Тогда

$$\Gamma_k = b_k - \hat{z}_k^{(k-1)}, \tag{7.5.I2}$$

а векторы $\hat{z}^{(k)}$ вычисляются рекуррентно следующим способом:

$$\hat{z}^{(k)} = \hat{z}^{(k-1)} + \left[0 \dots 0 \; I \; \rho_{k+1}^{(k)} \; \dots \; \rho_{n-1}^{(k)}\right]' \, \Gamma_k . \tag{7.5.I3}$$

Построенный векторный алгоритм имеет параллельную форму высотой $O(n)$.

§ 8. Дихотомия в векторных алгоритмах

8.I. Быстрые алгоритмы для тёплицевых систем

Рассмотрим систему (5.I.I), считая, что матрица $A=[a_{i-j}]_{ij=0}^{n-1}$ блочно-тёплицева, блочного порядка n, с блоками порядка p. Будем опираться на то, что процесс решения системы может быть разделен на два этапа: I) вычисление тёплицева разложения матрицы A^{-1}; 2) умножения матрицы A^{-1} на вектор правой части с использованием ее тёплицева разложения. Этап 2) реализуется ценой $O(n\log_2 n)$ операций, так как сводится к нескольким умножениям на вектор тёплицевых матриц. Поэтому если мы хотим ускорить процесс в целом, то это следует сделать в отношении этапа I).

Предположим, что матрица A невырожденная вместе со всеми ведущими подматрицами $A_k=[a_{i-j}]_{ij=0}^{k}$. Начнем с того, что возьмем (как основу для наших построений) векторный алгоритм (7.I.I0) и получим для него полиномиальное представление, т.е. заменим операции с векторами на операции с многочленами. Блочные многочлены от переменной z нулевого шага определим таким образом:

$$x^{(0)}(z) = y^{(0)}(z) = I,$$

$$\beta_i^{(0)}(z) = \sum_{j=0}^{n-2} \beta_{ij}^{(0)} z^j = \sum_{j=0}^{n-2} d_{i,n-2-j}^{(0)} z^j, \quad i=1,2,3,4. \tag{8.I.I}$$

Далее, многочлены k-го шага введем с помощью следующих рекуррентных соотношений

$$\left[x^{(k)}(z)\, y^{(k)}(z)\right] = \left[x^{(k-1)}(z)\, y^{(k-1)}(z)\right] \begin{bmatrix} I & t_k \\ s_k z & z \end{bmatrix}, \tag{8.I.2}$$

$$\left[\beta_1^{(k)}(z)\, \beta_2^{(k)}(z)\right] = \left[\beta_1^{(k-1)}(z)\, \beta_2^{(k-1)}(z)\right] \begin{bmatrix} z & t_k z \\ s_k & I \end{bmatrix}, \tag{8.I.3}$$

$$\left[\beta_3^{(k)}(z)\, \beta_4^{(k)}(z)\right] = \left[\beta_3^{(k-1)}(z)\, \beta_4^{(k-1)}(z)\right] \begin{bmatrix} I & t_k \\ s_k z & z \end{bmatrix}. \tag{8.I.4}$$

Сравнивая эти формулы с (7.I.I0), приходим к выводу: коэффициент при z^j в многочленах $x^{(k)}(z)$ и $y^{(k)}(z)$ совпадает с j-й компонентой соответственно векторов $x^{(k)}$ и $y^{(k)}$; коэффициент при z^j в многочлене $\beta_i^{(k)}(z)$ совпадает с $n-2-j$-й компонентой вектора $d_i^{(k)}$ $(i=1,2,3,4)$. Заметим также, что алгоритм, реализующий непосредственно формулы (8.I.I)-(8.I.4), содержит некоторые избыточные вычисления по сравнению с (7.I.I0). Тем не менее, чтобы построить быстрый алгоритм, мы будем опираться именно на формулы (8.I.I)-(8.I.4). В интересах сокращения записи положим

$$M^k = \left\{\beta_i^{(k)}(z), \quad i=1,2,3,4\right\}. \tag{8.I.5}$$

Введем в рассмотрение матрицы

$$R^{kl}(z) = \left[r_{ij}^{kl}(z)\right]_{ij=0}^{1} = \begin{bmatrix} I & t_{k+1} \\ s_{k+1} z & z \end{bmatrix} \begin{bmatrix} I & t_{k+2} \\ s_{k+2} z & z \end{bmatrix} \cdots \begin{bmatrix} I & t_l \\ s_l z & z \end{bmatrix}, \tag{8.I.6}$$

$$U^{kl}(z) = \left[u_{ij}^{kl}(z)\right]_{ij=0}^{1} = \begin{bmatrix} z & t_{k+1} z \\ s_{k+1} & I \end{bmatrix} \begin{bmatrix} z & t_{k+2} z \\ s_{k+2} & I \end{bmatrix} \cdots \begin{bmatrix} z & t_l z \\ s_l & I \end{bmatrix}, \tag{8.I.7}$$

Легко видеть, что

$$U^{kl}(z) = z^{l-k} R^{kl}\left(\frac{1}{z}\right), \qquad (8.I.8)$$

т.е. коэффициенты многочлена $u_{ij}^{kl}(z)$ - это взятые в обратном порядке коэффициенты многочлена $r_{ij}^{kl}(z)$.

