В. П. Майер, Алгоритмы прогонки для $(2m+1)$-диагональной матрицы, *Ж. вычисл. матем. и матем. физ.*, 1984, том 24, номер 5, 627–632

ЖУРНАЛ
ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ

Том 24, 1984 № 5

УДК 519.614

АЛГОРИТМЫ ПРОГОНКИ ДЛЯ $(2m+1)$-ДИАГОНАЛЬНОЙ МАТРИЦЫ

МАЙЕР В. П.

(Тюмень)

Описываются алгоритмы решения систем линейных алгебраических уравнений с $(2m+1)$-диагональными матрицами, являющиеся обобщением известного метода прогонки. Доказывается корректность алгоритмов и их устойчивость к погрешностям вычислений. Приводятся оценки количества выполняемых операций и величины требуемой памяти ЭВМ для промежуточных переменных и массивов.

Введение

Системы линейных алгебраических уравнений с $(2m+1)$-диагональными матрицами встречаются при разностной аппроксимации краевых задач для обыкновенных дифференциальных уравнений. Количество диагоналей m, расположенных над и под главной диагональю и содержащих все ненулевые элементы этой матрицы, зависит как от порядка дифференциального уравнения, так и от порядка аппроксимации соответствующего дифференциального оператора.

Подобные системы уравнений можно решать любыми известными способами, не учитывающими специальной структуры матрицы. При $m \ll n$, где n — количество неизвестных, эти методы оказываются неэффективными, а при больших n — и непосильными даже для современных ЭВМ.

Специальные методы решения таких систем, разработанные для случая $m=1$ (известный метод прогонки) и для $m=2$ (его обобщение на 5-диагональные матрицы [1]), гораздо более экономичны в смысле количества выполняемых операций сложения и умножения, необходимых для получения результата, а для большинства матриц, представляющих практический интерес, устойчивы к погрешностям вычисления [1], [2].

Описываемые в работе алгоритмы являются обобщением метода прогонки для случая $(2m+1)$-диагональной матрицы и обладают всеми его достоинствами.

§ 1. Вывод расчетных формул

Запишем систему уравнений с $(2m+1)$-диагональной матрицей в виде

$$\sum_{j=-j_2}^{j_1} A_{i,j}x_{i+j}=y_i, \qquad i=1,2,\dots,n, \tag{1.1}$$

где $0<m\leqslant n$. Здесь и далее $j_1=\min(n-i,\ m)$, $j_2=\min(i-1,\ m)$.

Для упрощения вывода расчетных формул расширим все диагонали матрицы до одинаковой длины, положив

$$A_{i,j}=0, \text{ если } i+j<1 \text{ или } i+j>n. \tag{1.2}$$

Применяя метод исключения Гаусса, обнаруживаем, что искомые неизвестные x_i, $i=1, 2, \ldots, n$, связаны зависимостями вида

$$(1.3)\qquad x_i=\beta_i+\sum_{k=1}^{m}\alpha_{i,k}x_{i+k}, \qquad i=1,2,\ldots,n,$$

где дополнительные неизвестные x_{n+k}, $k=1, 2, \ldots, m$, можно считать равными нулю.

Из (1.3) следует

$$(1.4)\qquad x_{i-j}=s_{ij}+\sum_{l=0}^{m-1} r_{ijl}x_{i+l}, \qquad j=1,2,\ldots,i-1, \quad i=2,3,\ldots,n.$$

Действительно, (1.4) удовлетворяется при $j=1$, если

$$s_{i1}=\beta_{i-1}, \qquad r_{i1l}=\begin{cases}\alpha_{i-1,l+1}, & l=0,1,\ldots,m-1,\\ 0, & l=m.\end{cases}$$

