лении $Q_i(t)$, могут быть выбраны так, что

$$|\delta_i| \leqslant 2\varepsilon_1(|d_i| + H) + \frac{\varepsilon_2}{2}(2R + 2 + 4H^2 + H),$$

$$|\gamma_j| \leqslant \varepsilon_1 \frac{(R+2)}{2}|c_j|, \quad j \in F(i).$$

Таким образом, вычисленную совокупность величин (7) можно считать точной для матрицы $A+B$, где B — симметричная обобщенная якобиева матрица структуры (1) с элементами δ_i и γ_i вместо d_i и c_i. Спектр матрицы возмущений B оценивается так же, как и для матрицы A, т. е. по теореме Гершгорина он лежит в промежутке $[-h, h]$, где

$$h = \max_i \left\{ |\delta_i| + |\gamma_i| + \sum_{j \in F(i)} |\gamma_j| \right\}.$$

Используя полученные для δ_i и γ_j оценки, найдем

$$|\delta_i| + |\gamma_i| + \sum_{j \in F(i)} |\gamma_j| \leqslant \varepsilon_1 \left\{ \frac{(R+2)}{2} \left(|c_i| + \sum_{j \in F(i)} |c_j| \right) + 2|d_i| \right\} + $$

$$+ 2\varepsilon_1 H + \frac{\varepsilon_2}{2}(2R + 2 + H + 4H^2).$$

Поскольку $2 \leqslant \frac{R+3}{2}$, то

$$h \leqslant \varepsilon_1 \cdot \frac{R+7}{2} H + \frac{\varepsilon_2}{2}(2R + 2 + H + 4H^2) = \Delta.$$

Пусть теперь $S(t) = r$. Это значит, что у матрицы $A+B+\Delta I$ левее $t+\Delta$ с учетом их кратности лежит ровно r собственных значений. Так как матрица $B+\Delta I$ положительно полуопределенная, то у матрицы A левее $t+\Delta$ лежит не менее r собственных значений. Аналогично у матрицы $A+B-\Delta I$ левее $t-\Delta$ лежит r собственных значений. Поскольку матрица $B-\Delta I$ отрицательно полуопределенная, то у матрицы A левее $t-\Delta$ лежит не более r собственных значений. Наконец, если вычисления проведены для $t=\alpha$ и $t=\omega$, $\omega > \alpha$, и установлено, что $S(\omega) - S(\alpha) = r$, то можно утверждать, что в промежутке $[\alpha-\Delta, \omega+\Delta]$ находится не менее r собственных чисел матрицы A, а в промежутке $(\alpha+\Delta, \omega-\Delta)$ — не более r собственных чисел.

В заключение отметим, что при получении величин α_0 и ω_0 по формулам (16) и (17) тоже следует оценить неточность их вычисления и надлежащим образом раздвинуть эти границы.

ЛИТЕРАТУРА

1. **Булавский В. А., Яковлева М. А.** Обобщенные трехдиагональные матрицы и теорема Штурма.— Докл. АН СССР, 1984, т. 275, № 2, с. 277—280.
2. **Воеводин В. В.** Вычислительные основы линейной алгебры.— М.: Наука, 1977.— 304 с.
3. **Годунов С. К.** Решение систем линейных уравнений.— Новосибирск: Наука, Сиб. отд-ние, 1980.— 177 с.

ВАРИАНТ АЛГОРИТМА ОРТОГОНАЛЬНОГО ПРИВЕДЕНИЯ МАТРИЦЫ К ДВУХДИАГОНАЛЬНОМУ ВИДУ

А. Г. АНТОНОВ

В работе представлен вариант приведения прямоугольной матрицы A размерности $M \times N$ к двухдиагональному виду с помощью ортогональных преобразований отражения. Такое приведение предложено Хаусхолдером в работе [3] и подробно описано, в частности, в [1] и [2].

Предлагаемый вариант является модификацией алгоритма, взятого из книги [1], и специально приспособлен для работы на ЭВМ с двухступенчатой памятью. Приведение матрицы к двухдиагональному виду достигается за $L/2$ просмотров ее столбцов, где $L=\min(M, N)$. Проанализировано влияние ошибок округления.

§ 1. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

В книге [1] описан алгоритм приведения квадратной матрицы A размерности $N\times N$ к двухдиагональному виду

$$C=\begin{bmatrix} d_1 & b_2 & & & \\ & d_2 & b_3 & & \\ & & \ddots & \ddots & \\ & & & d_{N-1} & b_N \\ & & & & d_N \end{bmatrix}$$

серией ортогональных преобразований отражения

$$C=\mathscr{P}^{(N-1)}\mathscr{P}^{(N-2)}\ldots\mathscr{P}^{(1)}A\mathscr{Q}^{(1)}\ldots\mathscr{Q}^{(N-2)},$$

где $\mathscr{P}^{(i)}$, $\mathscr{Q}^{(i)}$ — операторы отражения. Напомним, что оператор отражения $\mathscr{P}$ полностью определяется заданием вектора p нормами к гиперплоскости отражения. Отраженный вектор $\tilde{x}=\mathscr{P}x$ находится по формуле

$$\tilde{x}=x-\frac{2}{(p, p)}(p, x)\,p$$

или, если вектор p нормирован условием $(p, p)=2$, то по

$$\tilde{x}=x-(p, x)p.$$

Отражение $\mathscr{P}A$ матрицы A слева сводится к отражению ее столбцов относительно гиперплоскости, задаваемой вектором p, а отражение $A\mathscr{Q}$ матрицы A справа — к отражению ее строк относительно гиперплоскости, задаваемой q.

