ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ
(Вычислительные системы)
1982 год Выпуск 94

УДК 681.3.06

ПАРАЛЛЕЛЬНО-ПОТОЧНАЯ ИНТЕРПРЕТАЦИЯ МЕТОДА ГАУССА

С.Г.Седухин

Параллельная интерпретация существующих методов решения задач требует выявления и использования пространственно-временных особенностей реализации алгоритмов. Эти особенности определяют как структурную схему вычислений, так и необходимую для эффективной реализации структуру вычислительной установки. Если функциональная структура используемой ЭВМ плохо согласована со структурной схемой алгоритма, то основной вклад во время его реализации могут внести накладные расходы на моделирование требуемой вычислительной схемы в структуре ЭВМ. Анализ параллельных методов, проведенный без сопоставления с соответствующей структурой многопроцессорных вычислительных систем (см., например, обзор [1]), не учитывает присущих при реализации алгоритмов накладных расходов. Время исполнения параллельных алгоритмов зависит не только от числа арифметических операций, но и от способа и количества пересылок данных как между процессорами системы, так и между процессором и памятью, а также от задержек, связанных с синхронизацией [2]. При этом необходимое согласование между алгоритмическими процессами и процессами электронной обработки данных в ЭВМ обеспечивается в таких коммуникационно-вычислительных структурах, которые ценой специализации в точности отвечают пространственно-временным требованиям реализации алгоритмов решения задач.

В данной работе рассматриваются пространственно-временные особенности реализации метода Гаусса, позволяющие осуществить решение системы из n линейных уравнений на сети из n линейно связанных ЭВМ за время $O(n^2/2)$, а на сети из n^2 ортогонально связанных процессоров за время $O(5n)$. Последовательная интерпретация метода, например, на ЭВМ с одной последовательностью команд над одним потоком данных, требует $O(n^3/3)$ единиц времени.

1. Пусть задана система из n линейных алгебраических уравнений

$$\left.\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = a_{1,n+1},\\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = a_{2,n+1},\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = a_{n,n+1}. \end{aligned}\right\} \qquad (1)$$

Решение системы (I) методом Гаусса (компактная схема) состоит в нахождении корней $x_n, x_{n-1}, \ldots, x_1$ в указанном порядке из уравнений

$$\left.\begin{aligned} x_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \ldots + \alpha_{1n}x_n &= \alpha_{1,n+1},\\ x_2 + \alpha_{23}x_3 + \ldots + \alpha_{2n}x_n &= \alpha_{2,n+1},\\ \ldots\ldots\ldots\ldots\ldots\ldots\\ x_n &= \alpha_{n,n+1} \end{aligned}\right\} \qquad (2)$$

после вычисления элементов следующей $n \times (n+1)$-матрицы:

$$\begin{bmatrix} \gamma_{11}\alpha_{12}\alpha_{13}\ldots\alpha_{1n}\alpha_{1,n+1}\\ \gamma_{21}\gamma_{22}\alpha_{23}\ldots\alpha_{2n}\alpha_{2,n+1}\\ \gamma_{31}\gamma_{32}\gamma_{33}\ldots\alpha_{3n}\alpha_{3,n+1}\\ \ldots\ldots\ldots\ldots\ldots\ldots\\ \gamma_{n1}\gamma_{n2}\gamma_{n3}\ldots\gamma_{nn}\alpha_{n,n+1} \end{bmatrix} \qquad (3)$$

с помощью рекуррентных формул:

$$\left.\begin{aligned} &\gamma_{ik} = a_{ik} - \sum_{j=1}^{k-1}\gamma_{ij}\cdot\alpha_{jk} \quad (i,k = 1,2,\ldots,n;\ i \geq k),\\ &\alpha_{ik} = \Big(a_{ik} - \sum_{j=1}^{i-1}\gamma_{ij}\cdot\alpha_{jk}\Big)/\gamma_{ii} \quad (i=1,2,\ldots,n;k=2,3,\ldots,n+1;i<k). \end{aligned}\right\} \qquad (4)$$

Сведение системы уравнений (I) к виду (2) по рекуррентным формулам (4) называется прямым ходом вычисления, а нахождение корней системы уравнений из (2) в порядке $x_n, x_{n-1}, \ldots, x_i$ по формуле

$$x_i = \alpha_{i,n+1} - \sum_{k=1}^{n-i} \alpha_{i,n-k+1} * x_{n-k+1} \qquad (5)$$

- обратным ходом вычисления.