В случае $j>1$ выведем рекуррентные соотношения для s_{ij} и r_{ijl}. Из (1.3) и (1.4) находим

$$x_{i-j}=x_{(i-1)-(j-1)}=s_{i-1,j-1}+\sum_{l=0}^{m-1} r_{i-1,j-1,l}x_{i-1+l}=$$

$$=s_{i-1,j-1}+r_{i-1,j-1,0}x_{i-1}+\sum_{l=1}^{m-1} r_{i-1,j-1,l}x_{i-1+l}=$$

$$=s_{i-1,j-1}+r_{i-1,j-1,0}\left(\beta_{i-1}+\sum_{l=0}^{m-1}\alpha_{i-1,l+1}x_{i+l}\right)+\sum_{l=0}^{m-2} r_{i-1,j-1,l+1}x_{i+l}.$$

Отсюда

$$s_{ij}=s_{i-1,\,j-1}+\beta_{i-1}r_{i-1,\,j-1,\,0},$$

$$r_{ijl}=\begin{cases}r_{i-1,j-1,l+1}+\alpha_{i-1,l+1}r_{i-1,j-1,0}, & l=0,1,\ldots,m-1,\\ 0, & l=m,\end{cases}$$

$$j=2, 3, \ldots, i-1, \qquad i=2, 3, \ldots, n.$$

Подставляя (1.4) в (1.1) и учитывая (1.2), получаем

$$\sum_{j=1}^{m}A_{i,-j}\left(s_{ij}+\sum_{l=0}^{m-1}r_{ijl}x_{i+l}\right)+A_{i0}x_i+\sum_{l=1}^{m}A_{il}x_{i+l}=y_i, \qquad i=1,2,\ldots,n.$$

Отсюда

$$\Delta_i x_i=y_i-\sum_{j=1}^{m}A_{i,-j}s_{ij}-\sum_{l=1}^{m-1}\left(A_{il}+\sum_{j=1}^{m}A_{i,-j}r_{ijl}\right)x_{i+l}-A_{im}x_{i+m},$$

$$\Delta_i=A_{i0}+\sum_{j=1}^{m}A_{i,-j}r_{ij0}, \qquad i=1,2,\ldots,n.$$

Сопоставляя с (1.3), получаем

$$\beta_i=\frac{1}{\Delta_i}\left(y_i-\sum_{j=1}^{m}A_{i,-j}s_{ij}\right),$$

$$\alpha_{il}=\begin{cases} -\dfrac{1}{\Delta_i}\left(A_{il}+\sum\limits_{j=1}^{m}A_{i,-j}r_{ijl}\right), & l=1,2,\dots,m-1,\\ -\dfrac{A_{im}}{\Delta_i}, & l=m.\end{cases}$$

Учитывая теперь (1.2) и (1.3), получаем следующие расчетные формулы для первого алгоритма:

$$\Delta_i=A_{i0}+\sum_{j=1}^{j_2}A_{i,-j}r_{ij0}, \tag{1.5}$$

$$\beta_i=\left(y_i-\sum_{j=1}^{j_2}A_{i,-j}s_{ij}\right)\Delta_i^{-1}, \tag{1.6}$$

$$\alpha_{il}=-\left(A_{il}+\sum_{j=1}^{j_2}A_{i,-j}r_{ijl}\right)\Delta_i^{-1},\qquad l=1,2,\dots,j_1, \tag{1.7}$$

$$s_{ij}=\begin{cases}\beta_{i-1}, & j=1,\\ s_{i-1,j-1}+r_{i-1,j-1,0}\beta_{i-1}, & j=2,3,\dots,j_2,\end{cases} \tag{1.8}$$

$$r_{ijl}=\begin{cases}\alpha_{i-1,l+1}, & j=1,\quad l=0,1,\dots,j_3,\\ r_{i-1,j-1,l+1}+r_{i-1,j-1,0}\alpha_{i-1,l+1}, & j=2,3,\dots,j_2,\ l=0,1,\dots,j_4,\\ 0,\ l=m, & j_1=m,\ j=1,2,\dots,j_2\end{cases} \tag{1.9}$$

(здесь и далее $j_3=\min(n-i,\ m-1)$, $j_4=\min(i-1,\ m-1)$),

$$x_i=\beta_i+\sum_{l=1}^{j_1}\alpha_{il}x_{i+l},\qquad i=1,2,\dots,n. \tag{1.10}$$

Второй алгоритм (более экономичный по количеству необходимых арифметических операций, чем первый) можно получить, если обратить внимание на то, что в формулах (1.5)–(1.7) участвуют не r_{ijl} и s_{ij} сами по себе, а их свертки с элементами матрицы системы (1.1).