Алгоритм основан на последовательном выполнении следующих шагов. Пусть после $(k-1)$-го шага приведенная матрица $A^{(k-1)}=\mathscr{P}^{(k-1)}\mathscr{P}^{(k-2)}\ldots\mathscr{P}^{(1)}A\mathscr{Q}^{(1)}\ldots\mathscr{Q}^{(k-2)}\mathscr{Q}^{(k-1)}$ имеет вид

$$A^{(k-1)}=\underbrace{\begin{bmatrix} d_1 & b_2 & & & & & & \\ & d_2 & b_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & d_{k-1} & b_k & & & \\ & & & & \times & \times & \ldots & \times \\ & & & & \times & \times & \ldots & \times \\ & & & & \vdots & \vdots & & \vdots \\ & & & & \times & \times & \ldots & \times \end{bmatrix}}_{k-1\ \text{столбцов}} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} k-1\ \text{строк} \\ \\ \\ \\ \\ \end{matrix}$$

(здесь и далее крестиками обозначены элементы матрицы, которые могут отличаться от нуля). Тогда на k-м шаге подбирается такой вектор $p^{(k)}$, чтобы отраженный k-й столбец матрицы $\tilde{a}_{\cdot k}^{(k-1)}=\mathscr{P}^{(k)}a_{\cdot k}^{(k-1)}$ принял вид

$$[\tilde{a}_{\cdot k}^{(k-1)}]^T=[0, 0, \ldots, 0, b_k, d_k, 0, \ldots, 0],$$

и выполняется отражение $\mathscr{P}^{(k)}$. Отраженная матрица $\tilde{A}^{(k-1)}=\mathscr{P}^{(k)}A^{(k-1)}$

будет выглядеть следующим образом:

$$\tilde{A}^{(k-1)} = \underbrace{\begin{bmatrix} d_1 & b_2 & & & & & & & \\ & d_2 & b_3 & & & & & 0 & \\ & & \ddots & \ddots & & & & & \\ & & & d_{k-1} & b_k & & & & \\ & & & & d_k & \times & \times & \ldots & \times \\ & 0 & & & 0 & \times & \times & \ldots & \times \\ & & & & \vdots & \vdots & \vdots & & \vdots \\ & & & & 0 & \times & \times & \ldots & \times \end{bmatrix}}_{k-1 \text{ столбцов}} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} k \text{ строк} \\ \vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}} \end{matrix}$$

После этого подбирается такой вектор $q^{(k)}$, чтобы отраженная k-я строка матрицы $a_{k\cdot}^{(k)} = \tilde{a}_{k\cdot}^{(k-1)} Q^{(k)}$ приняла вид

$$a_{k\cdot}^{(k)} = [0, 0, \ldots, 0, d_k, b_{k+1}, 0, \ldots, 0],$$

и выполняется отражение $Q^{(k)}$.

После k-го шага преобразованная матрица $A^{(k)} = \tilde{A}^{(k-1)} Q^{(k)} = \mathcal{P}^{(k)} A^{(k-1)} Q^{(k)}$ выглядит так:

$$A^{(k)} = \underbrace{\begin{bmatrix} d_1 & b_2 & & & & & & \\ & d_2 & b_3 & & & & 0 & \\ & & \ddots & \ddots & & & & \\ & & & d_k & b_{k+1} & & & \\ & & & & \times & \times & \ldots & \times \\ & 0 & & & \times & \times & \ldots & \times \\ & & & & \vdots & \vdots & & \vdots \\ & & & & \times & \times & \ldots & \times \end{bmatrix}}_{k \text{ столбцов}} \begin{matrix} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} k \text{ строк} \\ \vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}} \end{matrix}$$

После того, как на $(N-1)$-м шаге алгоритма подбирается соответствующим образом вектор отражения $p^{(N-1)}$, отраженная матрица $\tilde{A}^{(N-2)} = \mathcal{P}^{(N-1)} A^{(N-2)}$ принимает двухдиагональный вид C.

§ 2. АЛГОРИТМ ПРИВЕДЕНИЯ К ДВУХДИАГОНАЛЬНОМУ ВИДУ ПРЯМОУГОЛЬНОЙ МАТРИЦЫ $A_{M\times N}$ ДЛЯ СЛУЧАЯ $M \geqslant N$

Рассмотрим алгоритм приведения к двухдиагональному виду прямоугольной матрицы A размерности $M \times N$, где число строк M больше или равно числу столбцов N. На нулевом шаге алгоритма находится отраженная по столбцам матрица $\tilde{A}^{(0)} = \mathcal{P}^{(1)} A$, затем на основании рекуррентных соотношений вида $\tilde{A}^{(k)} = \mathcal{P}^{(k+1)} \tilde{A}^{(k-1)} Q^{(k)}$ на k-м шаге определяются преобразованные матрицы $\tilde{A}^{(k)}$, полученные отражениями матриц $\tilde{A}^{(k-1)}$ сначала по строкам, а потом по столбцам. Тогда, если $M = N$, то после $(N-2)$-го шага получается двухдиагональная матрица $C = \tilde{A}^{(N-2)} = \mathcal{P}^{(N-1)}(\mathcal{P}^{(N-2)} \ldots (\mathcal{P}^{(2)}(\mathcal{P}^{(1)} A) Q^{(1)}) \ldots Q^{(N-3)}) Q^{(N-2)}$, а если $M > N$, то находится матрица $\tilde{A}^{(N-1)} = \mathcal{P}^{(N)} \tilde{A}^{(N-2)}$ следующего вида:

$$\tilde{A}^{(N-1)} = \left[\begin{array}{c} C \\ \hline 0 \end{array}\right] = \left[\begin{array}{ccccc} d_1 & b_2 & & & 0 \\ & d_2 & b_3 & & \\ & & \ddots & \ddots & \\ & & & d_{N-1} & b_N \\ 0 & & & & d_N \\ \hline & & 0 & & \end{array}\right].$$