Кроме того, согласно (8.I.2) находим

$$[x^{(k)}(z) \quad y^{(k)}(z)] = [I \quad I]\, R^{0k}(z). \qquad (8.I.9)$$

Следовательно, для многочленов $x^{(n-1)}(z)$, $y^{(n-1)}(z)$, которые, собственно, и составляют цель вычислений, получаем выражения

$$\begin{aligned} x^{(n-1)}(z) &= r_{00}^{(0,n-1)}(z) + r_{10}^{(0,n-1)}(z), \\ y^{(n-1)}(z) &= r_{01}^{(0,n-1)}(z) + r_{11}^{(0,n-1)}(z). \end{aligned} \qquad (8.I.I0)$$

Таким образом, основной задачей мы можем считать вычисление полиномиальной матрицы $R^{0,n-1}(z)$. Здесь-то и обнаруживается возможность дихотомии.

Итак, предположим, что $n = 2^L$, $L \geq 1$, и пусть по заданным M_0, p_0, q_0 требуется получить $R^{0,n-1}(z)$, p_{n-1}, q_{n-1}. Запишем

$$(R^{0,n-1},\ p_{n-1},\ q_{n-1}) = F(n, M^0, p_0, q_0). \qquad (8.I.II)$$

Алгоритм F определим следующим образом.

I. Если $n = 2$, то R^{01}, p_1, q_1 вычисляем в соответствии с (7.I.I0), (8.I.6).

2. Если $n \geq 4$, то положим $m = n/2$, вычислим R^{01}, p_1, q_1 по формулам (7.I.I0), (8.I.6) и образуем новые многочлены степени не выше $m-2$:

$$\hat{M}^1 = \left\{ \sum_{j=0}^{m-2} d^{(1)}_{i,m-2-j}\, z^j, \quad i = 1, 2, 3, 4 \right\}$$

3. $(R^{1m},\ p_m,\ q_m) = F(m,\ \hat{M}^1,\ p_1,\ q_1)$

4. Вычислим новые многочлены степени не выше $m-2$:

$$\hat{M}^m = \left\{ \sum_{j=0}^{m-2} \beta^{(m)}_{i,m+j}\, z^j, \quad i = 1, 2, 3, 4 \right\},$$

где

$$[\beta_1^{(m)}(z)\ \beta_2^{(m)}(z)] = [\beta_1^{(1)}(z)\ \beta_2^{(1)}(z)]\, R^{1m}\left(\frac{1}{z}\right) z^{m-1},$$

$$[\beta_3^{(m)}(z)\ \beta_4^{(m)}(z)] = [\beta_3^{(1)}(z)\ \beta_4^{(1)}(z)]\, R^{1m}(z).$$

5. $(R^{m,n-1},\ p_{n-1},\ q_{n-1}) = F(m,\ \hat{M}^m,\ p_m,\ q_m)$.

6. $R^{0,n-1} = R^{01} R^{1m} R^{m,n-1}$

Как видим, исходная задача сводится к двум аналогичным задачам вдвое меньшего размера (п. 3, 5). Вычислительные затраты на сведение определяются способом выполнения действий в п. 4, 6. При умножении многочленов естественно воспользоваться алгоритмами быстрой свертки; тогда затраты на сведение составят $O(p^2 n \log_2 n + p^3 n)$ арифметических операций. Поэтому алгоритм F реализуется ценой $O(p^2 n \log_2^2 n + p^3 n \log_2 n)$ операций.

Мы попытаемся выполнить редукцию как можно более экономичным способом. Для этого примем во внимание следующие утверждения.

Лемма 8.I.I. Пусть $a(z)$ и $b(z)$ - многочлены соответственно степени не выше $2m-1$ и $m-1$. Рассмотрим их произведение $c(z) = b(z)a(z)$ и многочлен $\hat{c}(z)$ степени не выше $2m-1$, интерполирующий его в точках

$$z_j = \exp\left(i\, \frac{2\pi}{2m}\, j\right), \quad j = 0, 1, \ldots, 2m-1, \qquad (8.I.I2)$$

здесь i - мнимая единица. Тогда многочлены $c(z)$ и $\hat{c}(z)$ имеют одинаковые коэффициенты при z^j, где $m \leq j \leq 2m-1$.

Доказательство. Воспользуемся следующей матрично-векторной записью правила умножения многочленов:

$$u_{ij}^{2m}(z_l) = (-\varepsilon)^l\, r_{ij}(z_{2m-l}),\quad l = 1,\dots,2m-1,$$

где $\varepsilon = \exp\left(-i\,\frac{2\pi}{n}\right)$; $\quad i$ - мнимая единица.

Эти формулы вытекают непосредственно из (8.I.8).