Рассмотрим следующие выражения:

$$p_{ik}=\sum_{j=1}^{j_5}A_{i+k,-j-k}s_{ij},\qquad q_{ilk}=\sum_{j=1}^{j_5}A_{i+k,-j-k}r_{ijl}, \tag{1.11}$$

$$l,k=0,1,\dots,j_1,\qquad i=2,3,\dots,n.$$

Здесь и далее $j_5=\min(i-1,\ m-k)$.

Видно, что $p_{ik}=0$ и $q_{ilk}=0$ при $i=1$ для всех допустимых l, k.

Пользуясь формулами (1.8) и (1.9), получаем следующие рекуррентные выражения:

$$p_{ik}=\begin{cases}p_{i-k,k+1}+\beta_{i-1}(A_{i+k,-k-1}+q_{i-1,0,k+1}), & k=0,1,\dots,j_3,\\ 0, & k=m=j_1,\end{cases} \tag{1.12а}$$

$$q_{ilk}=\begin{cases}q_{i-1,l+1,k+1}+\alpha_{i-1,l+1}(A_{i+k,-k-1}+q_{i-1,0,k+1}), & k=0,1,\dots\ j_3,\\ 0, & l,k=m=j_1.\end{cases} \tag{1.12б}$$

Исходя из формул (1.11) равенства (1.5)–(1.7) можно записать в виде

$$\Delta_i=A_{i0}+q_{i00}, \tag{1.13}$$

$$\beta_i=(y_i-p_{i0})/\Delta_i, \tag{1.14}$$

$$\alpha_{il}=-(A_{il}+q_{il0})/\Delta_i,\qquad l=1,\ 2,\dots,\ j_1,\quad i=1,\ 2,\dots,\ n. \tag{1.15}$$

§ 2. Алгоритмы решения системы (1.1)

Схема получения решения системы (1.1) такая же, как в обычной прогонке. Сначала определяются все β_i, α_{il} по формулам (1.5)–(1.9) (первый алгоритм) или по формулам (1.12)–(1.15) (второй алгоритм) для всех $i=1, 2, \ldots, n$. Затем по найденным β_i и α_{il} находятся x_i по формуле (1.10), где i меняется в обратном порядке, т. е. $i=n, \ldots, 2, 1$.

Более подробно этап вычисления α_{il}, β_i выглядит следующим образом. Для $i=1$ по формулам (1.5)–(1.7) находим Δ_1, β_1, α_{1l}, $l=1, 2, \ldots, j_1$. Начиная с $i=2$ можно определить α_{il} и β_i двумя способами.

Алгоритм 1. По формулам (1.8) и (1.9) находим s_{ij} и r_{ijl}, $l=0, 1, \ldots, j_1$, $j=j_2, j_2-1, \ldots, 1$. Благодаря такому изменению индекса j можно размещать массивы s_{ij} и r_{ijl} на месте массивов $s_{i-1,\,j}$, $r_{i-1,\,j,\,l}$. После этого по формулам (1.5), (1.6) находятся β_i, а если $i<n$ – то и α_{il}, $l=1, 2, \ldots, j_1$, по формуле (1.7).

Алгоритм 2. По формулам (1.12) определяем p_{ik} и q_{ilk} для l, $k=0, 1, \ldots, j_1$. Далее по формулам (1.13), (1.14) находим Δ_i, β_i, а если $i<n$ – то и α_{il}, $l=1, 2, \ldots, j_1$, по формуле (1.15).