Опишем теперь работу алгоритма подробнее. На нулевом шаге первоначально определяется вектор отражения с компонентами

$$\sigma^{(1)} = \begin{cases} +1, & \text{если } a_{11} \geqslant 0, \\ -1, & \text{если } a_{11} < 0; \end{cases}$$

$$p_1 = a_{11} + \sigma^{(1)} \sqrt{a_{11}^2 + a_{21}^2 + \ldots + a_{M1}^2};$$

$$p_i = a_{i1} \quad (i = 2, 3, \ldots, M).$$

Затем находятся компоненты вектора $x^{(1)}$ скалярных произведений вектора $p^{(1)}$ на столбцы матрицы A, и вычисляются столбцы отраженной матрицы $\tilde{A}^{(0)} = \mathscr{P}^{(1)}A$ по формулам

$$x_j^{(1)} = (p^{(1)}, a_{\cdot j});$$

$$\tilde{a}_{\cdot j}^{(0)} = a_{\cdot j} - \frac{2}{(p^{(1)}, p^{(1)})} x_j^{(k+1)} p^{(k+1)} \quad (j = 1, 2, \ldots, N).$$

Кроме того, на нулевом шаге определяется вектор скалярных произведений $w^{(1)}$ с компонентами

$$w_i^{(1)} = (z^{(1)}, \tilde{a}_{i\cdot}^{(0)}) \quad (i = 1, 2, \ldots, M),$$

где $z^{(1)}$ — вектор, имеющий компоненты $z_1^{(1)} = z_2^{(1)} = 0$, $z_j^{(1)} = \tilde{a}_{1,j}^{(0)}$ $(j = 3, 4, \ldots, N)$. На этом работа нулевого шага заканчивается.

Пусть теперь после $(k-1)$-го шага имеются матрица $\tilde{A}^{(k-1)}$ и вектор скалярных произведений $w^{(k)}$. Тогда на k-м шаге сначала находятся значения компонент вектора $q^{(k)}$ по формулам

$$q_1^{(k)} = q_2^{(k)} = \ldots = q_k^{(k)} = 0;$$

$$\tau^{(k)} = \begin{cases} +1, & \text{если } \tilde{a}_{k,k+2}^{(k-1)} \geqslant 0, \\ -1, & \text{если } \tilde{a}_{k,k+2}^{(k-1)} < 0; \end{cases}$$

$$q_{k+1}^{(k)} = \tilde{a}_{k,k+1}^{(k-1)} + \tau^{(k)} \sqrt{[\tilde{a}_{k,k+1}^{(k-1)}]^2 + [\tilde{a}_{k,k+2}^{(k-1)}]^2 + \ldots + [\tilde{a}_{k,N}^{(k-1)}]^2};$$

$$q_j^{(k)} = \tilde{a}_{k,j}^{(k-1)} \quad (j = k+2, k+3, \ldots, N).$$

Затем вычисляются компоненты вектора $y^{(k)}$

$$y_i^{(k)} = w_i^{(k)} + q_{k+1}^{(k)} \tilde{a}_{i,k+1}^{(k-1)} \quad (i = 1, 2, \ldots, M)$$

и строки отраженной матрицы $A^{(k)} = \tilde{A}^{(k-1)}Q^{(k)}$

$$a_{i\cdot}^{(k)} = \tilde{a}_{i\cdot}^{(k-1)} - \frac{2}{(q^{(k)}, q^{(k)})} y_i^{(k)} q^{(k)} \quad (i = 1, 2, \ldots, M).$$

Далее находится вектор отражения $p^{(k+1)}$ с компонентами

$$p_1^{(k+1)} = p_2^{(k+1)} = \ldots = p_k^{(k+1)} = 0;$$

$$\sigma^{(k+1)} = \begin{cases} +1, & \text{если } a_{k+1,k+1}^{(k)} \geqslant 0, \\ -1, & \text{если } a_{k+1,k+1}^{(k)} < 0; \end{cases}$$

$$p_{k+1}^{(k+1)} = a_{k+1,k+1}^{(k)} + \sigma^{(k+1)} \sqrt{[a_{k+1,k+1}^{(k)}]^2 + [a_{k+2,k+1}^{(k)}]^2 + \ldots + [a_{M,k+1}^{(k)}]^2};$$

$$p_i^{(k+1)} = a_{i,k+1}^{(k)} \quad (i = k+2, k+3, \ldots, M),$$

после чего вычисляются компоненты вектора $x^{(k+1)}$ скалярных произведений вектора $p^{(k+1)}$ на столбцы матрицы $A^{(k)}$ и определяются столбцы преобразованной матрицы $\tilde{A}^{(k)} = \mathscr{P}^{(k+1)}A^{(k)}$ по формулам

$$x_j^{(k+1)} = (p^{(k+1)}, a_{\cdot j}^{(k)}),$$

$$\tilde{a}_{\cdot j}^{(k)} = a_{\cdot j}^{(k)} - \frac{2}{(p^{(k+1)}, p^{(k+1)})} x_j^{(k+1)} p^{(k+1)} \quad (j = 1, 2, \ldots, N).$$

Помимо этого, на k-м шаге вычисляется вектор скалярных произведений $w^{(k+1)}$ с компонентами

$$w_i^{(k+1)} = \left(z^{(k+1)}, \tilde{a}_{i\cdot}^{(k)}\right) \quad (i = 1, 2, \ldots, M),$$