2. Зависимость в последовательности определения элементов матрицы (3), вносимая рекуррентными соотношениями (4), позволяет представить весь вычислительный процесс прямого хода в виде пространственной схемы, изображенной на рис.1,а. Из рекуррентных формул (4) следует, что при $n \gg 1$ затраты на выполнение скалярного произведения составят самую значительную часть всех вычислений. При этом для определения любого (i,k) элемента матрицы (3) по соотношениям (4) ($i = 1,2,\ldots,n$; $k = 2,3,\ldots,n+1$) требуется выполнить составное арифметическое выражение

$$s \leftarrow s + \gamma_{ij} * \alpha_{jk} \qquad (6)$$

либо $\sim k$ раз (при $i \geq k$), либо $\sim i$ раз (при $i < k$). Общее число исполнений арифметического выражения (6) при вычислении всех n^2 элементов матрицы (3) оценится величиной

$$p' = \sum_{i=1}^{n} \left(\sum_{k=2}^{i} k + \sum_{k=i+1}^{n+1} i \right) = O(n^3/3).$$

Зависимость в последовательности определения корней системы уравнений (1), вносимая рекуррентным соотношением (5), требует осуществления обратного хода вычисления по схеме, изображенной на рис.1,б. Данная схема использует элементы, определенные на прямом ходе вычисления. Получение каждого значения x_i ($i=n-1, n-2,\ldots,1$) по выражению (5) требует выполнения порядка $(n-i)$ раз составного арифметического выражения типа (6). Общее число исполнений такого выражения при обратном ходе вычисления определится величиной

$$p'' = \sum_{i=1}^{n-1} (n-i) = O(n^2/2).$$

Таким образом, при больших логических размерах систем линейных уравнений ($n \gg 1$) общее число арифметических операций (выражений типа (6)), расходуемых на проведение метода Гаусса, оценится величиной $p = p'+p'' = O(n^3/3)$. Полученная величина характеризует операционную сложность алгоритма. Емкостная сложность задачи, определяющая число исходных данных, равна величине $O(n^2)$.

