A FAST ALGORITHM FOR TOEPLITZ-BLOCK-TOEPLITZ LINEAR SYSTEMS

Andrew E. Yagle

Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2122 aey@eecs.umich.edu

ABSTRACT

A Toeplitz-block-Toeplitz (TBT) matrix is block Toeplitz with Toeplitz blocks. TBT systems of equations arise in 2-D interpolation, 2-D linear prediction and 2-D least-squares deconvolution problems. Although the doubly Toeplitz structure should be exploitable in a fast algorithm, existing fast algorithms only exploit the block Toeplitz structure, not the Toeplitz structure of the blocks. Iterative algorithms can employ the 2-D FFT, but usually take thousands of iterations to converge. We develop a new fast algorithm that assumes a smoothness constraint (described in the text) on the matrix entries. For an $M^2 \times M^2$ TBT matrix with M $M \times M$ Toeplitz blocks along each edge, the algorithm requires only $O(6M^3)$ operations to solve an $M^2 \times M^2$ linear system of equations; parallel computing on 2M processors can be performed on the algorithm as given. Two examples show the operation and performance of the algorithm.

1. INTRODUCTION

1.1. Formulation and Applications

The goal is to develop a fast algorithm to solve a Toeplitz-block-Toeplitz (TBT) linear system of equations. Specifically, the goal is to solve $Tx = y$, where

$$T = \begin{bmatrix} T_0 & T_{-1} & \dots & T_{1-M} \\ T_1 & T_0 & \dots & T_{2-M} \\ \vdots & \vdots & \ddots & \vdots \\ T_{M-1} & T_{M-2} & \dots & T_0 \end{bmatrix};$$

$$x = \begin{bmatrix} x_0^0 \\ \vdots \\ x_{M-1}^0 \\ \vdots \\ x_{M-1}^{M-1} \end{bmatrix}; \quad y = \begin{bmatrix} y_0^0 \\ \vdots \\ y_{M-1}^0 \\ \vdots \\ y_{M-1}^{M-1} \end{bmatrix} \qquad (1)$$

where each $M \times M$ block T_i itself has Toeplitz structure. The system (1) has M blocks, each of which is $M \times M$, for a total size of $M^2 \times M^2$. No assumptions of symmetry are required, i.e., $T_i \neq T_{-i}^H$.

TBT linear systems of equations (TBT systems) arise in several image processing problems. The problem of reconstructing an image from its 2-D Fourier transform values on a non-rectangular raster arises in magnetic resonance imaging (MRI) and synthetic aperture radar (SAR); the lack of a 2-D Lagrange interpolation formula requires its formulation as a 2-D least-squares interpolation problem, resulting in a TBT system. The Yule-Walker equations for 2-D linear prediction of a 2-D homogeneous random field are TBT. And 2-D least squares deconvolution of an image from a known spatially-invariant point-spread or blurring function also results in a TBT system. There are other applications.

1.2. Previous Work

Toeplitz structure (constant along diagonals) can readily be exploited using the Levinson algorithm. Block Toeplitz structure can be exploited using the multichannel Levinson algorithm; a linear system of equations with Toeplitz blocks can be rewritten as a block Toeplitz system. However, there does not seem to be any way to exploit the *doubly* Toeplitz TBT structure. The only published algorithms [1],[2] for TBT matrices exploit only the persymmetry (symmetry about the main antidiagonal) of the Toeplitz blocks, to reduce the computation required by the multichannel Levinson algorithm by one-half. These algorithms still require $O(M^5)$ operations, since the multichannel Levinson algorithm requires $O(M^2)$ *matrix* operations at $O(M^3)$ operations each. This is impractical for many image processing applications.