где $z^{(k+1)}$ — вектор, имеющий компоненты

$$z_1^{(k+1)} = z_2^{(k+1)} = \ldots z_{k+2}^{k+1} = 0, \quad z_j^{(k+1)} = \tilde{a}_{k+1,j}^{(k)} \quad (j = k+3, k+4, \ldots, N).$$

Легко видеть, что для компонент вектора $w^{(k+1)}$ выполняются равенства

$$w_i^{(k+1)} + q_{k+2}^{(k+1)}\tilde{a}_{i,k+1}^{(k)} = \left(q^{(k+1)}, \tilde{a}_{i\cdot}^{(k)}\right) = y_i^{(k+1)} \quad (i = 1, 2, \ldots, M).$$

Если $M = N$, то после $(N-2)$-го шага матрица приведена к двухдиагональному виду C. Если же $M > N$, то на $(N-1)$-м шаге находится вектор отражения $p^{(N)}$ с компонентами

$$p_1^{(N)} = p_2^{(N)} = \ldots = p_{N-1}^{(N)} = 0;$$

$$\sigma^{(N)} = \begin{cases} +1, & \text{если } \tilde{a}_{N,N}^{(N-2)} \geqslant 0, \\ -1, & \text{если } \tilde{a}_{N,N}^{(N-2)} < 0; \end{cases}$$

$$p_N^{(N)} = \tilde{a}_{N,N}^{(N-2)} + \sigma^{(N)} \sqrt{\left[\tilde{a}_{N,N}^{(N-2)}\right]^2 + \left[\tilde{a}_{N+1,N}^{(N-2)}\right]^2 + \ldots + \left[\tilde{a}_{M,N}^{(N-2)}\right]^2},$$

$$p_i^{(N)} = \tilde{a}_{i,N}^{(N-2)} \quad (i = N+1, N+2, \ldots, M).$$

После этого вычисляются компоненты вектора $x^{(N)}$ и находятся столбцы отраженной матрицы $\tilde{A}^{(N-1)} = \mathscr{P}^{(N)}\tilde{A}^{(N-2)}$ по формулам

$$x_j^{(N)} = \left(p^{(N)}, \tilde{a}_{\cdot j}^{(N-2)}\right);$$

$$\tilde{a}_{\cdot j}^{(N-1)} = \tilde{a}_{\cdot j}^{(N-2)} - \frac{2}{\left(p^{(N)}, p^{(N)}\right)} x_j^{(N)} p^{(N)} \quad (j = 1, 2, \ldots, N).$$

Полученная матрица $\tilde{A}^{(N-1)}$ будет иметь нужный нам двухдиагональный вид.

Отметим, что приведенные формулы расчета элементов отраженных матриц и векторов отражения аналогичны полученным в [1] для случая квадратных матриц, за исключением формул для $y^{(k)}$.

§ 3. РЕАЛИЗАЦИЯ АЛГОРИТМА НА ЭВМ С ДВУХСТУПЕНЧАТОЙ ПАМЯТЬЮ. АНАЛИЗ ОШИБОК ОКРУГЛЕНИЯ

Рассмотрим реализацию алгоритма приведения к двухдиагональному виду прямоугольной матрицы A размерности $M \times N$ $(M \geqslant N)$ на машине с двухступенчатой памятью. Сначала приведем алгоритм, строящий компоненты $p_m, p_{m+1}, \ldots, p_N$ вектора отражения p, нормированного условием $(p, p) = 2$, и элемента $d = \tilde{s}_m$ отраженного вектора $\tilde{s} = \mathscr{P}s$ по заданным компонентам $s_m, s_{m+1}, \ldots, s_N$ отражаемого вектора s и числу m. Этот алгоритм взят из книги [1]. Он состоит из следующих шагов.

1. Среди элементов $s_m, s_{m+1}, \ldots, s_N$ выбирается максимальный по модулю. Пусть это будет число ω. Если $\omega = 0$, то полагается $d = 0$, $p_m = \sqrt{2}$, $p_{m+1} = p_{m+2} = \ldots = p_N = 0$ и на этом работа алгоритма заканчивается. Если $\omega \neq 0$, то выполняются последующие шаги.

2. По внутреннему машинному представлению числа ω: $\omega = \pm\gamma^r m_\omega$ $(1/\gamma \leqslant m_\omega < 1)$, где γ — основание машинного счисления, зависящее от типа ЭВМ, вместо исходной последовательности $s_m, s_{m+1}, \ldots, s_N$ строится последовательность $t_m, t_{m+1}, \ldots, t_N$ по $t_i = s_i\gamma^{-r}$ $(i = m, m+1, \ldots, N)$.

3. Вычисляется с повышенной точностью сумма $\varkappa = t_m^2 + t_{m+1}^2 + \ldots + t_N^2$.

4. Определяются значения t и d по формулам

$$t=\begin{cases}-\sqrt{\varkappa}, & \text{если } t_m\geqslant 0\\ \sqrt{\varkappa}, & \text{если } t_m<0\end{cases},\quad d=\gamma^T t.$$

5. Находятся компоненты ненормированного вектора отражения r_m, $r_{m+1}, \ldots, r_N$: $r_m=t_m-t$, $r_i=t_i$ $(i=m+1,\ m+2,\ \ldots,\ N)$, а также нормирующий множитель $\pi:\pi=\sqrt{1/2\left(r_m^2+r_{m+1}^2+\ldots+r_N^2\right)}$.