§ 3. Обоснование алгоритмов

Корректность алгоритмов нуждается в обосновании, поскольку они предполагают операцию деления. Кроме того, необходимо исследование алгоритмов на устойчивость к погрешностям вычислений.

В следующей теореме формулируются достаточные условия, при которых алгоритмы устойчивы и все $\Delta_i \neq 0$, $i=1, 2, \ldots, n$.

Теорема. *Пусть выполнены условия*

$$(3.1)\qquad |A_{i0}| \geqslant \sum_{j=1}^{j_2} |A_{i,-j}| + \sum_{j=1}^{j_1} |A_{ij}|, \qquad i=1, 2, \ldots, n,$$

причем неравенство должно быть строгим либо при $i=1$, *либо при* $i=2$. *Тогда для всех* $i=1, 2, \ldots$ *имеем* $\Delta_i \neq 0$ *и*

$$(3.2)\qquad \sum_{l=1}^{j_1} |\alpha_{il}| \leqslant 1.$$

Прежде чем доказывать теорему, докажем следующие леммы.

Лемма 1. *Если при некотором* $i<n$ *справедливы условия* (3.2), *а также*

$$(3.3)\qquad \sum_{l=0}^{j_3} |r_{ijl}| \leqslant 1, \qquad j=1, 2, \ldots, j_2,$$

то (3.3) *остаются справедливыми при замене* i *на* $i+1$. *Причем в* (3.3) *будут выполняться строгие неравенства, если строгим было неравенство в* (3.2).

В самом деле, если $j=1$, то

$$\sum_{l=0}^{j_6} |r_{i+1,j,l}| = \sum_{l=0}^{j_1-1} |\alpha_{i,l+1}| = \sum_{l=1}^{j_1} |\alpha_{il}|.$$

Здесь и далее $j_6 = \min(n-i-1, m-1)$.

Если $j>1$, то

$$\sum_{l=0}^{j_6}|r_{i+1,j,l}|\leqslant\sum_{l=0}^{j_6}|r_{i,j-1,l+1}|+|r_{i,j-1,0}|\sum_{l=0}^{j_6}|\alpha_{i,l+1}|=$$

$$=\sum_{l=1}^{j_1}|r_{i,j-1,l}|+|r_{i,j-1,0}|\sum_{l=1}^{j_1}|\alpha_{il}|\leqslant\sum_{l=0}^{j_3}|r_{i,j-1,l}|.$$

Лемма 2. *Если при некотором $i>1$, $i<n$ справедливы неравенства (3.1) и (3.3), то для этого же i справедливо*

$$\sum_{l=1}^{j_1}\left|A_{il}+\sum_{j=1}^{j_2}A_{i,-j}r_{ijl}\right|\leqslant|\Delta_i|. \tag{3.4}$$

Причем в (3.4) будет строгое неравенство, если оно было строгим либо в (3.1), либо в (3.3).

В самом деле,

$$\sum_{l=1}^{j_1}\left|A_{il}+\sum_{j=1}^{j_2}A_{i,-j}r_{ijl}\right|\leqslant\sum_{l=1}^{j_1}|A_{il}|+\sum_{j=1}|A_{i,-j}|\times$$

$$\times\sum_{l=1}^{j_1}|r_{ijl}|\leqslant\sum_{l=1}^{j_1}|A_{il}|+\sum_{j=1}^{j_2}|A_{i,-j}|(1-|r_{ij0}|)\leqslant|A_{i0}|-$$

$$-\sum_{j=1}^{j_2}|A_{i,-j}||r_{ij0}|\leqslant\left|A_{i0}+\sum_{j=1}^{j_2}A_{i,-j}r_{ij0}\right|=|\Delta_i|.$$

Доказательство теоремы теперь легко провести по индукции. Действительно, при $i=1$ справедливость утверждений теоремы видна непосредственно из формул (1.5), (1.7) и условия (3.1). Если условие (3.1) выполняется со строгим неравенством при $i=1$, то в (3.2) также будет строгое неравенство. Далее, пользуясь формулами (1.9), убеждаемся в справедливости неравенств (3.3) при $i=2$. Причем последние будут строгими, если строгим было неравенство (3.2) при $i=1$. Пользуясь теперь леммой 2 и условием (3.1), убеждаемся в выполнении строгого неравенства (3.4) при $i=2$, из которого следует, что $\Delta_i\neq0$, а также выполнение (3.2) в форме строгого неравенства.