The reason that the Toeplitz structure of the blocks cannot be exploited in a Levinson-like algorithm is apparent from a scattering perspective on the multichannel Levinson algorithm. The relation between the multichannel reflection response at a given time to an impulse in a given channel (which constitutes the Toeplitz blocks) and the matrix reflection coefficients is nonlinear. Hence the low displacement rank of the Toeplitz blocks is quickly lost as the algorithm progresses. An exception occurs if the Toeplitz blocks are in fact circulant, since circulant structure is preserved under the basic matrix operations of addition, multiplication and inversion. Hence any fast algorithm for TBT systems must be non-Levinson-like in nature.

Since multiplication of a TBT matrix by a vector can be performed in $O(M^2 \log M)$ operations, iterative algorithms can be used to solve (1). However, iterative algorithms can take thousands of iterations to converge, and are not always parallelizable. A parallelizable algorithm with a known and small operation count is desirable.

1.3. New Approach

This paper reformulates the solution of a TBT system as a 2-D deconvolution problem in which the blurred image is only partially known (this problem is interesting in its own right). The blurring function defined by the entries of the TBT matrix is assumed to be lowpass, in that its 2-D discrete Fourier transform (DFT) is zero in the upper halfbands. This is not a significant restriction in practice, and it is reminiscent of the Calderon-Zygmund condition for wavelet-basis sparsification of matrices. A fast algorithm for extrapolation of bandlimited signals which we have developed [3],[4] and employed [5],[6] previously is used to extrapolate the unknown regions of the blurred image, and the known blurring function is deconvolved only at frequencies where it is nonzero. The extrapolation algorithm is used again to extrapolate the remaining frequencies of the image, which is the solution to the TBT system.

Section 2 illustrates this approach on purely Toeplitz systems, although the resulting algorithm is no faster than other approaches. Section 3 reviews the fast algorithm for bandwidth extrapolation developed in [3],[4]. Section 4 applies this algorithm to the solution of TBT systems. Section 5 presents two examples: (1) a small illustrative example; and (2) a more realistic example.

2. PURELY TOEPLITZ SYSTEMS

2.1. Deconvolution Formulation

We wish to solve the Toeplitz system

$$\begin{bmatrix} t(0) & \ddots & t(1-M) \\ \ddots & \ddots & \ddots \\ t(M-1) & \ddots & t(0) \end{bmatrix} \begin{bmatrix} x(0) \\ \vdots \\ x(M-1) \end{bmatrix} = \begin{bmatrix} y(0) \\ \vdots \\ y(M-1) \end{bmatrix} \tag{2}$$

Note that the Toeplitz matrix is specified by the $2M-1$ values $\{t(n), |n| \leq M-1\}$. This is one period of the periodic extension of order $2M-1$ of $t(n)$.

We can rewrite (2) as the *cyclic* convolution of order $2M-1$

$$t(n) * \tilde{x}(n) = \tilde{y}(n); \tag{3a}$$

$$\tilde{x}(n) = \{\underbrace{x(0)\dots x(M-1)}_{M\ values}, \underbrace{0\dots 0\dots 0}_{M-1\ values}\}; \tag{3b}$$

$$\tilde{y}(n) = \{\underbrace{y(0)\dots y(M-1)}_{M\ values}, \underbrace{z(M)\dots z(2M-2)}_{M-1\ values}\}, \tag{3c}$$

where $\{z(n), M \leq n \leq 2M-2\}$ denotes an *unknown* extension of $y(n)$. If we could somehow determine $z(n)$, the computation of $x(n)$ from $t(n)$ and $\{y(n), z(n)\}$ would become a simple 1-D deconvolution. However, there is no way to do this without solving another linear system, which would be more difficult than the original problem.

2.2. Bandlimited Assumption

Now suppose that $\{t(n), |n| \leq M-1\}$ is *bandlimited* to highest DTFT (discrete-time Fourier transform) frequency $\pi/2$. That is, define the discrete Fourier transform (DFT) $T(k)$ of order $2M-1$ of $t(n)$ as

$$T(k) = \sum_{n=0}^{2M-2} t(n) e^{-j2\pi nk/(2M-1)}, k = 0\dots 2M-2 \tag{4}$$

and suppose $T(k) = 0$ for $\frac{1}{4}(2M-1) < k < \frac{3}{4}(2M-1)$ (note that these bounds need not be integers). Recall that the DFT samples the DTFT on the unit circle, and implicitly creates periodic extensions.

