

ВЫБОР МЕТОДА И ГРУППОВЫХ ОПЕРАЦИЙ ДЛЯ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

Г.Г.Иванов, В.П.Толстьев

Развитие методов автоматического программирования привело к постановке задачи создания крупноблочных языков программирования, в которых в качестве базового элемента в записи вычислительных планов использовались бы не отдельные числа (скаляры), а более крупные единицы информации - величины (например, векторы, матрицы, списки); соответственно, над этими величинами определяются операции языка [1].

На современном этапе развития вычислительной техники возможно создание устройств для обработки более или менее крупных массивов информации. Возрастающая сложность решаемых задач требует существенного повышения быстродействия, чем и обусловлена необходимость создания подобных устройств.

Широкий класс задач, решаемых на ЦВМ, представляют задачи, связанные с решением систем линейных уравнений (СЛУ). В данной работе рассматриваются известные методы решения СЛУ и исследуются возможности наиболее рационального использования в них укрупненных операций, т.е. операций над матрицами и векторами (групповых операций). По нашему мнению, сравнение и выбор методов следует вести с учетом следующих требований:

- 1) универсальность, т.е. метод должен быть применим к СЛУ с матрицей произвольного вида;
- 2) в качестве начальной должна использоваться исходная матрица коэффициентов без приведения к специальному виду;
- 3) регулярность вычислительной схемы;
- 4) минимальная длина шага метода;
- 5) минимальная система групповых операций.

Сравнение методов ведется по результатам анализа одного шага алгоритма.

Удобным средством для описания алгоритмов с применением групповых операций является язык APL, [2]. В данной работе язык APL используется с некоторыми изменениями.

1. Матрицы обозначаются прописными буквами - A, B, \dots , i -й вектор-строка матрицы A - a_i , i -й вектор-столбец матрицы A - $a_{:i}$, (i, j) -й элемент матрицы A - a_{ij} .

2. Введен оператор распространения.

Распространение скаляра.

$$a) \text{ по строке: } \underline{b}_i \leftarrow \Theta(n) a \iff \underline{b}_i \leftarrow (\underline{a}, \underline{a}, \dots, \underline{a}),$$

$$b) \text{ по столбцу: } \underline{b}_{:i} \leftarrow \Phi(n) a \iff \underline{b}_{:i} \leftarrow (\underline{a}, \underline{a}, \dots, \underline{a})$$

Распространение вектора.

$$a) \text{ по строкам: } \underline{B} \leftarrow \Theta(n) \underline{a}_{:i} \iff \underline{b}_{:j} \leftarrow \underline{a}_{:i}, (j=1+n),$$

$$b) \text{ по столбцам: } \underline{B} \leftarrow \Phi(n) \underline{a}_i \iff \underline{b}_j \leftarrow \underline{a}_i, (j=1+n).$$

Наиболее известным приемом решения СЛУ является метод Гаусса, относящийся к группе методов последовательного исключения неизвестных. Исходная информация задается в виде присоединенной матрицы A размерности $n \times (n+1)$. Все переменные, используемые на k -м шаге алгоритма, будут помечаться верхним индексом k . Так например, исходная матрица k -го шага, т.е. матрица, оставшаяся после исключения $k-1$ неизвестных, обозначается A^k и имеет размерность $(n-k+1) \times (n-k+2)$, ее элементы обозначаются a_{ij}^k .

k -й шаг прямого хода алгоритма Гаусса:

$$\langle 1 \rangle \underline{b}_1^k \leftarrow \Theta(n-k+2) a_{11}^k$$

$$\langle 2 \rangle \underline{c}_1^k \leftarrow \underline{a}_1^k \div \underline{b}_1^k$$

$$\langle 3 \rangle \underline{m}_k \leftarrow \underline{c}_1^k$$

$$\langle 4 \rangle \underline{D}^k \leftarrow \Phi(n-k) \underline{c}_1^k$$

$$\langle 5 \rangle \underline{q}_{:1}^k \leftarrow \omega^{n-k}(n-k+1) / \underline{a}_{:1}^k$$

$$\langle 6 \rangle \underline{F}^k \leftarrow \Theta(n-k+2) \underline{q}_{:1}^k$$

$$\langle 7 \rangle \underline{H}^k \leftarrow \underline{D}^k \cdot \underline{F}^k$$

$$\langle 8 \rangle \underline{J}^k \leftarrow \omega^{n-k}(n-k+1) / \underline{A}^k$$

$$\langle 9 \rangle \underline{L}^k \leftarrow \underline{J}^k - \underline{H}^k$$

$$\langle 10 \rangle \underline{A}^{k+1} \leftarrow \omega^{n-k+1}(n-k+2) / \underline{L}^k, \text{ где } k=(1+n)$$

Результатом выполнения n шагов прямого хода является верхняя треугольная матрица M размерности $n \times (n+1)$, строки которой формируются операцией $\langle 3 \rangle$ прямого хода. Матрица M является исходной для обратного хода, l -й шаг которого записывается следующим образом:

$$\langle 1 \rangle \underline{b}_{11}^l \leftarrow \underline{\varepsilon}^l(n) / \underline{m}_{:1}$$

$$\langle 2 \rangle \underline{c}_{:1}^l \leftarrow \Phi(l-1) \underline{b}_{11}^l$$

$$\langle 3 \rangle \underline{d}_{:1}^l \leftarrow \omega^{l-1}(l) / \underline{m}_{:1-l}$$

$$\langle 4 \rangle \underline{f}_{:1}^l \leftarrow \underline{c}_{:1}^l \cdot \underline{d}_{:1}^l$$

$$\langle 5 \rangle \underline{m}_{:1-l} \leftarrow \underline{m}_{:1-l} - \underline{f}_{:1}^l, \text{ где } l=(n+2)$$

Вектор решений, сформированный в результате обратного хода, находится в первом столбце $\underline{m}_{:1}$ матрицы M .