Понятно, что обе задачи - вычисление значений многочлена в точках z_l $(l = 0,1,\dots,n-1)$ и получение коэффициентов интерполяционного многочлена степени не выше $n-1$ по значениям в этих же точках - решаются выполнением обратного или прямого преобразования Фурье порядка n. Нетрудно подсчитать, что редукция с учетом леммы I, 2 потребует выполнения 20 преобразований Фурье порядка n. Если ориентироваться на сплит-алгоритмы, то затраты на них составят 20×(I/3)$p^2 n\log_2 n$ комплексных умножений и 20$p^2 n\log_2 n$ комплексных сложений-вычитаний. Примем во внимание соотношение

$$n\log_2 n + 2\cdot\frac{n}{2}\log_2\frac{n}{2} + 2^2\,\frac{n}{2^2}\log_2\frac{n}{2^2} + \dots \sim \frac{1}{2}\,n\log_2^2 n.$$

Поэтому описанный выше алгоритм F может быть реализован с затратой (I0/3)$p^2 n\log_2^2 n + O(p^3 n\log_2 n)$ комплексных умножений и I0$p^2 n\log_2^2 n + O(p^3 n\log_2 n)$ комплексных сложений-вычитаний.

Заметим, что полиномиальную матрицу $R^{0,n-1}(z)$ в алгоритме F мы находим по предварительно вычисленным матрицам $R^{0,n-1}(z_{nl})$, где

$$z_{nl} = \exp\left(i\,\frac{2\pi l}{n}\right),\quad l = 0,1,\dots,n-1; \qquad (8.I.I3)$$

i - мнимая единица. Это последнее действие требует, очевидно, лишь $O(n\log_2 n)$ операций. Поэтому с нашей целевой функцией - найти эффективный способ вычисления многочленов $x^{(n-1)}(z)$ и $y^{(n-1)}(z)$ - вполне согласуется рассмотрение следующей задачи:

- по заданным n, M^0, p_0, q_0 вычислить

$$\left\{R^{0,n-1}(z_{nl}),\ l = 0,1,\dots,n-1\right\},\ p_{n-1},\ q_{n-1}.$$

Пользуясь терминологией, связанной с преобразованием Фурье, можно сказать, что мы делаем попытку перевести вычисления в "частотную область". Запишем

$$\left(\left\{R^{0,n-1}(z_{nl})\right\},\ p_{n-1},\ q_{n-1}\right) = F_1(n; M_0, p_0, q_0). \qquad (8.I.I4)$$

$$\begin{bmatrix} c_0 \\ c_1 \\ \dots \\ c_{m-1} \\ c_m \\ \dots \\ c_{2m-1} \end{bmatrix} = \begin{bmatrix} b_0 & & & & & & \\ b_1 & b_0 & & & 0 & & \\ \dots & & & & & & \\ b_{m-1} & \dots & b_1 & b_0 & & & \\ & b_{m-1} & \dots & b_1 & b_0 & & \\ & & & \dots & & & \\ & 0 & & & b_{m-1} & \dots & b_1 \ \ b_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{m-1} \\ a_m \\ \dots \\ a_{2m-1} \end{bmatrix}. \qquad (8.I.I5)$$

Коэффициенты интерполяционного многочлена $\hat c(z)$ найдем с помощью спектральной теоремы для циркулянтных матриц (см. § I):

$$\begin{bmatrix} \hat c_0 \\ \hat c_1 \\ \dots \\ \hat c_{m-2} \\ \hat c_{m-1} \\ \hat c_m \\ \dots \\ \hat c_{2m-1} \end{bmatrix} = \begin{bmatrix} b_0 & & & & & b_{m-1} & \dots & b_2 & b_1 \\ b_1 & b_0 & . & & . & & & b_{m-1} \dots & b_2 \\ \dots & \dots & & . & & & & \dots & \dots \\ b_{m-2} & \dots & b_1 & b_0 & & 0 & & & b_{m-1} \\ b_{m-1} & b_{m-2} \dots & & b_1 & b_0 & & & & \\ & b_{m-1} & b_{m-2} \dots & & b_1 & b_0 & & & \\ & & & \dots & & & & & \\ & 0 & & & b_{m-1} & b_{m-2} \dots & b_1 & b_0 & \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{m-2} \\ a_{m-1} \\ a_m \\ \dots \\ a_{2m-1} \end{bmatrix}. \qquad (8.I.I6)$$

В (8.I.I3) матрица тёплицева и притом ленточная, в (8.I.I4) - циркулянтная. Сравнивая эти соотношения, находим $\hat c_j = c_j$ при $m \le j \le 2m-1$. Лемма доказана.

Лемма 8.I.2. Пусть z_l определяются согласно (8.I.I2). Тогда значения многочленов $u_{ij}^{2m}(z_l)$ выражаются через $r_{ij}^{2m}(z_l)$ по формулам

$$u_{ij}^{2m}(z_0) = r_{ij}^{2m}(z_0), \qquad (8.I.I7)$$

Алгоритм F_1 сформулируем следующим образом.

1. Если $n = 2$, то $\{R^{01}(z_{2l})\}$, p_1, q_1, вычисляем, используя (7.1.10), (8.1.6).