One way that this can happen is to start with an arbitrary sequence of length M, upsample (insert a zero between successive numbers) to form a sequence of length $2M-1$ (this compresses and repeats the spectrum) and then interpolate the result with a discrete sinc function.

In practice, the Toeplitz matrix entries $t(n)$ define a reasonably smooth sequence which can be well approximated by M DFT components (including the conjugate symmetric components). If $t(n)$ is (say) a measured impulse response, it may even be desirable to filter the data in this way.

$t(n)$ is bandlimited, so extended sequence $\{y(n), z(n)\}$ is also bandlimited. Since we know M values $y(n)$ of a sequence bandlimited to M nonzero frequency values, we should be able to reconstruct $z(n)$ from $y(n)$ using the algorithm described in Section 3.

2.3. Purely Toeplitz Procedure

To solve the purely Toeplitz system (2), where $t(n)$ is bandlimited as described above, proceed as follows:

1. Extrapolate $y(n)$ to $\{y(n), z(n)\}$ using the algorithm described in Section 3;
2. Compute the DFT of $\{y(n), z(n)\}$ and of $t(n)$, and divide at those frequencies where both are nonzero;
3. Extrapolate *the DFT of* $x(n)$ since $x(n)$ is *time*-limited. Compute the inverse DFT of the result to obtain $x(n)$.

This procedure requires $2M^2 + 3M\log_2 M + 3M$ real multiplications-and-adds (MADS), which compares favorably to the Levinson algorithm for an asymmetric Toeplitz system with an arbitrary right side. Furthermore, the algorithm can almost be written out in closed form. However, the large numbers arising in the bandwidth extrapolation algorithm for large M (see Section 3) make this impractical for large M. But this is much less of a problem for TBT systems (see Section 4).

3. BANDWIDTH EXTRAPOLATION

3.1. Problem Formulation

We quickly summarize the results of [3],[4] which present a fast algorithm for exact extrapolation of a discrete-time periodic band-limited signal from its known values in an interval having the same length as the bandwidth of the signal.

We consider the discrete-discrete band-limited extrapolation problem: Given $2M+1$ consecutive values of a discrete-time periodic sequence $x(n)$ with period N whose discrete Fourier transform (DFT) $X(k)$ is known to be zero for $M < |k| \leq N/2$, determine the other values of $x(n)$. Note that simultaneous time-limitation and band-limitation is impossible for the continuous-time problem; it is entirely possible in the discrete-discrete problem, since periodic extensions are being made in both time and frequency.

3.2. Basic Concept

$x(n)$ is band-limited if N-point DFT $X(k)$ satisfies

$$X(k)S(k) = X(k); \quad S(k) = \begin{cases} 1, & \text{if } |k| \leq M; \\ \neq 1, & \text{if } |k| > M. \end{cases} \quad (5)$$

We make the following choice for $S(k)$: $z_k = e^{j2\pi k/N}$

$$S(k) = 1 + \prod_{i=-M}^{M} (z_k - z_i) = \sum_{n=0}^{2M} s(n) z_k^n \qquad (6)$$

The inverse DFT of (5) becomes

$$x(n) = \sum_{i=0}^{2M} s(i) x(n-i). \qquad (7)$$

This shows that the unknown $x(n)$ can be computed recursively from $2M+1$ consecutive known values of $x(n)$, without solving a system of equations, without even a division! The $s(n)$ can be computed ahead of time.

For a 2-D signal $x(n_1, n_2)$ band-limited in frequencies k_1 and k_2 *separately* (as well as jointly), we may apply the 1-D algorithm to extrapolate first in the n_1 direction, and then in the n_2 direction. Furthermore, the extrapolations in k_1 for each k_2 can all be performed in parallel. The decoupling between different k_2 also reduces the cumulative effect of roundoff error. Note that while the total number of values to be extrapolated is $N^2 - (2M+1)^2$, the extrapolations all use the 1-D extrapolation coefficients for extrapolating an $2M+1$-point signal to an N-point signal.