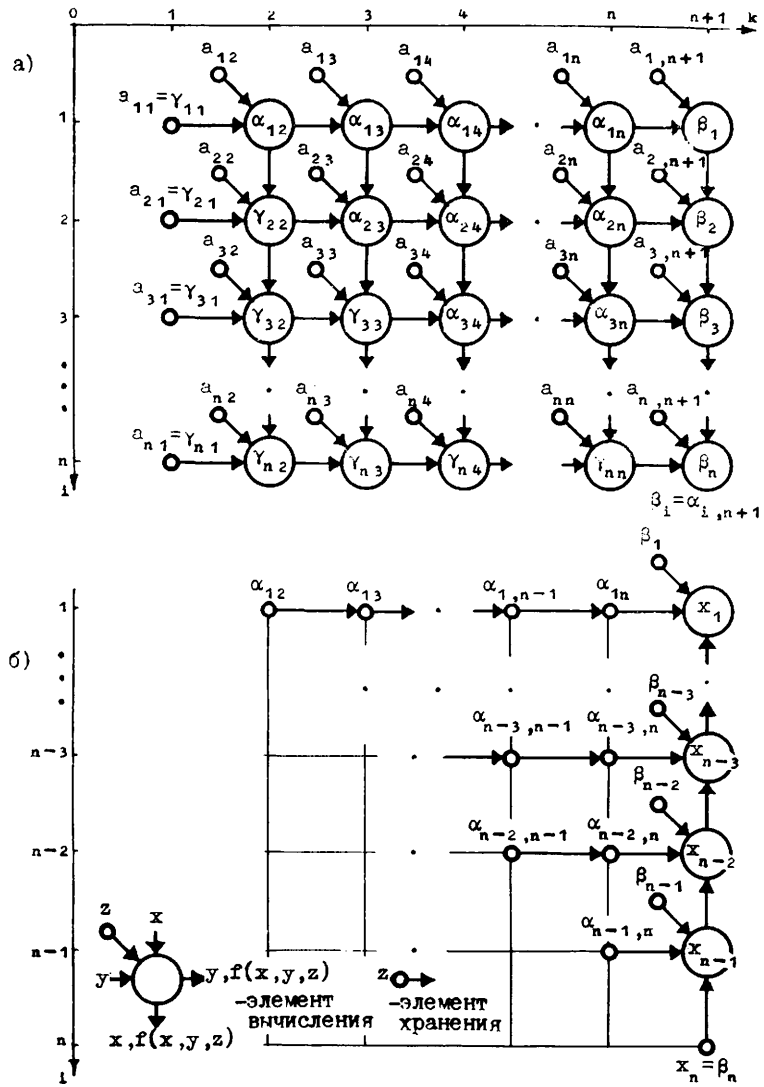

Рис.I. Структурная схема решения системы линейных уравнений по методу Гаусса.

3. Реализация метода на ЭВМ требует выработки некоторых соглашений относительно архитектуры, структуры и функционирования вычислительной машины. Во-первых, рассматриваемый метод должен быть оформлен в виде программы обработки, которая вместе с исходными данными задачи должна быть загружена в память ЭВМ. Если за единицу времени принять длительность ввода в ЭВМ одного слова, то общее время последовательного определения исходной информации в памяти ЭВМ, при достаточно больших размерах n, будет определяться емкостной сложностью задачи. При этом мы полагаем отсутствие ограничений на размер памяти ЭВМ.

Во-вторых, мы требуем обработки в ЭВМ вещественных чисел, представленных в формате с плавающей запятой. Исполнение любой арифметической операции в ЭВМ требует обращений к памяти за соответствующими операндами.

В-третьих, за единицу времени (временной шаг) мы принимаем время исполнения составного арифметического выражения типа (6), включающего операции доступа к памяти ЭВМ за операндами (равномерный весовой критерий [5]). Мы полагаем также, что пропускная способность памяти согласована с быстродействием процессора ЭВМ, исполняющим операции программы. Таким образом за единицу времени (такт работы) ЭВМ определяет промежуточный результат вычисления, существенный с точки зрения решаемой задачи. Отметим при этом, что программа обработки составного арифметического выражения типа (6) должна специфицировать не только арифметические операции и операции доступа к памяти, но и ряд других операций. Всего для получения каждого промежуточного результата требуется исполнить порядка десяти различных операций. Современные ЭВМ типа Cray 1 или AP-I20B, используя параллелизм и поточность при обработке выражения типа (6), способны выполнять все десять операций за один такт работы. При этом реализуется логическая глубина вычисления, как максимально возможное для данного выражения число одновременных операций.

При выполнении выше перечисленных соглашений временная сложность последовательной интерпретации метода Гаусса на ЭВМ с одной последовательностью команд и одним потоком данных (SISD-структура) будет равна операционной сложности алгоритма, т.е. величине $T_1 = O(n^3/3)$. Такая ЭВМ фактически последовательно сканирует от элемента к элементу вычислительную схему, изображенную на рис. I, по одному из возможных, определенных зависимостью в определении последовательности элементов, путей вычисления.

В этих условиях решение системы, например из $n = 10^4$ линейных уравнений на ЭВМ, обладающей быстродействием 1MFLOPS (миллион операций с плавающей запятой в секунду), осуществится за время порядка 100 часов. Между тем, практическое ограничение на время решения такой задачи составляет 60 секунд [6], т.е. требуется вычислительная установка с быстродействием порядка 10^4MFLOPS . Существующие супер-ЭВМ характеризуются пока быстродействием порядка 10^2MFLOPS[7].

С другой стороны, изображенная на рис.1 схема, учитывающая пространственные особенности процесса вычисления, позволяет выявить и использовать присущий методу Гаусса параллелизм, т.е. одновременность в определении элементов. Непосредственно из приведенной схемы следует, что возможно распараллеливание (расщепление) метода по одной, либо по двум координатам.