2. Если $n \geq 4$, то положим $m = n/2$, вычислим

$$R^{01}(z_{nl}), \quad l=0,1,\dots,n-1, \quad p_1, q_1, M^1$$

в согласии с (7.1.10), (8.1.6) и образуем $\hat{M}^1$ (новые многочлены степени не выше $m-2$) так же, как в п. 2 алгоритма F.

3. $(\{R^{1m}(z_{mj})\}, p_m, q_m) = F_1(m, \hat{M}^1, p_1, q_1)$.

4. По значениям $R^{1m}(z_{ml})$, $l=0,1,\dots,m-1$, вычислим $R^{1m}(z)$.

5. Вычислим значения $R^{1m}(z_{nl})$, $l=0,1,\dots,n-1$.

6. Вычислим значения $\beta_i^{(1)}(z_{nl})$, $l=0,1,\dots,n-1$; $i=1,2,3,4$.

7. $[\beta_1^{(m)}(z_{nl})\beta_2^{(m)}(z_{nl})] = [\beta_1^{(1)}(z_{nl})\beta_2^{(1)}(z_{nl})]\, z_{nl}^{m-1} R^{1m}\left(\frac{1}{z_{nl}}\right)$,

$[\beta_3^{(m)}(z_{nl})\beta_4^{(m)}(z_{nl})] = [\beta_3^{(1)}(z_{nl})\beta_4^{(1)}(z_{nl})]\, R^{1m}(z_{nl})$,

$l=0,1,\dots,n-1$.

8. Вычислим многочлены $\hat{\beta}_i(z)$ степени не выше $n-1$, принимающие в точках z_{nl}, $l=0,1,\dots,n-1$, соответственно значения $\{\beta_i^{(m)}(z_{nl})\}$, $i=1,2,3,4$. Образуем новые многочлены степени не выше $m-2$:

$$\hat{M}^m = \left\{\sum_{j=0}^{m-2} \hat{\beta}_{i,m+j}\, z^j, \quad i=1,2,3,4\right\}.$$

9. $(\{R^{m,n-1}(z_{mj})\}, p_{n-1}, q_{n-1}) = F_1(m, \hat{M}^m, p_m, q_m)$.

10. По значениям $R^{m,n-1}(z_{ml})$, $l=0,1,\dots,m-1$, вычислим $R^{m,n-1}(z)$.

11. Вычислим значения $R^{m,n-1}(z_{nl})$, $l=0,1,\dots,n-1$.

12. $R^{0,n-1}(z_{nl}) = R^{01}(z_{nl})R^{1m}(z_{nl})R^{m,n-1}(z_{nl})$, $l=0,1,\dots,n-1$.

Поясним, каким образом выполняется п. 5. Пусть F_n обозначает блочную матрицу обратного дискретного преобразования Фурье порядка n:

$$F_n = \left\{\exp\left(i\frac{2\pi}{n}kl\right)I\right\}_{kl=0}^{n-1}$$

i - мнимая единица. Тогда в п. 5 четыре раза реализуется следующая матрично-векторная операция:

$$v = \begin{bmatrix} v_0 \\ \dots \\ v_{2m-1} \end{bmatrix} = F_n \begin{bmatrix} w_0 \\ \dots \\ w_{m-1} \\ 0 \\ \dots \\ 0 \end{bmatrix}. \tag{8.1.18}$$

При этом следует учесть, что вектор $F_m[w_0 \dots w_{m-1}]'$ является известным (это исходный вектор для п. 4).

Обозначим через P матрицу перестановки, которая в векторе Pv собирает в первых позициях блочные компоненты из v с четными номерами, а затем - с нечетными. Тогда имеем (см. также /75/)

$$PF_n = \begin{bmatrix} F_m & F_m \\ F_m D & -F_m D \end{bmatrix}, \tag{8.1.19}$$

где D - диагональная матрица из поворачивающих множителей. Таким образом,

$$v = P'\begin{bmatrix} F_m & w \\ F_m & Dw \end{bmatrix}, \qquad w = [w_0 \dots w_{m-1}]'. \tag{8.1.20}$$

Значит, при вычислении v реализуется одно преобразование Фурье порядка m. Всего в п. 5 требуется выполнить четыре преобразования Фурье порядка m. Аналогичное замечание нужно учесть и при выполнении п. 11.

Вычислительные затраты на редукцию, определяющие главный член (по зависимости от n), отвечают пп. 4, 5, 6, 8, 10, 11. В пп. 6, 8 требуется выполнить восемь преобразований Фурье порядка n. В каждом из пп. 4, 5, 10, 11 реализуется по четыре преобразования Фурье порядка m , т.е. всего шестнадцать преобразований Фурье порядка m. Поскольку одно преобразование порядка n по затратам (в главном члене) равносильно двум преобразованиям порядка $m = n/2$, общие затраты на редукцию определяются выполнением шестнадцати преобразований Фурье порядка n . Ориентируясь на использование сплит-алгоритмов, заключаем, что алгоритм F_1 требует выполнения $(8/3)p^2 n \log_2^2 n$ комплексных умножений и $8p^2 n \log_2^2 n$ комплексных сложений-вычитаний (в главном члене). К этому еще добавляются $O(p^3 n \log_2 n)$ операций. В вещественном случае затраты сокращаются вдвое.