Допустим, что утверждения теоремы справедливы при некотором $k=$ $=i-1$, причем в (3.2) — строгое неравенство. Докажем, что утверждения теоремы останутся справедливыми и при $k=i$, причем (3.2) — в форме строгого неравенства.

Согласно лемме 1, строгие неравенства (3.2) и (3.3) при $k=i-1$ влекут выполнение строгих неравенств (3.3) при $k=i$. Тогда при $i<n$ из леммы 2 следует $\Delta_i\neq0$ и, кроме этого, строгое неравенство (3.2) при $k=i$. Если $i=n$, то

$$|\Delta_n|\geqslant|A_{n0}|-\sum_{j=1}^{j_7}|A_{n,-j}|\,|r_{nj0}|>|A_{n0}|-\sum_{j=1}^{j_7}|A_{n,-j}|\geqslant0,$$

где $j_7=\min(n-1, m)$, чем и завершается доказательство теоремы.

Замечание. Из утверждения леммы 2 видно, что если $\Delta_i=0$, то числители формул (1.7) и (1.15) также равны нулю. Из этого следует, что алгоритмы могут быть использованы и для решения совместных вырожденных систем уравнений вида (1.1), диагонали которых подчиняются условию (3.1) без требования выполнения строгого неравенства хотя бы в одном уравнении. Для этого достаточно при нулевом Δ_i считать результатом операции деления любое число в формулах (1.6) и (1.14) и нуль в формулах (1.7) и (1.15). Обобщенные таким образом алгоритмы позволят найти некоторое частное решение системы (1.1), если она вырожденная. Индикатором несовместности системы в этих алгоритмах будет ненулевой числитель при нулевом знаменателе в формулах (1.6) и (1.14).

§ 4. Оценки количества операций и памяти

Анализируя формулы (1.5)–(1.10) и (1.12)–(1.15), можно увидеть, что общее количество операций сложения, вычитания, умножения и деления не больше $C_i nm^2$, $i=1, 2$, где C_i — некоторые константы, не зависящие от n и m, причем $C_1 > C_2$. Здесь i — номер алгоритма. Отсюда следует, что время вычисления по алгоритмам 1 и 2 зависит линейно от числа неизвестных и квадратично — от количества побочных диагоналей матрицы системы.

Последнее подтверждается следующими примерами. Время решения системы с 7-диагональной матрицей с размерами 10×10 составило 0.01–0.02 с по алгоритму 1 и 0.01 по алгоритму 2. При увеличении размеров матрицы до 100×100 и количества диагоналей — до 61 время решения составило 12 и 7.5 с, соответственно, по алгоритмам 1 и 2. Расчеты проводились на ЭВМ ЕС-1040 с помощью программы на ФОРТРАНе.

Количество требуемой памяти ЭВМ для промежуточных переменных и массивов в обоих алгоритмах составляет величину в $m(n+m-1)+1$ машинных слов. При этом учитывалось, что массивы β_i и x_i можно размещать на одном месте.

Литература

1. *Самарский А. А., Николаев Е. С.* Методы решения сеточных уравнений. М.: Наука, 1978.
2. *Николаев Е. С.* Об устойчивости метода прогонки при реализации разностных схем для уравнений теплопроводности и колебаний.– Ж. вычисл. матем. и матем. физ., 1970, т. 10, № 2, с. 478–481.

Поступила в редакцию 4.VI.1982