6. Вычисляются компоненты нормированного вектора отражения p_m, $p_{m+1}, \ldots, p_N$: $p_i=r_i/\pi$ $(i=m,\ m+1,\ \ldots,\ N)$ и на этом работа алгоритма заканчивается.

Пусть матрица A размерности $M\times N$ $(M\geqslant N)$ хранится во внешней памяти машины по столбцам. Тогда предлагаемый алгоритм потребует следующего распределения памяти. Во внешней памяти помимо места для исходной матрицы A должно быть отведено место для хранения векторов отражения $p^{(1)}, p^{(2)}, \ldots, p^{(N)}$ и $q^{(1)}, q^{(2)}, \ldots, q^{(N-2)}$. В оперативной памяти машины отводится место для хранения главной диагонали $d=(d_1, d_2, \ldots, d_N)$ и наддиагонали $b=(b_2, b_3, \ldots, b_N)$ матрицы, приведенной к нужному двухдиагональному виду. Помимо этого, на k-м шаге работы алгоритма в оперативной памяти машины необходимо хранить следующие массивы:

$p=(p_1, p_2, \ldots, p_M)$, в котором будет размещаться вектор отражения $p^{(k+1)}$;

$q=(q_1, q_2, \ldots, q_N)$ с вектором отражения $q^{(k)}$;

$a=(a_1, a_2, \ldots, a_M)$, в котором будет размещаться обрабатываемый столбец матрицы $\tilde{A}^{(k)}$;

$c=(c_1, c_2, \ldots, c_N)$, с k-й строкой матрицы $\tilde{A}^{(k)}$;

$y=(y_1, y_2, \ldots, y_M)$ и $w=(w_1, w_2, \ldots, w_M)$, в которых будут размещаться векторы $y^{(k)}$ и $w^{(k+1)}$.

По информации об объеме доступной оперативной памяти можно получить ограничения сверху на размерности M и N приводимой матрицы.

Перед началом нулевого шага алгоритма будем иметь исходную матрицу A, хранящуюся во внешней памяти. Нулевой шаг состоит из следующих пунктов.

1. Вводится первый столбец матрицы A.

2. По элементам введенного столбца строится вектор отражения $p^{(1)}$ и находится элемент главной диагонали d_1 по алгоритму из [1] при $m=1$. Вектор $p^{(1)}$ заносится во внешнюю память.

3. Вектор w принимается равным нулю.

4. Обрабатывается j-й столбец матрицы $(j=2, 3, \ldots, N)$.

4.1. Вводится j-й столбец.

4.2. Находится скалярное произведение вектора отражения $p^{(1)}$ на столбец $x\leftarrow\sum\limits_{l=1}^{m} p_l a_l$.

4.3. Находятся элементы отраженного столбца $a_i\leftarrow a_i-p_i x$ $(i=1, 2, \ldots, M)$.

4.4. Запоминается j-я компонента первой строки преобразованной матрицы $c_j\leftarrow a_1$.

4.5. Подсчитываются частичные суммы для $j\geqslant 3$: $w_i\leftarrow w_i+a_1 a_i$ $(i=2, 3, \ldots, M)$.

4.6. Отраженный столбец записывается во внешнюю память. На этом нулевой шаг заканчивается. По его окончании во внешней памяти машины будут храниться матрица $\tilde{A}^{(0)}=\mathscr{P}^{(1)}A$ и вектор отражения $p^{(1)}$, а в оперативной памяти — первая строка матрицы $\tilde{A}^{(0)}$, вектор $w^{(1)}$ и элемент d_1 двухдиагональной матрицы.

Пусть перед началом работы k-го шага во внешней памяти хранится матрица $\tilde{A}^{(k-1)}$, векторы отражения $p^{(1)}, p^{(2)}, \ldots, p^{(k)}$ и $q^{(1)}, q^{(2)}, \ldots$ $\ldots, q^{(k-1)}$, а в оперативной — k-я строка матрицы $\tilde{A}^{(k-1)}$, вектор $w^{(k)}$, а также элементы $d_1, d_2, \ldots, d_k$ и $b_2, b_3, \ldots, b_k$ двухдиагональной матрицы. Из особенностей построения векторов $q^{(k)}$ и $p^{(k+1)}$ явствует, что на k-м шаге алгоритма не изменяются первые k элементов обрабатываемых строк и столбцов, следовательно, можно начинать обработку матрицы $\tilde{A}^{(k-1)}$ с k-го столбца. Тогда последовательность действий k-го шага будет выглядеть следующим образом.

1. По элементам k-й строки матрицы $\tilde{A}^{(k-1)}$ строятся вектор отражения $q^{(k)}$ и элемент наддиагонали b_{k+1} согласно алгоритму из [1] при $m=k+1$. Вектор $q^{(k)}$ заносится во внешнюю память.

2. Нормируются компоненты вектора $w^{(k)}$, хранящегося в w: $w_i \leftarrow \leftarrow w_i\gamma^{-r}/\pi$ $(i=k+1,\ k+2,\ \ldots,\ M)$, где значения r и π определяются в алгоритме из [1].

3. Вводится $(k+1)$-й столбец матрицы $\tilde{A}^{(k-1)}$.

4. Находятся компоненты вектора $y^{(k)}$ и заносятся в y: $y_i \leftarrow w_i + q_{k+1}a_i$ $(i=k+1,\ k+2,\ \ldots,\ M)$.

5. Элементы $(k+1)$-го столбца матрицы $\tilde{A}^{(k-1)}$ преобразуются по формулам $a_i \leftarrow a_i - q_{k+1}y_i$ $(i=k+1,\ k+2,\ \ldots,\ M)$.