Теперь предположим дополнительно, что матрица A блочно-симметрична. Тогда для всех k имеет место равенство $s_k = t_k$, вследствие чего блочные коэффициенты многочленов $r_{00}^{kl}(z)$ и $r_{11}^{kl}(z)$ одинаковы соответственно при мономах z^j и z^{l-k-j} $(j=0,1,\ldots,l-k)$; то же справедливо по отношению к многочленам $r_{01}^{kl}(z)$ и $r_{10}^{kl}(z)$. Таким образом, можно ограничиться вычислением, скажем, многочленов $r_{00}^{kl}(z)$ и $r_{01}^{kl}(z)$. Кроме того, из четырех многочленов $\beta_i^{(k)}$ $(i=1,2,3,4)$ достаточно вести вычисление двух многочленов $\beta_3^{(k)}$ и $\beta_4^{(k)}$ (или $\beta_1^{(k)}$ и $\beta_2^{(k)}$). После исключения из F_1 избыточных операций мы получим алгоритм F_2 , для которого число комплексных умножений равно $(4/3) \times p^2 n \log_2^2 n$, а число комплексных сложений-вычитаний равно $4p^2 n \log_2^2 n$ (в главном члене по зависимости от n). Алгоритм такой же трудоёмкости можно получить и в более общем случае, когда блоки матрицы A удовлетворяют условию

$$a_l = g a_{-l} h, \quad l = 0, 1, \ldots, n-1, \tag{8.I.2I}$$

где

$$g^2 = h^2 = I. \tag{8.I.22}$$

В этом случае

$$\begin{gathered} p_k = h q_k g, \quad s_k = h t_k h, \\ d_1^{(k)} = g d_4^{(k)} h, \quad d_2^{(k)} = g d_3^{(k)} h, \\ x_j^{(k)} = h y_{k-j}^{(k)} h, \quad j = 0, 1, \ldots, k. \end{gathered} \tag{8.I.23}$$

Если рассматриваемые матрицы и блоки вещественные, то затраты снижаются вдвое.

Обратим внимание на то, что в построенных алгоритмах в действительности вычисляются p_k, q_k, s_k, t_k для всех k , только

это происходит в другой последовательности - в отличие от алгоритма (7.I.I0). Пусть $p = 1$, т.е. матрица A неблочная. Легко видеть, что в этом случае

$$p_k = q_k = \det A_k \,/\, \det A_{k-1}. \tag{8.I.24}$$

Поэтому быстрые алгоритмы можно использовать и для вычисления определителей. В эрмитовом случае это может послужить базой для алгоритма бисекций, вычисляющего собственные значения.

Чтобы записать тёплицево разложение матрицы A^{-1} (см. § 5), достаточно иметь ее первые и последние блочные столбцы и строки. Описанные здесь быстрые алгоритмы позволяют получить первый и последний блочные столбцы. Если $p = 1$, то в силу персимметричности матрицы A^{-1} ее первая и последняя строки состоят из взятых в обратном порядке компонент соответственно последнего и первого столбцов. Если $p > 1$, то требуются дополнительные вычисления: первую и последнюю блочные строки мы находим очевидным образом, получив первый и последний блочные столбцы для $(A^T)^{-1}$.

8.2. Быстрые алгоритмы для ганкелевых систем

Пусть $A = A_{n-1}$ - ганкелева матрица и все ее ведущие подматрицы A_k $(0 \le k \le n-1)$ обратимы. Тогда если известны последние столбцы в A_{n-1}^{-1} и A_{n-2}^{-1} , то решение системы с матрицей A можно найти с затратой $O(n \log_2 n)$ операций (см. /I7/). В блочном случае нужно иметь для этого еще и последние блочные строки в A_{n-1}^{-1} и A_{n-2}^{-1} (заметим, что они легко получаются из последних столбцов матриц $(A_{n-1}^T)^{-1}$ и $(A_{n-2}^T)^{-1}$). Таким образом, чтобы ускорить решение системы, достаточно найти быстрый способ вычисления последних (блочных) столбцов в A_{n-1}^{-1} и A_{n-2}^{-1} . Мы сделаем это, взяв за основу векторный алгоритм (7.2.7).

Введем следующие блочные многочлены от переменной z :

$$\begin{gathered} y^{(k)}(z) = \sum_{j=0}^{k} y_j^{(k)} z^j, \\ w^{(k)} = \sum_{j=0}^{2n-1} w_j^{(k)} z^j = \sum_{j=0}^{2n-2} v_{2n-1-j}^{(k)} z^j, \\ k = 0, 1. \end{gathered} \tag{8.2.I}$$

При $k \geq 2$ многочлены $y^{(k)}(z)$ и $w^{(k)}(z)$ определим с помощью рекуррентных соотношений

$$\left[y^{(k-1)}(z)\ y^{(k)}(z)\right] = \left[y^{(k-2)}(z)\ y^{(k-1)}(z)\right] \begin{bmatrix} 0 & \beta_k \\ I & \alpha_k + I\cdot z \end{bmatrix}. \qquad (8.2.2)$$

$$\left[w^{(k-1)}(z)\ w^{(k)}(z)\right] = \left[w^{(k-2)}(z)\ w^{(k-1)}(z)\right] \begin{bmatrix} 0 & \beta_k \\ I & \alpha_k + I\cdot z \end{bmatrix}. \qquad (8.2.3)$$

Тогда коэффициенты многочленов $y^{(k)}(z)$ будут совпадать с соответствующими компонентами векторов $y^{(k)}$, а для многочленов $w^{(k)}(z)$ имеем $w_j^{(k)} = v_{2n-2-j}^{(k)}$ при $0 \leq j \leq 2n-2$.