The well-known ill-posedness of the bandlimited extrapolation problem manifests itself in the large values of the coefficients $s(n)$ arising in (6) and (7). Care must be taken to ensure sufficient wordlengths to store these coefficients. Other than this, there are no problems with numerical roundoff error, unlike in the usual solution to this problem by solving a linear system.

4. TOEPLITZ-BLOCK-TOEPLITZ SYSTEMS

4.1. 2-D Deconvolution Formulation

The goal is now to solve the TBT system (1). Note that the TBT matrix is specified by the 2-D function $\{t(i,j), |i|, |j| \leq M-1\}$, where the index j specifies the diagonal within the i^{th} block. We follow the 2-D analogue of the procedure used in Section 2.

Rewrite (1) as the 2-D *cyclic* convolution having order $(2M-1) \times (2M-1)$

$$t(i,j) ** \tilde{x}(i,j) = \tilde{y}(i,j); \qquad (8a)$$

$$\tilde{x}(i,j) = \begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix}; \quad x = \begin{bmatrix} x_0^0 & \dots & x_{M-1}^0 \\ \vdots & \dots & \vdots \\ x_0^{M-1} & \dots & x_{M-1}^{M-1} \end{bmatrix} \quad (8b)$$

$$\tilde{y}(i,j) = \begin{bmatrix} y & z_{12} \\ z_{21} & z_{22} \end{bmatrix}; \quad y = \begin{bmatrix} y_0^0 & \dots & y_{M-1}^0 \\ \vdots & \dots & \vdots \\ y_0^{M-1} & \dots & y_{M-1}^{M-1} \end{bmatrix} \quad (8c)$$

where $\{z_{12}(i,j), z_{21}(i,j), z_{22}(i,j)\}$ denotes the *unknown* extension of y. We need to extrapolate these from y to convert the TBT system into a 2-D deconvolution problem.

As before, suppose that the 2-D function $\{t(i,j), |i|, |j| \leq M-1\}$ is 2-D bandlimited to DTFT frequency $\pi/2$ in both i and j. Again, this seems to be a reasonable assumption in practice. If the TBT matrix comes from sampling a point-spread or blurring function at double the Nyquist rate in each direction, then this assumption is satisfied.

4.2. TBT Procedure

1. Extrapolate y to $\{y, z_{12}, z_{21}, z_{22}\}$ using the extrapolation algorithm first in one direction, and then in the other direction;
2. Compute the 2-D DFTs of $\{y, z_{12}, z_{21}, z_{22}\}$ and $t(i,j)$, and divide at those 2-D frequencies where both are nonzero;
3. Extrapolate *the 2-D DFT of* $x(i,j)$ since $x(i,j)$ has finite *spatial* support (see (8)). Compute the inverse 2-D DFT of the result to obtain $x(i,j)$.

This procedure requires $6M^3 + 12M^2 \log_2(2M) + 3M^2$ real (MADS), since each 2-D extrapolation must compute $(2M)^2 - M^2 = 3M^2$ values at M MADs each. The 2-D FFTs can be *pruned* since only $1/4$ of the values are nonzero; savings from pruning are not included here. This is a considerable savings over the $O(M^5)$ MADs required by the algorithms of [1],[2], and it is comparable to just a few iterations of an iterative algorithm that uses the 2-D FFT to compute matrix-vector products.

Note that the actual implementations of the extrapolations are 1-D, and they are only from M to $2M$ for an $M^2 \times M^2$ TBT system. For many problems of practical interest, the coefficients in the extrapolation will not be large. The algorithm can almost be written out in closed form.