4. Расщепление метода по одной координате, например,по координате i, позволяет одновременно определять элементы i-х строк ($i = 1,2,\ldots,n$) представленной схемы [3]. Вычисление элементов i-й строки осуществляется на отдельной i-й ЭВМ, хранящей i-ю строку исходных коэффициентов $\{a_{ik}\}$ ($k = 1,2,\ldots,n+1$) и программу обработки. Прямой ход вычисления начинается с первой ($i=1$) ЭВМ, которая последовательно определяет и передает соседней ($i=2$)ЭВМ элементы α_{1k}. Каждая последующая ($i>1$) ЭВМ принимает очередной элемент от предыдущей ($i-1$)-й ЭВМ, транслирует его ($i+1$)-й ЭВМ ($i \neq n$) и производит необходимое вычисление промежуточных результатов. После обработки всех элементов, принятых от первой ЭВМ,вторая ($i=2$) ЭВМ последовательно передает соседней ($i=3$) ЭВМ вычисленные ранее элементы α_{2k} ($k=2,3,\ldots, n+1$) . Все последующие ($i>2$) ЭВМ, принимая и транслируя дальше элементы от второй ЭВМ, доопределяют необходимые элементы соответствующих строк. Этот процесс повторяется вплоть до последней ($i=n$) ЭВМ, заканчивающей прямой ход вычисления.

При принятых выше соглашениях время проведения прямого хода вычисления на системе из n линейно связанных ЭВМ определится величиной $T'_n = \max\limits_{1 \le i \le n} \{(i-1)+p'_i\}$, где $(i-1)$ - временная задержка получения i-й ЭВМ первого операнда α_{12} от первой ЭВМ системы, а p'_i - временная (операционная) сложность определения i-й ЭВМ всех n элементов соответствующей строки:

$$p'_i = \sum_{k=2}^{i} k + \sum_{k=i+1}^{n+1} i .$$

Очевидно, что временная сложность проведения прямого хода вычисления определится временем проведения обработки последней (i = n) ЭВМ системы; тогда

$$T'_n = 2n-1+ \sum_{k=2}^{n} k = O(n^2/2).$$

Обратный ход вычисления система из n ЭВМ осуществляет в противоположном прямому ходу порядке, т.е. от последней до первой ЭВМ. Проведение обратного хода ясно из рис. I,б. Временная сложность вычисления на системе всех корней определится величиной $T''_n = \max_{1 \le i \le n} \{(n-i) + p''_i\}$, где (n-i) - временная задержка получения i-й ЭВМ первого корня x_n от последней ЭВМ системы, а p''_i - временная (операционная) сложность получения в i-й ЭВМ значения x_i : $p''_i = n - i$. Очевидно, что временная сложность проведения обратного хода определится временем обработки данных первой (i = 1) ЭВМ системы; тогда $T''_n = 2n-2$.

Общее время параллельной интерпретации метода Гаусса на системе из n линейно связанных ЭВМ определится величиной $T_n = T'_n + T''_n = O(n^2/2)$. Таким образом, время реализации метода стало пропорциональным емкостной сложности задачи. Отметим, что проводимые оценки делались в предположении, что каждая ЭВМ системы обрабатывает данные в темпе их поступления от соседней ЭВМ. Достижимая в системе из n ЭВМ степень перекрытия операций (ускорение процесса вычисления) составляет величину $S_n = T_1/T_n = O(2n/3)$, а эффективность использования ЭВМ системы - величину $E_n = S_n/n = 2/3$, которая не зависит от логического размера задачи.

Полученные оценки позволяют осуществить за требуемое время решение выше упомянутой практически важной задачи на системе из $n = 10^4$ линейно связанных ЭВМ, с быстродействием каждой порядка 1MFLOPS . Успехи, достигнутые при создании дешевых, надежных и быстродействующих супермикропроцессоров на одном кристалле СБИС [8], позволяют надеяться на реальное построение вычислительных систем, содержащих указанный порядок ЭВМ.

5. Вычислительная схема, изображенная на рис.I, позволяет также расщепить метод Гаусса по координатам i и k , т.е. интерпре-

тировать метод на системе из n^2 процессоров. В этом случае определение каждого (i,k) элемента вычислительной схемы $(i = 1,2,\ldots,n;\ k = 2,3,\ldots,n+1)$ осуществляется на соответствующем (i,k) процессоре так, что сеть из n^2 ортогонально связанных процессоров моделирует параллельный (одновременный) процесс вычисления. Обработка данных начинается с первого $(1,2)$ процессора системы, который, передачей соответствующих операндов, активизирует соседние процессоры $(2,2)$ и $(1,3)$, которые, в свою очередь, запускают на обработку процессоры $(3,2)$, $(2,3)$ и $(1,4)$ и т.д., вплоть до последнего $(n, n+1)$-го процессора, который заканчивает прямой ход вычисления.