При $1 \leq k < l \leq n-1$ введем полиномиальные матрицы

$$R^{kl}(z) = \left[r_{ij}^{kl}(z)\right]_{ij=0}^{1} =$$

$$= \begin{bmatrix} 0 & \beta_{k+1} \\ I & \alpha_{k+1} + Iz \end{bmatrix} \begin{bmatrix} 0 & \beta_{k+2} \\ I & \alpha_{k+2} + Iz \end{bmatrix} \cdots \begin{bmatrix} 0 & \beta_l \\ I & \alpha_l + Iz \end{bmatrix}. \qquad (8.2.4)$$

Тогда

$$\left[y^{(n-2)}(z)\ y^{(n-1)}(z)\right] = \left[I\ \ -a_0^{-1}a_1 + Iz\right] R^{1,n-1}(z); \qquad (8.2.5)$$

т.е. нам достаточно найти быстрый способ вычисления полиномиальной матрицы $R^{1,n-1}(z)$.

Итак, поставим следующую задачу: по заданным $n = 2^L$, $L \geq 2$, и многочленам $w^{(0)}(z)$ и $w^{(1)}(z)$ степени не выше $2n-2$ вычислить $R^{1,n-1}(z)$. Запишем

$$R^{1,n-1}(z) = \mathfrak{F}\left(n,\ w^{(0)}(z),\ w^{(1)}(z)\right). \qquad (8.2.6)$$

Алгоритм $\mathfrak{F}$ определим следующим обарзом.

1. Если $n = 4$, то $R^{13}(z)$ вычисляется в соответствии с (7.2.7), (8.2.4).

2. При $n \geq 8$ положим $m = n/2$, вычислим $w^{(2)}(z)$, $w^{(3)}(z)$ и $R^{13}(z)$ по формулам (7.2.7), (8.2.3), (8.2.4) и образуем новые многочлены степени не выше $2m-2$:

$$\hat{w}^{(0)}(z) = \sum_{j=0}^{2m-2} w_{n-2-j}^{(2)} z^j, \qquad \hat{w}^{(1)}(z) = \sum_{j=0}^{2m-2} w_{n-2-j}^{(3)} z^j.$$

3. $R^{3,m+1}(z) = \mathfrak{F}\left(m,\ \hat{w}^{(0)}(z),\ \hat{w}^{(1)}(z)\right)$.

4. Вычислим новые многочлены степени не выше $2m-2$:

$$\tilde{w}^{(0)}(z) = \sum_{j=0}^{2m-2} w_{m+j}^{(m)} z^j, \qquad \tilde{w}^{(1)}(z) = \sum_{j=0}^{2m-2} w_{m+j}^{(m+1)} z^j,$$

где

$$\left[w^{(m)}(z)\ w^{(m+1)}(z)\right] = \left[w^{(2)}(z)\ w^{(3)}(z)\right] R^{3,m+1}(z).$$

5. $R^{m+1,n-1}(z) = \mathfrak{F}\left(m,\ \tilde{w}^{(0)}(z),\ \tilde{w}^{(1)}(z)\right)$.

6. $R^{1,n-1}(z) = R^{13}(z)\, R^{3,m+1}(z)\, R^{m+1,n-1}(z)$.

Как видим, в алгоритме $\mathfrak{F}$ реализуется дихотомия исходной задачи: в пп.3, 5 решается та же задача вдвое меньшего размера. Если при умножении многочленов используется быстрая свертка, то общие затраты составят $O(p^2 n \log_2^2 n + p^3 n \log_2 n)$ операций.

Уточним способ реализации п. 4. При $k = 2, 3$ положим

$$W^{(k)}(z) = \sum_{j=0}^{n-1} w_j^{(k)} z^j, \qquad \hat{W}^{(k)}(z) = \sum_{j=0}^{n-1} w_{m+j}^{(k)} z^j$$

и определим многочлены $W^{(k)}(z)$, $\hat{W}^{(k)}(z)$ при $k = m,\ m+1$ по следующим формулам:

$$\left[W^{(m)}(z)\ W^{(m+1)}(z)\right] = \left[W^{(2)}(z)\ W^{(3)}(z)\right] R^{3,m+1}(z),$$

$$\left[\hat{W}^{(m)}(z)\ \hat{W}^{(m+1)}(z)\right] = \left[\hat{W}^{(2)}(z)\ \hat{W}^{(3)}(z)\right] R^{3,m+1}(z). \qquad (8.2.7)$$

