным при создании эффективных матричных алгоритмов в коммутативных кольцах ([1-5]). Он позволил построить алгоритмы, имеющие сложность матричного умножения.

Программа пересмотра матричных алгоритмов и создания алгоритмов, удовлетворяющих принципу близкодействия, была запущена с момента выхода работы [1] в 2002г. Алгоритмы, известные к тому времени, не могли преодолеть проблему возможного появления нулевых ведущих элементов.

В 2005 г. появился первый алгоритм обращения матрицы, удовлетворяющий принципу близкодействия [6]. В 2008 г. был создан алгоритм вычисления ядра линейного оператора и присоединенной матрицы для матриц над коммутативными областями [7]. Для матрицы над полем произвольной характеристики алгоритм вычисления обобщенного разложения Брюа был опубликован в 2010 г. [8, 9].

Задача создания алгоритма разложения Гаусса в коммутативной области, удовлетворяющего такому принципу близкодействия, пока ожидает своего решения. Известный блочный рекурсивный алгоритм вычисления разложения Гаусса предполагает, что у исходной матрицы ранга r все r ведущих угловых миноров отличны от нуля [1]. Это ограничение не позволяет создавать универсальный блочный алгоритм разложения Гаусса. Отметим, решение этой задачи приведет к обобщению разложения Брюа на область целостности, а также должно привести к ускорению вычисления базиса Гребнера полиномиального идеала по методу Фужера.

3 Разложение Гаусса

Пусть R – коммутативное кольцо, $A = (a_{i,j})$ – матрица с элементами $a_{i,j}$, размера $n \times m$ ранга r над R, I_n – единичная матрица порядка n, $\alpha^k_{i,j}$ – минор порядка k, полученный окаймлением верхнего левого углового минора порядка $k-1$ строкой i и столбцом j в матрице A, $k \leqslant \mathbf{min}(n,m)$, $i \leqslant n$, $j \leqslant m$, $\alpha^i = \alpha^i_{i,i}$ – угловой минор порядка i. Пусть все ведущие угловые миноры $\alpha^i, (1 < i < r)$ отличны от нуля.

Разложение матрицы A в произведение нижней треугольной, обратной диагональной и верхней треугольной, которое называют разложением Гаусса, имеет вид

$$A = L^{1,n}_{1,r} D^{-1} U^{1,r}_{1,m},$$

где нижняя треугольная и верхняя треугольная матрицы $L^{1,n}_{1,r}$ и $U^{1,r}_{1,m}$ имеют размер $n \times r$ и $r \times m$ соответственно:

$$L^{1,n}_{1,r} = \begin{pmatrix} a_{11} & 0 & \dots & 0 & 0 \\ a_{2,1} & a^2_{2,2} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{r-1,1} & a^2_{r-1,2} & \dots & a^{r-1}_{r-1,r-1} & 0 \\ a_{r,1} & a^2_{r,2} & \dots & a^{r-1}_{r,r-1} & a^r_{r,r} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a^2_{n,2} & \dots & a^{r-1}_{n,r-1} & a^r_{n,r} \end{pmatrix},$$

$$U^{1,r}_{1,m} = \begin{pmatrix} a_{11} & a_{1,2} & a_{1,3} & \dots & a_{1,r} & \dots & a_{1,m} \\ 0 & a^2_{2,2} & a^2_{2,3} & \dots & a^2_{2,r} & \dots & a^2_{2,m} \\ 0 & 0 & a^3_{3,3} & \dots & a^3_{3,r} & \dots & a^3_{3,m} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a^r_{r,r} & \dots & a^r_{r,m} \end{pmatrix},$$

а диагональная матрица имеет порядок r:

$$D = \mathrm{diag}\,(\alpha^1, \alpha^1\alpha^2, \cdots, \alpha^{r-2}\alpha^{r-1}, \alpha^{r-1}\alpha^r).$$

Будем полагать, что $n = m$, и дополним это тождество единичными и нулевыми блоками следующим образом:

$$A = \begin{pmatrix} L^{1,r}_{1,r} & 0 \\ L^{r+1,n}_{1,r} & I \end{pmatrix} \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} U^{1,r}_{1,r} & U^{1,r}_{r+1,n} \\ 0 & I \end{pmatrix}.$$

В таком разложениии обе треугольные матрицы имеют полный ранг n, а диагональная матрица имеет ранг r.

Разложение Гаусса в таком виде может быть обобщено на случай произвольной матрицы ранга $r \geqslant 0$. Алгоритм для общего случая неизвестен. Отметим несколько частных случаев.

Для ненулевой матрицы первого порядка имеется единственное разложение $a = aa^{-1}a$.

Для ненулевых матриц второго порядка $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ в зависимости от расположения нулевых элементов возможны следующие случаи.

Если $a \neq 0$ и $\delta = \mathbf{det}A \neq 0$, то $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & 0 \\ c & \delta \end{pmatrix} \begin{pmatrix} a^{-1} & 0 \\ 0 & (a\delta)^{-1} \end{pmatrix} \begin{pmatrix} a & b \\ 0 & \delta \end{pmatrix}$.