6. По элементам преобразованного $(k+1)$-го столбца находится вектор отражения $p^{(k+1)}$ и элемент главной диагонали d_{k+1} по алгоритму из [1] при $m=k+1$. Вектор $p^{(k+1)}$ заносится во внешнюю память.

7. Вектор w принимается равным нулю.

8. Обрабатывается j-й столбец матрицы $(j=k+2,\ k+3,\ \ldots,\ N)$.

8.1. Вводится j-й столбец.

8.2. Элементы столбца преобразуются по формулам $a_i \leftarrow a_i - q_jy_i$ $(i=k+1,\ k+2,\ \ldots,\ M)$.

8.3. Находится скалярное произведение вектора отражения $p^{(k+1)}$ на столбец $a_l : x \leftarrow \sum\limits_{l=k+1}^{M} p_la_l$.

8.4. Элементы столбца преобразуются по формулам $a_i \leftarrow a_i - p_ix$ $(i=k+1,\ k+2,\ \ldots,M)$.

8.5. Запоминается j-я компонента $(k+1)$-й строки преобразованной матрицы: $c_j \leftarrow a_{k+1}$.

8.6. Подсчитываются частичные суммы для $j \geqslant k+3$: $w_i \leftarrow w_i + a_{k+1}a_i$ $(i=k+2,\ k+3,\ \ldots,\ M)$.

8.7. Преобразованный столбец заносится во внешнюю память.

На этом k-й шаг заканчивается. По его окончании во внешней памяти будут размещаться матрица $\tilde{A}^{(k)} = \mathscr{P}^{(k+1)}\tilde{A}^{(k-1)}\mathcal{Q}^{(k)}$, векторы отражения $p^{(1)},\ p^{(2)},\ \ldots,\ p^{(k+1)}$ и $q^{(1)},\ q^{(2)},\ \ldots,\ q^{(k)}$, а в оперативной памяти — $(k+1)$-я строка матрицы $\tilde{A}^{(k)}$, вектор $w^{(k+1)}$ и элементы $d_1,\ d_2,\ \ldots,\ d_{k+1}$ и $b_2,\ b_3,\ \ldots,\ b_{k+1}$ двухдиагональной матрицы.

Перед началом $(N-1)$-го шага алгоритма во внешней памяти хранятся матрица $\tilde{A}^{(N-2)}$ и векторы отражения $p^{(1)},\ p^{(2)},\ \ldots,\ p^{(N-1)}$, $q^{(1)},\ q^{(2)},\ \ldots,\ q^{(N-2)}$. В оперативной памяти находятся элементы $d_1,\ d_2,\ \ldots$ $\ldots,\ d_{N-1}$ и $b_2,\ b_3,\ \ldots,\ b_{N-1}$ двухдиагональной матрицы. Значения $\tilde{a}_{N-1}^{(N-2)}$ и $w^{(N-1)}$ на этом шаге не требуются. Последовательность действий $(N-1)$-го шага:

1. Вводится N-й столбец матрицы $\tilde{A}^{(N-2)}$.

2. Значение b_N кладется равным $(N-1)$-му элементу столбца.

3. Сравниваются значения M и N. Если $M=N$, то значение d_N берется равным N-му элементу столбца. Если же $M>N$, то по элементам N-го столбца находятся компоненты вектора $p^{(N)}$ и элемент d_N по алгоритму из [1] при $m=N$, после чего вектор $p^{(N)}$ заносится во внешнюю память.

На этом работа $(N-1)$-го шага заканчивается. По его окончании во внешней памяти хранятся векторы отражения $p^{(1)},\ p^{(2)},\ \ldots,\ p^{(N)}$ и $q^{(1)},\ q^{(2)},\ \ldots,\ q^{(N-2)}$, а в оперативной памяти — диагональ $d=(d_1,\ d_2,\ \ldots$ $\ldots,\ d_N)$ и наддиагональ $b=(b_2,\ b_3,\ \ldots,\ b_N)$ двухдиагональной матрицы нужного вида. Векторы $p^{(1)},\ p^{(2)},\ \ldots,\ p^{(N)}$ можно хранить на месте соответствующих столбцов матрицы A, а для хранения векторов $q^{(1)}$, $q^{(2)},\ \ldots,\ q^{(N-2)}$ следует организовать специальный массив.

Как можно видеть из приведенного алгоритма, на k-м шаге ($k = 0, 1, \ldots, N-1$) требуется просматривать $N-k$ последних столбцов преобразуемой матрицы. Тогда для приведения матрицы к двухдиагональному виду потребуется $\sum\limits_{k=0}^{N-1}(N-k) \approx N/2$ просмотров ее по столбцам.

При реализации предложенного алгоритма на ЭВМ могут возникнуть дополнительные ошибки по сравнению с аналогичным алгоритмом, описанным в книге [1]. Они обусловлены тем, что компоненты вектора $y^{(k)}$ считаются на k-м шаге иным способом, чем в [1]. В данном случае, из-за того, что вектор $q^{(k)}$ сначала не нормируется условием $(q, q) = 2$, может произойти замена одного или нескольких произведений вида $q_j^{(k)}\tilde{a}_{ij}^{(k-1)}$ машинным нулем.

Если на k-м шаге работы алгоритма будут известны априорные оценки для норм перемножаемых векторов $\sqrt{M\varepsilon_2/\varepsilon_1} < \|q^{(k)}\|$, $\sqrt{M\varepsilon_2/\varepsilon_1} < \|\tilde{a}_{i\cdot}^{(k-1)}\|$ ($i = k+1, k+2, \ldots, M$) и ограничение на N: $N < 1/\varepsilon_1$, где ε_1 и ε_2 — машинные константы, определенные в [1], то никаких дополнительных погрешностей не возникает. Если же таких оценок нет, то гарантировать работу алгоритма с точностью не хуже, чем в [1], можно нормировкой вектора $w^{(k+1)}$. Для этого на k-м шаге выполняются следующие дополнительные действия.