Тогда

$$w_{m+j}^{(m)} = W_{m+j}^{(m)}, \quad w_{m+j}^{(m+1)} = W_{m+j}^{(m+1)}, \quad j = 0, 1, \ldots, m-1;$$

$$w_{m+j}^{(m)} = \hat{W}_{m+j}^{(m)}, \quad w_{m+j}^{(m+1)} = \hat{W}_{m+j}^{(m+1)}, \quad j = m, m+1, \ldots, 2m-1. \qquad (8.2.8)$$

Далее, пусть $U^{(m)}(z)\,U^{(m+1)}(z)\,\hat{U}^{(m)}(z),\ \hat{U}^{(m+1)}(z)$ — многочлены степени не выше $n-1$, интерполирующие $W^{(m)}(z)$, $W^{(m+1)}(z)$, $\hat{W}^{(m)}(z)$, $\hat{W}^{(m+1)}(z)$ соответственно в точке z_j вида (8.I.I2). Тогда согласно лемме 8.I.I имеем

$$W_{m+j}^{(m)}=U_{m+j}^{(m)},\qquad W_{m+j}^{(m+1)}=U_{m+j}^{(m+1)},$$

$$\hat{W}_{m+j}^{(m)}=\hat{U}_{m+j}^{(m)},\qquad \hat{W}_{m+j}^{(m+1)}=\hat{U}_{m+j}^{(m+1)},\quad j=0,1,\dots,m-1. \tag{8.2.9}$$

Таким образом, в п. 4 для вычисления значений многочленов потребуется выполнить восемь дискретных преобразований Фурье порядка n; при решении задачи интерполяции потребуется выполнить четыре преобразования Фурье порядка n. При реализации п. 6 сначала находим значения $R^{m+1,n+1}(z_j)$, затем $R^{1,n-1}(z_j)$, после чего вычисляем коэффициенты многочленов $r_{ij}^{1,n-1}(z)$. Все это требует выполнения восьми преобразований Фурье порядка n. Итак, вычислительные затраты (в главном члене по зависимости от n), отвечающие пп.4, 6, определяются выполнением двадцати преобразований Фурье порядка n. Если ориентироваться на сплит-алгоритмы, то это будет $(20/3)p^2 n\log_2^2 n$ комплексных умножений и $20p^2 n\log_2^2 n$ комплексных сложений-вычитаний, идущих на осуществление редукции. В целом реализация алгоритма $\mathcal{F}$ будет связана с выполнением $(10/3)p^2 n\log_2^2 n + O(p^3 n\log_2 n)$ комплексных умножений и $10p^2 n\log_2^2 n + O(p^3 n\log_2 n)$ комплексных сложений-вычитаний. В вещественном случае затраты можно сократить вдвое.

Построим ниже еще один быстрый алгоритм для обращения ганкелевой матрицы, связанный с вычислением первых и последних столбцов матриц A_k^{-1} - обозначим их соответственно $x^{(k)}$ и $y^{(k)}$. Если векторы $x^{(n-1)}$ и $y^{(n-1)}$ известны, то решение системы с матрицей A дополнительно потребует выполнения лишь $O(n\log_2 n)$ операций (см. /I7/). Рассмотрим следующий алгоритм, использующий, в отличие от (5.2.I0), двучленные рекуррентные соотношения (см. /94/):

$$k=0:\quad x_0^{(0)}=y_0^{(0)}=a_0^{-1};$$

$$k=1,\dots,n-1:\qquad s_k=\sum_{j=0}^{k-1}a_{k+j}\,x_j^{(k-1)},$$

$$t_k=\sum_{j=0}^{k-1}a_{k+1+j}\,x_j^{(k-1)},\quad u_k=\sum_{j=0}^{k-1}a_{kj}\,y_j^{(k-1)}, \tag{8.2.10}$$

$$\alpha_k=(t_k-u_k s_k)^{-1},\qquad \beta_k=s_k\alpha_k,$$

$$y^{(k)}=\uparrow x^{(k-1)}\alpha_k-\uparrow y^{(k-1)}\beta_k, \tag{8.2.10}$$

$$x^{(k)}=\uparrow x^{(k-1)}-y^{(k)}s_k.$$

Очевидно, это T -алгоритм; преобразуем его в векторный алгоритм. В соответствии с методикой, предложенной в § 7, введем для этого дополнительные векторы размерности $2n-1$:

$$\varphi^{(k)}=H[x^{(k)}\ 0\dots0]',\qquad \psi^{(k)}=H[y^{(k)}\ 0\dots0]', \tag{8.2.11}$$

где H - ганкелева матрица порядка $2n-1$ с элементами h_{i+j}, имеющими вид

$$h_{i+j}=\begin{cases}a_{i+j}, & 0\le i+j\le 2n-2,\\ 0, & 2n-2< i+j\le 4n-4.\end{cases}$$