Время проведения прямого хода вычислений на системе из n^2 процессоров определится величиной

$$T'_{n^2} = \max_{\substack{1 \le i \le n \\ 2 \le k \le n+1}} \{(i+k-3) + p'_{ik}\},$$

где $(i+k-3)$ - временная задержка получения (i,k) процессором системы первых операндов γ_{i1} и α_{1k} по координатам i и k соответственно, а p'_{ik} - временная (операционная) сложность определения (i,k) процессором соответствующего (i,k) элемента:

$$p'_{ik} = \begin{cases} k, & \text{если } i \ge k; \\ i, & \text{если } i < k. \end{cases}$$

Очевидно, что временная сложность реализации прямого хода вычисления определится временем обработки данных последним $(n,n+1)$-процессором системы, т.е. $T'_{n^2} = 3n-2$.

Обратный ход вычисления выполняется вертикальной цепочкой $(i,n+1)$-х процессоров $(i = n-1,n-2,\ldots,1)$ так же, как и в системе из n ЭВМ. В данном случае горизонтальная цепочка (i,k)-х процессоров $(k = i+1,i+2,\ldots,n)$ является памятью, поставляющей операнды α_{ik} в обрабатывающий $(i,n+1)$-й процессор системы. Таким образом, общее время параллельной интерпретации метода Гаусса на системе из n^2 ортогонально связанных процессоров определится величиной $T_{n^2} = 5n-4$.

Очевидно, что вся сеть процессоров выполнит $O(n^3/3)$ арифметических выражений типа (6), т.е. обеспечивается степень совмещения операций (ускорение), равная величине $S_{n^2} = T_1/T_{n^2} = O(n^2/15)$. Эффективность обработки данных равна величине $E_{n^2} = S_{n^2}/n^2 = 1/15$, которая не зависит от размера задачи n.

Отметим, что так как время реализации метода Гаусса на системе из n^2 процессоров существенно меньше емкостной сложности задачи, то процедура ввода исходных данных в систему должна быть также распараллелена. В противном случае, т.е. при последовательной загрузке исходных операндов, время ввода может определить общее время решения задачи и свести на нет выигрыш от параллельной обработки данных на системе процессоров. Параллельный ввод данных в систему может быть осуществлен либо по строкам, либо по столбцам, либо по диагоналям исходной матрицы коэффициентов так, чтобы общее время ввода было одного порядка с временной сложностью реализации метода. Кроме того, практическое использование метода на системе из n^2 процессоров усугубляется необходимостью определения (ввода) программы обработки в каждой процессор системы. В данном случае размер программы будет заведомо больше, чем размер исходных данных, хранящихся в каждом процессоре. Тем не менее, при разумном учете этих ограничений, рассмотренная параллельно-поточная интерпретация метода может быть практически использована.

Полученные временные оценки реализации метода Гаусса позволяют заключить, что сеть из 10^8 ортогонально связанных процессоров, с быстродействием каждого не ниже 10^3 операций с плавающей запятой в секунду, осуществит решение выше упомянутой системы из 10^4 уравнений за требуемое время.

Укажем на немаловажную особенность параллельно-поточной реализации метода Гаусса. Так как всякий раз одновременный процесс вычисления возможен только для части элементов матрицы (3) и для их определения требуется исполнение различных (по длине и по составу) последовательностей команд, то эффективная параллельная реализация метода возможна только на вычислительных системах с несколькими потоками команд над многими потоками данных (MIMD-структура). Реализация метода на вычислительной системе с одним (общим) потоком команд над многими потоками данных (SIMD-структура) будет характеризоваться низкой скоростью и эффективностью.

6. Отметим также некоторые характерные особенности параллельно-поточного режима работы MIMD-систем.