При просмотре $(k+2)$-го столбца матрицы $\tilde{A}^{(k-1)}$ берется элемент $\tilde{a}_{k+1,k+2}^{(k-1)} = \pm\gamma^{r_{k+2}}m_{k+2}$ $(1/\gamma \leqslant m_{k+2} < 1)$ и находится его порядок r_{k+2}, размещаемый в r. При просмотре j-го столбца матрицы $\tilde{A}^{(k-1)}$ ($j = k+3, k+4, \ldots, N$) берется элемент $\tilde{a}_{k+1,j}^{(k-1)}$ и находится его порядок r_j. Если $r_j \leqslant r$, то из $\tilde{a}_{k+1,j}^{(k-1)}$ выносится порядок r: $\hat{a}_{k+1,j}^{(k-1)} \leftarrow \tilde{a}_{k+1,j}^{(k-1)}\gamma^{-r}$ и находится значение соответствующей частичной суммы $w_i^{(k+1)} \leftarrow w_i^{(k+1)} + \hat{a}_{k+1,j}^{(k-1)}a_{i,j}^{(k-1)}$ ($i = k+2, k+3, \ldots, M$). Если же $r_j > r$, то из $\tilde{a}_{k+1,j}^{(k-1)}$ выносится порядок r_j: $\hat{a}_{k+1,j}^{(k-1)} \leftarrow \tilde{a}_{k+1,j}^{(k-1)}\gamma^{-r_j}$, а из сумм $w_i^{(k+1)}$ — порядок $r_j - r$: $\hat{w}_i^{(k+1)} \leftarrow w_i^{(k+1)}\gamma^{-(r_j-r)}$ и вычисляется значение соответствующих частичных сумм $w_i^{(k+1)} \leftarrow \hat{w}_i^{(k+1)} + \hat{a}_{k+1,j}^{(k-1)}\hat{a}_{i,j}^{(k-1)}$ ($i = k+2, k+3, \ldots, M$). Порядок r принимается в этом случае равным r_j. При такой организации подсчета $w^{(k+1)}$ дополнительные погрешности не возникают. Кроме того, попутно можно найти $\omega = \max\left(|\tilde{a}_{k+1,k+1}^{(k-1)}|, |\tilde{a}_{k+1,k+2}^{(k-1)}|, \ldots, |\tilde{a}_{k+1,M}^{(k-1)}|\right)$, что позволит упростить шаг 1 алгоритма вычисления вектора отражения из [1].

§ 4. АЛГОРИТМ ПРИВЕДЕНИЯ К ДВУХДИАГОНАЛЬНОМУ ВИДУ ПРЯМОУГОЛЬНОЙ МАТРИЦЫ $A_{M\cdot N}$ ДЛЯ СЛУЧАЯ $M<N$

Рассмотрим алгоритм приведения матрицы A размерности $M \times N$, где число строк M меньше числа столбцов N. Пусть перед k-м шагом имеем матрицу $A^{(k-1)} = \mathcal{P}^{(k-1)}\mathcal{P}^{(k-2)}\ldots\mathcal{P}^{(1)}AQ^{(1)}\ldots Q^{(k-2)}Q^{(k-1)}$ вида

$$\tilde{A}^{(k-1)} = \begin{array}{l} \left[\begin{array}{ccccc|c} d_M & & & & & \\ b_M & d_{M-1} & & & 0 & \\ & b_{M-1} & \ddots & & & \\ & & \ddots & & & \\ & & & d_{M-k+2} & & \\ & & & b_{M-k+2} & \times\ \times \ldots \times & \\ & & & & \times\ \times \ldots \times & \\ & 0 & & & \vdots & \\ & & & & \times\ \times \ldots \times & \end{array}\right] \\ \quad\underbrace{\hphantom{d_M\ b_{M-1}\ \ddots\ d_{M-k+2}}}_{k-1 \text{ столбцов}} \end{array} \left.\vphantom{\begin{array}{c} 1\\1\\1\\1\\1 \end{array}}\right\} k-1 \text{ строк}.$$

Тогда на k-м шаге строится вектор отражения $q^{(k)}$ такой, чтобы отраженная k-я строка $\tilde{a}_{k\cdot}^{(k-1)} = a_{k\cdot}^{(k-1)}Q^{(k)}$ приняла вид $\tilde{a}_{k\cdot}^{(k-1)} = [0, 0, \ldots, 0, b_{M-k+2}, d_{M-k+1}, 0, \ldots, 0]$, и находится матрица $\tilde{A}^{(k-1)} = A^{(k-1)}Q^{(k)}$. После чего вычисляются компоненты вектора отражения $p^{(k)}$ такого, чтобы отраженный k-й столбец $a_{\cdot k}^{(k)} = \mathcal{P}^{(k)}\tilde{a}_{\cdot k}^{(k-1)}$ принял вид $\left[a_{\cdot k}^{(k)}\right]^T = [0, 0, \ldots, 0, d_{M-k+1}, b_{M-k+1}, 0, \ldots, 0]$, и строится матрица $A^{(k)} = \mathcal{P}^{(k)}\tilde{A}^{(k-1)}$. По окончании k-го шага преобразованная матрица $A^{(k)} = \mathcal{P}^{(k)}A^{(k-1)}Q^{(k)}$ будет иметь следующий вид:

$$A^{(k)} = \left.\underbrace{\begin{bmatrix} d_M & & & & & & \\ b_M & d_{M-1} & \ddots & & 0 & & \\ & b_{M-1} & \ddots & & \vdots & & \\ & & \ddots & d_{M-k-1} & \times & \times \ldots & \times \\ & & \cdot & b_{M-k+1} & \times & \times \ldots & \times \\ & 0 & & & \cdot & \cdot \ldots & \cdot \\ & & & & \times & \times \ldots & \times \end{bmatrix}}_{k \text{ столбцов}}\right\} k \text{ строк}.$$