Чтобы получить векторный алгоритм, достаточно учесть следующие соотношения:

$$s_k=\varphi_k^{(k-1)},\qquad t_k=\varphi_{k+1}^{(k-1)},\qquad u_k=\psi_k^{(k-1)},$$

$$\psi^{(k)}=\uparrow\varphi^{(k-1)}\alpha_k-\psi^{(k-1)}\beta_k, \tag{8.2.12}$$

$$\varphi^{(k)}=\varphi^{(k-1)}-\psi^{(k)}s_k.$$

Далее, введем многочлены

$$x^{(0)}(z)=y^{(0)}(z)=a_0^{-1},$$

$$\Phi^{(0)}(z)=\sum_{j=0}^{2n-2}\Phi_j^{(0)}z^j=\sum_{j=0}^{2n-2}\varphi_{2n-2-j}^{(0)}z^j, \tag{8.2.13}$$

$$\Psi^{(0)}(z)=\sum_{j=0}^{2n-2}\Psi_j^{(0)}z^j=\sum_{j=0}^{2n-2}\psi_{2n-2-j}^{(0)}z^j$$

и положим

$$\left[\Phi^{(k)}(z)\,\Psi^{(k)}(z)\right]=\left[\Phi^{(k-1)}(z)\,\Psi^{(k-1)}(z)\right]\begin{bmatrix} 1-\alpha_k s_k z & \alpha_k z \\ \beta_k s_k & -\beta_k \end{bmatrix}, \qquad (8.2.14)$$

$$\left[x^{(k)}(z)\,y^{(k)}(z)\right]=\left[x^{(k-1)}(z)\,y^{(k-1)}(z)\right]\begin{bmatrix} 1-\alpha_k s_k z & -\alpha_k z \\ \beta_k s_k & -\beta_k \end{bmatrix}.$$

Следовательно,

$$\left[x^{(n-1)}(z)\,y^{(n-1)}(z)\right]=\left[a_0^{-1}\;\; a_0^{-1}\right]R^{0,n-1}(z), \qquad (8.2.15)$$

где

$$R^{kl}(z)=\begin{bmatrix} 1-\alpha_{k+1} s_{k+1} z & \alpha_{k+1} z \\ \beta_{k+1} s_{k+1} & -\beta_{k+1} \end{bmatrix}\cdots\begin{bmatrix} 1-\alpha_l s_l z & \alpha_l z \\ \beta_l s_l & -\beta_l \end{bmatrix}, \qquad (8.2.16)$$

$$0\le k<l\le n-1.$$

Поставим задачу: по заданным n, $\Phi^{(0)}(z)$ и $\Psi^{(0)}(z)$ вычислить $R^{0,n-1}(z)$. Ее дихотомия осуществляется подобно тому, как это сделано в алгоритме $\mathfrak{F}$. В результате получается быстрый алгоритм, по вычислительным затратам аналогичный $\mathfrak{F}$. Заметим, что как для этого алгоритма, так и для $\mathfrak{F}$ можно построить модификации, переводящие вычисления в "частотную область", - по аналогии с алгоритмом F_1 для тёплицевых матриц.

Литература

1. Ахо, Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов.- М.: Мир, 1979.- 536 с.
2. Бабенко К.И. О тёплицевых и ганкелевых матрицах//УМН.- 1986. - Т. 41, № 1.- С. 171-178.
3. Бадева В., Морозов В.А. Алгоритмы быстрого и ускоренного решения некоторых специальных систем линейных алгебраических уравнений //Численный анализ на Фортране. Вып. 20.- М.: МГУ, 1977.- С. 80-88.
4. Балинский А.И., Ли Пон-ы, Об обращении ганкелевых и тёплицевых матриц//Мат.методы и физ.-мех.поля.- 1979.- № 9.- С. 31-37.
5. Банч Дж.Р. К решению тёплицевых систем линейных уравнений //Вычис.методы линейной алгебры: Тр. Всес.конф.- М.: ОВМ АН СССР, 1983.- С. 7-24.
6. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы.- М.: Наука, 1987.- 600 с.
7. Белоцерковский С.М., Лифанов И.К. Численные методы в сингулярных интегральных уравнениях.- М.: Наука, 1985.- 256 с.
8. Белоцерковский С.М., Лифанов И.К., Ништ М.И. Метод дискретных вихрей в задачах аэродинамики и теория многомерных сингулярных интегральных уравнений//VI Международная конференция по численным методам в гидродинамике.- Тбилиси.- 1978.- С. 30-34.
9. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. - М.: Мир, 1989.- 448 с.
10. Блехер П.М. Обращение тёплицевых матриц//Тр. Моск.мат.общества.- 1979.- Т. 40.- С. 207-240.
11. Воеводин В.В. Вычислительные основы линейной алгебры.- М.: Наука, 1977.- 304 с.
12. Воеводин В.В., Кузнецов Ю.А. Матрицы и вычисления.- М.: Наука, 1984.- 320 с.
13. Воеводин В.В., Свешников А.Г., Тыртышников Е.Е. Эффективный численный метод решения интегрального уравнения II рода в задачах электродинамики//Вестн. МГУ. Вычисл.матем. и киберн.- 1980.- № 1. - С. 14-26.
14. Воеводин В.В., Тыртышников Е.Е. Численные методы решения задач с матрицами типа тёплицевых//ЖВМ и МФ.- 1981.- Т. 21, № 3. - С. 531-544.