5. EXAMPLES

5.1. Simple Illustrative Example

We wish to solve the TBT system

$$\begin{bmatrix} 9 & 4 & 3 & 1 \\ 5 & 9 & 2 & 3 \\ 6 & 3 & 9 & 4 \\ 3 & 6 & 5 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 32 \\ 40 \\ 60 \\ 62 \end{bmatrix} \tag{9}$$

It is easily observed that (9) can be rewritten as the 2-D anticyclic convolution

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 9 & 5 \\ 3 & 6 & 3 \end{bmatrix} * \begin{bmatrix} x & y & 0 \\ z & w & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & 32 & 40 \\ * & 60 & 62 \end{bmatrix} \tag{10}$$

where the $*$ denotes the unknown values $\{z_{12}, z_{21}, z_{22}\}$ to be extrapolated. Note that the elements of each Toeplitz block form a row of $t(i, j)$.

Examination of $t(i, j)$ shows that the middle element in each row or column is the sum of the two values at the ends of that row or column. This is equivalent to the 2-D DTFT of $t(i, j)$ having zeros at $k_x = \pi$ and $k_y = \pi$, as can easily be confirmed. Hence we can use our algorithm to solve this.

Due to the small size of the problem, we must sample on the unit circle at $z = -e^{j2\pi k/3}$, in order to include the zero at $z = -1$ (frequency π). This is why we use an *anticyclic* (mod$(z^3 + 1)$) convolution in (10). With this change, the extrapolation coefficients are computed from (6) to be

$$\begin{aligned} S(k) = 1 + (e^{j2\pi/3} - e^{j2\pi k/3})(e^{-j2\pi/3} - e^{j2\pi k/3}) \rightarrow \\ \{S(0), S(1), S(2)\} = \{4, 1, 1\} \rightarrow \\ \{s(0), s(1), s(2)\} = \{2, -1, 1\} \end{aligned} \tag{11}$$

(remember the sampling on the unit circle). Using the coefficients in (11) to extrapolate the unknown $*$ values in (10) using the formula (7) results in

$$\begin{bmatrix} -6 & -28 & -22 \\ -8 & 32 & 40 \\ -2 & 60 & 62 \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \end{bmatrix} \tag{12}$$

which is indeed the 2-D anticyclic convolution in (10).

Note that (11) implements $x(n) * s(n) = x(n)$ (7), which in this example reduces to

$$x(n) - x(n-1) + x(n-2) = 0 \tag{13}$$

taking periodic extensions of period 3. This is clearly equivalent to a zero at frequency π. The extrapolation is easier to see using (13): $-28 = 32 - 60$; $-22 = 40 - 62$, etc.

5.2. More Realistic Example

The algorithm was applied to a 1024×1024 TBT matrix with 32×32 Toeplitz blocks. The 64×64 image associated with the TBT matrix is shown in Figure 1. It was generated by lowpass filtering the result of Matlab's rand function. The 1-D extrapolations required were all from 32 to 64 points, so the extrapolation coefficients are not too large. The solution was computed correctly.

6. REFERENCES

[1] N. Kalouptsidis et al., “On Block Matrices with Elements of Special Structure,” Proc. ICASSP, Paris, May 1982, 1744-1747.

[2] M. Wax and T. Kailath, “Efficient Inversion of Toeplitz-Block-Toeplitz Matrix,” IEEE Trans. ASSP 31(5), 1218-1221, Oct. 1983.

[3] H. Soltanian-Zadeh and A.E. Yagle, “A Fast Algorithm for Extrapolation of Discrete-Time Periodic Bandlimited Signals,” Signal Processing 33(2), 183-196, Aug. 1993.

[4] Ibid, Proc. ICASSP, Minn., April 1993, III-591-594.

[5] A.E. Yagle, ”An Explicit Closed-Form Solution to the Limited-Angle Discrete Tomography Problem,” Int. J. Imaging Systems and Technology 9(2/3), 174-180, 1998 (special issue on discrete tomography).

[6] Ibid, Proc. ICIP, Chicago, Oct. 1998, 1:44-48.