Кроме того, на k-м шаге ищется вектор $w^{(k+1)}$ скалярных произведений некоторого вектора $z^{(k+1)}$ на строки матрицы $A^{(k)}$.

Формулы, по которым вычисляются компоненты векторов $q^{(k)}$, $p^{(k)}$, $w^{(k+1)}$, $z^{(k+1)}$, а также элементы матрицы $A^{(k)}$ аналогичны приведенным в § 2 для случая $M \geqslant N$. Для успешной работы алгоритма необходимо перед первым шагом алгоритма сосчитать $w^{(1)}$.

После $(M-2)$-го шага алгоритма получается матрица $A^{(M-2)} = (\mathcal{P}^{(M-2)}(\mathcal{P}^{(M-1)}\ldots(\mathcal{P}^{(1)}AQ^{(1)})\ldots Q^{(M-1)})Q^{(M-2)})$. Затем на $(M-1)$-м и M-м шагах алгоритма подбираются такие векторы отражения $q^{(M-1)}$ и $q^{(M)}$ соответственно, что преобразованная матрица $A^{(M)} = A^{(M-1)}Q^{(M)} = A^{(M-2)}Q^{(M-1)}Q^{(M)}$ имеет вид

$$A^{(M)} = \left[\begin{array}{c|c} \underbrace{\begin{matrix} d_M & & & & \\ b_M & d_{M-1} & \cdot & \cdot & 0 \\ & b_{M-1} & \cdot & \cdot & \\ & & & \cdot & d_2 \\ 0 & & & \cdot b_2 & \cdot\; d_1 \end{matrix}}_{M \text{ столбцов}} & 0 \end{array}\right].$$

На $(M+1)$-м шаге алгоритма матрица $A^{(M)}$ умножается на две матрицы перестановок: слева — на F размерности $M \times M$ вида

$$F = \begin{bmatrix} 0 & & 1 \\ & \cdot\;\cdot\;\cdot 1 & \\ 1 & & 0 \end{bmatrix},$$

а справа — на G размерности $N \times N$ вида

$$G = \left[\begin{array}{c|c} F & 0 \\ \hline 0 & \begin{matrix} 1 & & \\ & 1 \cdot \cdot & 0 \\ 0 & & 1 \end{matrix} \end{array}\right].$$

Полученная матрица $A^{(M+1)}=FA^{(M)}G$ будет иметь нужный вид

$$A^{(M+1)}=[C\,|\,0]=\left[\begin{array}{ccccc|c} d_1 & b_2 & & & & \\ & d_2 & b_3 & & 0 & \\ & & \ddots & \ddots & & 0 \\ & & & d_{M-1} & b_M & \\ 0 & & & & d_M & \end{array}\right].$$

M столбцов

Реализация этого алгоритма на машине с двухступенчатой памятью во многом аналогична описанной в § 3, и ее не будем рассматривать. Отметим только, что $(M+1)$-й шаг алгоритма сводится к перестановке в обратном порядке элементов d и b и к занесению во внешнюю память отметок о сделанной перестановке.

Аналогично тому, как это сделано в § 3, для предложенного здесь алгоритма можно получить число просмотров матрицы (при приведении ее к двухдиагональному виду), приблизительно равное $M/2$, и провести анализ возникающих ошибок округления.

* * *

Автор выражает глубокую благодарность С. К. Годунову за постановку задачи и постоянное внимание к работе.

ЛИТЕРАТУРА

1. **Годунов С. К.** Решение систем линейных уравнений.— Новосибирск: Наука, Сиб. отд-ние, 1980.— 177 с.
2. **Уилкинсон Дж. X.** Алгебраическая проблема собственных значений.— М.: Наука, 1970.— 564 с.
3. **Householder A. S.** Unitary Triangularization of a nonsymmetric Matrix.— J. Assoc. Comput. Math., 1958, v. 5, p. 339—342.

МЕТОД ОРТОГОНАЛЬНЫХ ВСТРЕЧНЫХ ПРОГОНОК ДЛЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ С ЛЕНТОЧНОЙ МАТРИЦЕЙ

С. И. ФАДЕЕВ

1. Краткое описание метода. Предлагаемая работа посвящена методу ортогональных встречных прогонок для решения системы из N линейных алгебраических уравнений с действительными коэффициентами

$$Ax=F. \tag{1}$$

Матрица A размерности $N\times N$ имеет ленточную структуру, F — вектор правых частей, x — вектор неизвестных. Будем предполагать, что A достаточно хорошо обусловлена, не вводя никаких других ограничений на коэффициенты ленточного массива.

Алгоритм встречных прогонок, основанный на исключении неизвестных по методу Гаусса, описан, например, в [1]. Если ширину ленты матрицы A составляет $M+1$ элемент, $N>M$, или, иными словами, если M — порядок разностной краевой задачи, как еще можно рассматривать (1), то в результате применения встречных ходов прогонки (прямой правый и прямой левый ходы) образуется система относительно группы