$$\text{Если } a \neq 0 \text{ и } \mathbf{det}A = 0, \text{ то } \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & 0 \\ c & 1 \end{pmatrix} \begin{pmatrix} a^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}.$$

$$\text{Если } b \neq 0 \text{ и } c \neq 0, \text{ то } \begin{pmatrix} 0 & b \\ c & d \end{pmatrix} = \begin{pmatrix} -\delta & 0 \\ 0 & c \end{pmatrix} \begin{pmatrix} 0 & -(c\delta)^{-1} \\ c^{-1} & 0 \end{pmatrix} \begin{pmatrix} c & d \\ 0 & \delta \end{pmatrix}.$$

$$\text{Если } b \neq 0, \text{ то } \begin{pmatrix} 0 & b \\ 0 & d \end{pmatrix} = \begin{pmatrix} b & 0 \\ d & 1 \end{pmatrix} \begin{pmatrix} 0 & b^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & b \end{pmatrix}.$$

$$\text{Если } c \neq 0, \text{ то } \begin{pmatrix} 0 & 0 \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & c \end{pmatrix} \begin{pmatrix} 0 & 0 \\ c^{-1} & 0 \end{pmatrix} \begin{pmatrix} c & d \\ 0 & 1 \end{pmatrix}.$$

$$\text{Если } d \neq 0, \text{ то } \begin{pmatrix} 0 & 0 \\ 0 & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & d^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix}.$$

Если матрица имеет полный ранг, $n = m = \operatorname{rank}(A)$ и все ведущие угловые миноры отличны от нуля, $\alpha^k \neq 0$, $1 \leqslant k \leqslant n$, то разложение Гаусса имеет вид

$$A = \begin{pmatrix} a_{11} & 0 & \dots & 0 & 0 \\ a_{2,1} & a_{2,2}^2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1} & a_{n-1,2}^2 & \dots & a_{n-1,n-1}^{n-1} & 0 \\ a_{n,1} & a_{n,2}^2 & \dots & a_{n,n-1}^{n-1} & 1 \end{pmatrix} \times$$

$$\times \begin{pmatrix} \alpha^1 & 0 & 0 & \dots & 0 \\ 0 & \alpha^1\alpha^2 & 0 & \dots & 0 \\ 0 & 0 & \alpha^2\alpha^3 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \alpha^{n-1} \end{pmatrix}^{-1} \times \begin{pmatrix} a_{11} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ 0 & a_{2,2}^2 & a_{2,3}^2 & \dots & a_{2,n}^2 \\ 0 & 0 & a_{3,3}^3 & \dots & a_{3,n}^3 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{n,n}^n \end{pmatrix}.$$

Такое разложение матрицы Д.П.Желобенко (см. [10]) предложил называть разложением Гаусса. Разложение, приведенное в [6], отличается только в том, что обе треугольные матрицы имеют единичные элементы на диагонали, дробные внедиагональные элементы, а диагональная матрица имеет вид

$$D = \operatorname{diag}(\alpha^1, \alpha^{-1}\alpha^2, \cdots, \alpha^{-r+2}\alpha^{r-1}, \alpha^{-n+1}\alpha^n).$$

Если же матрица нулевая и имеет произвольный размер, то у нее существует разложение вида $0 = I0I$. Задача состоит в том, чтобы получить разложения Гаусса для произвольной матрицы.

ЛИТЕРАТУРА

1. *Малашонок Г.И.* Матричные методы вычислений в коммутативных кольцах /монография/. Тамбов, ТГУ, 2002.

2. *Malaschonok G.I.* ParCA2: Arcitecture and Experiments // Mathematical Modeling and Computation Physics (MMCP'2009): Book of Abstracts of the International Conference (Dubna, July 7-11, 2009). - Dubna: JINR, 2009. P. 175.

3. *Malaschonok G.I.* On the project of parallel computer algebra // Tambov University Reports.Natural and Technical Sciences. V. 14, part. 4, 2009. p. 744-748.

4. *Malaschonok G.I.* Architecture of ParCA3 project // International conference Polynomial Computer Algebra. St.Petersburg, PDMI RAS, 2010. P. 44-47.

5. Алгоритмы компьютерной алгебры. Ч. 1: Учебное пособие / Г.И. Малашонок, О.Н. Переславцева, О.А. Сажнева, М.В. Старов; Федеральное агенство по образованию, Тамб. гос. ун-т им. Г.Р. Державина. Тамбов: Издательский дом ТГУ им. Г.Р. Державина, 2008. 89 с.

6. *Малашонок Г.И.* Параллельные алгоритмы компьютерной алгебры // Тезисы публичных лекций на юбилейной конференции, посвященной 75-летию Института математики, физики и информатики ТГУ им. Г.Р. Державина (23-24 ноября 2005 года). Тамбов: ИМФИ ТГУ им. Г.Р. Державина, 2005.

7. *Malaschonok G.I.* On computation of kernel of operator acting in a module // Tambov University Reports. Natural and Technical Sciences. Tambov, 2008. V. 13. Issue. 1. P. 129-131.

8. *Malaschonok G.I.* Fast matrix decomposition in parallel computer algebra // Tambov University Reports. Series: Naturaland Technical Sciences. Tambov, 2010. V. 15. Issue. 4. P. 1372-1385.

9. *Malaschonok G.I.* Fast Generalized Bruhat Decomposition // Computer Algebra in Scientific Computing, LNCS 6244, Springer, Berlin 2010. P. 194-202.

10. *Желобенко Д.П.* Компактные группы Ли и их представления.М.: Наука, 1970. С. s50-51

БЛАГОДАРНОСТИ: Работа выполнена при поддержке программы «Развитие потенциала высшей школы» (проект 2.1.1/10437).

Поступила в редакцию 12 ноября 2010 г.

UDK 519.688

CONSTRUCTIVE MATHEMATICS AND CONCEPTION OF SHORT-RANGE INTERACTION

© **G. I. Malaschonok**

Key words: constructive mathematics, conception of short-range interaction, parallel algorithms, cluster computations.

Modern directions of constructive mathematics development are discussed. This directions are connected with occurrence of multiprocessing computing systems. Requirements to algorithms and date structures which are destined for multiprocessing systems are discussed.