Отметим, что вычислительная схема метода Гаусса не является регулярной, т.к. алгоритм состоит из двух последовательных итераций различного типа.

Метод оптимального исключения [3] состоит в том, что на k -м шаге: 1) исключаются $k-1$ неизвестных k -й строки, 2) исключаются $k-1$ неизвестных k -го столбца в строках $1+k-1$. Метод характеризуется регулярностью вычислительной схемы, однако рассмотрение его показывает, что пункт 2) по своей структуре сходен с прямым ходом метода Гаусса, а пункт 1) - гораздо сложнее обратного хода метода Гаусса, причем дополнительно требуется введение операции редуктивного вычитания строк матрицы; выполнение этой операции требует больших затрат времени. Поэтому с точки зрения поставленных нами требований указанный метод по своим характеристикам хуже метода Гаусса.

К методам последовательного исключения относится также метод Жордана. Схема метода является регулярной. Последовательность операций k -го шага алгоритма может быть записана следующим образом:

$$\langle 1 \rangle \underline{b}_{11}^k \leftarrow \underline{\varepsilon}^k(n) / \underline{a}_{:1}^k$$

$$\langle 2 \rangle \underline{c}_{:1}^k \leftarrow \Theta(n-k+2) \underline{b}_{11}^k$$

$$\langle 3 \rangle \underline{a}_{:k}^k \leftarrow \underline{a}_{:k}^k \div \underline{c}_{:1}^k$$

$$\langle 4 \rangle \underline{D}^k \leftarrow \Phi(n) \underline{a}_k^k.$$

$$\langle 5 \rangle \underline{q}_{\cdot,1}^k \leftarrow \underline{a}_{\cdot,1}^k.$$

$$\langle 6 \rangle \underline{q}_{k,1}^k \leftarrow 0$$

$$\langle 7 \rangle \underline{F}^k \leftarrow \Theta(n-k+2) \underline{q}_{\cdot,1}^k.$$

$$\langle 8 \rangle \underline{H}^k \leftarrow \underline{D}^k \cdot \underline{F}^k$$

$$\langle 9 \rangle \underline{J}^k \leftarrow \underline{A}^k - \underline{H}^k$$

$$\langle 10 \rangle \underline{A}^{k+1} \leftarrow \underline{\omega}^{n-k+1} (n-k+2) / \underline{J}^k, \text{ где } k = (1 \div n).$$

После выполнения n шагов алгоритма получается столбцовая матрица \underline{A}^n , которая представляет собой вектор решений СЛУ.

Сравнительная характеристика методов Гаусса и Жордана по набору групповых операций и частоте их вхождений в один шаг алгоритма приведена в таблице.

Т а б л и ц а

Наименование групповой операции	Число вхождений операций в I шаг	
	Метод Гаусса	Метод Жордана
Распространение скаляра	2	1
Распространение вектора	2	2
Сжатие вектора	3	1
Сжатие матрицы	2	1
Деление векторов	1	1
Разность векторов	1	-
Разность матриц	1	1
Произведение векторов ^{*)}	1	-
Произведение матриц ^{*)}	1	1
Общее число операций	14	8

Метод отражений [3], основанный на унитарном преобразовании исходной системы к системе с треугольной матрицей имеет сложную нерегулярную вычислительную схему. Этим же недостатком обладает схема Халещкого [4], т.к. она состоит в том, что

*) Произведение - покомпонентное.

исходная матрица коэффициентов \underline{A} представляется в виде произведения нижней треугольной матрицы \underline{B} на верхнюю треугольную матрицу \underline{C} : $\underline{A} = \underline{B} \cdot \underline{C}$, и решение исходной системы $\underline{A} \underline{x} = \underline{b}$ находится из цепи уравнений $\underline{B} \underline{y} = \underline{b}$, $\underline{C} \underline{x} = \underline{y}$.

Метод квадратного корня [3] не является универсальным, т.к. применим только для решения СЛУ с эрмитовой матрицей.

Специально не рассматривались итерационные методы, т.к. во-первых, для обеспечения сходимости линейную систему необходимо преобразовывать к специальному виду [4], во-вторых, эти методы требуют введения медленной операции редуцитивного сложения.

Таким образом, из рассмотренных методов поставленным выше требованиям в наибольшей степени удовлетворяет метод Жордана. Этот метод является универсальным, отличается регулярностью и простотой вычислительной схемы, шаг алгоритма имеет небольшую длину, и набор операций, необходимых для его выполнения, приведенный в таблице, включает 7 сравнительно быстрых, простых операций. В выделенном наборе наиболее массовыми являются операции покомпонентного деления векторов, вычитания и умножения матриц. Именно эти операции занимают большую часть времени решения СЛУ на ЦВМ. Поскольку эти операции над компонентами сложных переменных можно проводить независимо друг от друга, то введение некоторого параллелизма вычислений приведет к значительному сокращению времени решения СЛУ.

Л и т е р а т у р а

1. КАНТОРОВИЧ Л.В. Перспективы работы в области автоматизации программирования на базе крупноблочной системы. - Труды Матем. ин-та АН СССР, 1968, 96, 5-15.
2. Iverson K.E. A Programming Language. Wiley, New York, 1962.
3. ВОБВОДИН В.В. Численные методы алгебры. Наука, 1966.
4. ДЕМИДОВИЧ Б.П., МАРОН И.А. Основы вычислительной математики. Наука, 1966.