A Superfast Version of the Split Schur Algorithm

Steven J. Kifowit
Northern Illinois University

Gregory S. Ammar
Northern Illinois University

We describe a new superfast algorithm for solving positive definite Toeplitz systems of equations. The new algorithm is based on a doubling procedure applied to a complex analog of the split Schur algorithm. This procedure computes the solution of the Yule-Walker equations by processing a family of split Levinson symmetric polynomials.

We consider the problem of solving a system of linear equations of the form

$$C_n r = [\sigma_n, 0, 0, \ldots, 0]^T, \tag{1}$$

where C_n is a Hermitian positive definite Toeplitz matrix of order $n+1$.

The matrix C_n has the following appearance:

$$C_n = \begin{bmatrix} c_0 & c_{-1} & c_{-2} & \cdots & c_{-n} \\ c_1 & c_0 & c_{-1} & \cdots & c_{1-n} \\ c_2 & c_1 & c_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & c_{-1} \\ c_n & c_{n-1} & \cdots & c_1 & c_0 \end{bmatrix},$$

where $c_{-j} = \bar{c}_j$.

The solution vector r and the real number σ_n are uniquely determined by the normalization $r_0 = 1$.

- The Levinson algorithm is commonly used to solve the system (1) and compute the Schur parameters associated with the matrix C_n.

- Delsarte and Genin have presented the split Levinson algorithm as a more efficient alternative to the Levinson algorithm.

- The split Levinson algorithm processes a sequence of polynomials p_k, where each polynomial possesses the symmetry property

$$p_k(z) = z^k \bar{p}_k(1/\bar{z}).$$

- The split Levinson symmetric polynomials satisfy the three-term recurrence relation

$$p_{k+1}(z) = (\alpha_k + \bar{\alpha}_k z) p_k(z) - z p_{k-1}(z). \qquad (2)$$

- The symmetric polynomials p_n and p_{n+1} determine the solution r of (1).

- Delsarte and Genin have presented split analogs of several of the classical algorithms in linear prediction theory.
  - The split Levinson algorithm
  - The split Schur algorithm
  - The split lattice algorithm
  - The normalized split lattice algorithm

- The split Schur algorithm can be used to generate the split Levinson symmetric polynomials, while computing the Schur parameters associated with the Toeplitz matrix C_n.

The Split Schur Algorithm

Input: An analytic function $\chi_0(z)$

$$
\begin{aligned}
&\alpha_0 = \tfrac{1}{\chi_0(0)} \\
&\lambda_0 = \infty \\
&\textbf{for } k = 0, 1, \ldots, \textbf{ while } \lambda_k > 0 \\
&\qquad \chi_{k+1}(z) = \bar{\alpha}_k + \tfrac{1}{z}\left[\alpha_k - \tfrac{1}{\chi_k(z)}\right] \\
&\qquad \alpha_{k+1} = \tfrac{1}{\chi_{k+1}(0)} \\
&\qquad \lambda_{k+1} = 2\mathrm{Re}(\alpha_k) - \tfrac{1}{\lambda_k} \\
&\textbf{end}
\end{aligned}
$$

The function $(1 - z) \cdot \chi_0$ is a Carathéodory function if and only if one of the following holds:

1. $\lambda_k > 0$ for $k = 1, 2, \ldots$

2. $\lambda_k > 0$ for $k = 1, 2, \ldots, n$, $\lambda_{n+1} = 0$, and $\chi_n(z) = 0$

If condition (2) holds, the χ_0 is a rational function.

Split Schur Algorithm

Input: $w_0(z)$, analytic in the unit disk

$$
\begin{aligned}
&w_{-1}(z) = 1 - z \\
&\alpha_0 = 1/w_0(0) \\
&\lambda_0 = \infty \\
&\textbf{for } k = 0, 1, 2, \ldots \ \textbf{ while } \lambda_k > 0 \\
&\quad w_{k+1}(z) = \overline{\alpha_k} w_k(z) + \frac{1}{z}\left[\alpha_k w_k(z) - w_{k-1}(z)\right] \\
&\quad \lambda_{k+1} = 2\text{Re}(\alpha_k) - \frac{1}{\lambda_k} \\
&\quad \alpha_{k+1} = \frac{w_k(0)}{w_{k+1}(0)} \\
&\textbf{end}
\end{aligned}
$$

The function w_0 is a Carathéodory function if and only if one of the following holds:

1. $\lambda_k > 0$ for $k = 1, 2, 3, \ldots$

2. $\lambda_k > 0$ for $k = 1, 2, \ldots, n$, $\lambda_{n+1} = 0$, and $w_{n+1}(z) \equiv 0$

- To solve the Toeplitz system (1), we apply the split Schur algorithm to the Carathéodory function defined by

$$w_0(z) = c_0 + 2\sum_{j=1}^{n} c_j z^j$$

- We define χ_{k+1} so that

$$\chi_{k+1}(z) = \frac{w_{k+1}(z)}{w_k(z)} \qquad k = -1, 0, 1, \ldots, n$$

- It follows from the split Schur recurrence relation that

$$\chi_k(z) = \frac{1}{(\alpha_k + \bar{\alpha}_k z) - z\chi_{k+1}(z)} \tag{3}$$

- Repeated use of (3) yields a continued fraction expansion for χ_0.

- If we define $T_k(\tau)$ such that $T_k(\chi_k)$ represents k steps of the continued fraction expansion, then we obtain

$$T_{k+1}(\tau) = \frac{u_{k+1}(z) + v_{k+1}(z)\tau}{w_{k+1}(z) + x_{k+1}(z)\tau} \tag{4}$$

  The polynomials u_k, v_k, w_k, and x_k collectively form two distinct families of symmetric polynomials satisfying (2).

- The polynomials w_k are precisely the symmetric polynomials computed in the split Levinson algorithm.

- In the case when $n = 2^p$, we may apply a doubling procedure to the continued fraction expansion. Upon doing so, we obtain an efficient method for computing the solution to (1) and all Schur parameters.

- The method processes two families of split Levinson symmetric polynomials, and is consequently rich in polynomial multiplication.

- These multiplications may be carried out efficiently by using fast Fourier transform techniques.

- In the real case, it appears that the solution to (1) can be computed in approximately $7n \log^2 n$ real arithmetic operations.