I. Процесс ввода исходных данных (равно как и вывода) в вычислители (ЭВМ или процессоры) системы отличается от процесса ввода/вывода информации в одну общую память. При последовательной загрузке исходных данных это отличие будет определять затраты, кратные числу вычислителей в системе.

2. Параллельно-поточная реализация алгоритмов в MIMD-системе предполагает, в отличие от SIMD-системы, наличие программы обработки в каждом вычислителе. Последовательный ввод программы в вычислители,так же как и в первом случае, будет сопровождаться ростом затрат, пропорциональным числу вычислителей в системе.

3. Так как каждый вычислитель системы имеет свою память для хранения данных и программ, то возможно осуществлять параллельный ввод/вывод информации и эффективно совмещать его с вычислениями. При этом увеличение числа вычислителей в системе будет обеспечивать не только пропорциональный рост потенциально возможного быстродействия, но и эквивалентное возрастание возможной пропускной способности системы к вводу/выводу. В условиях решения конкретной задачи данное свойство системы может быть использовано для обеспечения необходимого соответствия между скоростями обработки и ввода/вывода информации.

4. Получение предельной скорости решения некоторой задачи требует числа вычислителей в системе, равного логическому размеру задачи. Так как логический размер реальных задач является величиной произвольной, то для решения конкретной задачи необходимо будет выделять и настраивать на требуемую структурную схему соответствующее число вычислителей. Последнее также вносит затраты, характерные только для вычислительных систем с программируемой структурой.

5. Параллельно-поточный режим работы системы и присущая инерционность вычислителей вызывает характерные для магистрального способа обработки данных переходные процессы. Существование переходных процессов снижает потенциально возможную эффективность проведения вычислений и увеличивает общее время решения задачи на системе. Особенно это сказывается при реализации алгоритмов, распараллеленных по двум и более координатам.

6. Параллельно-поточная реализация методов является осуществлением одного из возможных порядков выполнения операций данного алгоритма. Известно [9], что различные последовательности выполнения операций, эквивалентные в математическом смысле, могут приводить к различным результатам, в особенности с точки зрения численной устойчивости. Необходимо оценить влияние ошибок при параллельно-поточной реализации численных методов.

Анализ перечисленных особенностей является предметом дальнейших исследований. В заключение отметим, что так как реальные

задачи характеризуются, с одной стороны, различными логическими размерами, а, с другой стороны, их решение осуществляется по различным алгоритмам, то есть основания утверждать, что только вычислительные системы с функциональной структурой, настраиваемой под конкретный размер задачи и под конкретную структурную схему вычисления, в состоянии обеспечить оптимальность и эффективность применительно к решению широкого класса задач.

Литература

1. ФАДДЕЕВА В.Н., ФАДДЕЕВ Д.К. Параллельные вычисления в линейной алгебре. -Кибернетика, 1982, № 3, с. 18-31, 44.

2. МИШИН А.И., СЕДУХИН С.Г. Однородные вычислительные системы и параллельные вычисления. -АВТ, 1981, №1, с. 20-24.

3. МИШИН А.И., СЕДУХИН С.Г. Вычислительные системы и параллельные вычисления с локальными взаимодействиями. - В кн.: Математическое обеспечение вычислительных систем (Вычислительные системы, вып. 78). Новосибирск, 1979, с.90-103.

4. KUNG H.T., LEISERSON C.E. Systolic arrays (for VLSI). - Proc.Symp.Sparse Matrix Computations and Applications, Society for Industrial and Applied Mathematics,1979,p.256-282.

5. АХО А., ХОПКРОФТ Дж., УЛЬМАН Дж. Построение и анализ вычислительных алгоритмов. -М.: Мир, 1979. - 536 с.

6. Методы алгоритмизации непрерывных производственных процессов /Под ред. В.В.Иванова. -М.: Наука, 1975.

7. COZDOROWICKI E.W., THEIS D.I. Second Generation of Vector Supercomputers.- Computer,1980,N 11,p.71-83.

8. Новые 32-разрядные микропроцессоры. - Радиоэлектроника за рубежом. Вып. 25(945), 1981, с.1-4.

9. ВОЕВОДИН В.В. Вычислительные основы линейной алгебры.-М.: Наука, 1977. -304 с.

Поступила в ред.-изд.отд.
6 августа 1982 года