Extending Winograd's Small Convolution Algorithm to Longer Lengths *

Ivan W. Selesnick † and C. Sidney Burrus
Department of Electrical and Computer Engineering
Rice University, Houston, TX 77251-1892

December 14, 1994

Abstract

For short data sequences, Winograd's convolution algorithms attaining the minimum number of multiplications also attain a low number of additions, making them very efficient. However, for longer lengths they require a larger number of additions. Winograd's approach is usually extended to longer lengths by using a nesting approach such as the Agarwal-Cooley [1] or Split-Nesting [7] algorithms. Although these nesting algorithms are organizationally quite simple, they do not make the greatest use of the factorability of the data sequence length. The algorithm we propose adheres to Winograd's original approach more closely than do the nesting algorithms. By evaluating polynomials over simple matrices we retain, in algorithms for longer lengths, the basic structure and strategy of Winograd's approach, thereby designing computationally refined algorithms. This tactic is arithmetically profitable because Winograd's approach is based on a theory of minimum multiplicative complexity.

1 Introduction

In the 1970's Winograd developed a theory of multiplicative complexity for bilinear forms and designed algorithms for convolution attaining the minimum number of multiplications by irrational numbers [11, 12]. For short data sequences, these algorithms also attain a low number of additions, making them very efficient. However, for longer lengths they require a larger number of additions and become very cumbersome to design. Modifying Winograd's approach to achieve a more practical balance between multiplications and additions to obtain nonminimum multiply algorithms for longer lengths is a relevant problem [10] for which, to our knowledge, there exists no adequate theoretical framework. Winograd's approach is usually extended by using a nesting approach such as the Agarwal-Cooley [1] or Split-Nesting [7] algorithms. Although these nesting algorithms are organizationally quite simple, they do not make the greatest use of the factorability of the data sequence length. In this paper we give an arithmetically more efficient method for circular convolution by adhering to Winograd's original approach more closely than do the nesting algorithms.

The algorithm for convolution we propose allows us to reduce arithmetic complexity by converting polynomial products into products of matrix polynomials and by employing a matrix Toom-Cook technique. Recall that the efficiency of the Toom-Cook technique as it is used in Winograd's approach depends upon the availability of small simple integers (namely $0, \pm 1, \pm 2$) at which to evaluate polynomials [2]. However, the arithmetic simplicity of the scalar Toom-Cook technique breaks down for longer convolutions because it becomes necessary to evaluate higher order polynomials at additional points. We overcome this problem by evaluating polynomials over simple matrices. In this way we successfully retain, in algorithms for longer lengths, the basic structure and strategy of Winograd's approach, thereby designing computationally refined

*Appears in the 1994 ISCAS proceedings
†This work has been supported by DARPA under NDSEG fellowship N00014-89J-3204.

algorithms. This tactic is arithmetically profitable because Winograd's approach is based on a theory of minimum multiplicative complexity.

Moreover, the method presented here extends and combines the ideas of the Split-Nesting and the polynomial transform convolution algorithm of Nussbaumer [6]. That is, in case the length is the product of distinct primes (square free), one version of the proposed structure coincides with that of Split-Nesting, while if the length is a power of two, it is the polynomial transform based algorithm. In addition, although this method of extending Winograd's algorithms was alluded to in [9], it was left undeveloped.

2 Preliminaries

The companion matrix of a monic polynomial, $M(s) = m_0 + m_1 s + \cdots + m_{n-1}s^{n-1} + s^n$, is given by

$$C_M = \begin{bmatrix} & & & -m_0 \\ 1 & & & -m_1 \\ & \ddots & & \vdots \\ & & 1 & -m_{n-1} \end{bmatrix}. \tag{1}$$

It permits the following matrix formulation of convolution. If $X(s), H(s), Y(s)$ are polynomials of order $n-1$ and $M(s)$ is a monic polynomial of order n, then

$$Y(s) = \langle H(s)X(s)\rangle_{M(s)} \quad \Longleftrightarrow \quad y = \left(\sum_{k=0}^{n-1} h_k C_M^k\right) x \tag{2}$$

where the vectors x, h, y are the coefficients of $X(s)$, $H(s)$, $Y(s)$ and C_M is the companion matrix of $M(s)$. In (2), y is the convolution of x and h with respect to $M(s)$. In the case of circular convolution, $M(s) = s^n - 1$ and C_{s^n-1} is the circular shift matrix denoted by S_n,

$$S_n = \begin{bmatrix} & & & 1 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{bmatrix}.$$

Below we rely on the notion of companion matrices of matrix polynomials, in which case the m_i in (1) are taken to be matrices. Then (2) can be understood to represent the convolution of a matrix sequence, H_k, and a vector sequence, X_k, with respect to a matrix polynomial. This is a central notion in our extension of Winograd's algorithm.

Similarity transformations can be used to interpret the action of some convolution algorithms. If $C_M = T^{-1}AT$ for some matrix T (C_M and A are similar, denoted $C_M \sim A$), then (2) becomes

$$y = T^{-1}\left(\sum_{k=0}^{n-1} h_k A^k\right) Tx. \tag{3}$$

That is, by employing the similarity transformation given by T in this way, the action of C_M^k is replaced by that of A^k. Many convolution algorithms can be understood by understanding the manipulations made to C_M and the resulting matrix A. If the transformation T is to be useful, it must satisfy two requirements: (a) Tx must be simple to compute, and (b) A must have some advantageous structure.

Winograd's algorithm can be described in this manner. Suppose $M(s)$ can be factored as $M(s) = M_1(s)M_2(s)$ where M_1 and M_2 have no common roots, then $C_M \sim (C_{M_1} \oplus C_{M_2})$. Using this similarity and recalling (2), the original convolution is decomposed into disjoint convolutions. This is, in fact, a statement of the Chinese Remainder Theorem for polynomials expressed in matrix notation. In the case of circular convolution, $s^n - 1 = \prod_{d|n} \Phi_d(s)$, so that S_n is transformed to a block diagonal matrix,

$$S_n \sim \begin{bmatrix} C_{\Phi_1} & & \\ & \ddots & \\ & & C_{\Phi_n} \end{bmatrix} = \bigoplus_{d|n} C_{\Phi_d} \tag{4}$$

where Φ_d is the d^{th} cyclotomic polynomial [5]. In this case, each block represents a convolution with respect to a cyclotomic polynomial. Winograd's approach carries out these 'cyclotomic' convolutions using the Toom-Cook algorithm. Note that, because the order of Φ_d is $\phi(d)$, where $\phi(\cdot)$ is the Euler totient function, each convolution is of length $\phi(d)$.

The Agarwal-Cooley algorithm utilizes the fact that $S_n = P^t(S_{n_1} \otimes S_{n_2})P$ where $n = n_1 n_2$, $(n_1, n_2) = 1$, $\otimes$ is the Kronecker product, and P is an appropriate permutation. This converts a one dimensional circular convolution to a two dimensional one. Then an n_1-point and an n_2-point circular convolution algorithm can be combined to obtain an n-point algorithm.

The Split-Nesting algorithm combines the structures of the Winograd and Agarwal-Cooley methods, so that S_n is transformed to a block diagonal matrix as in (4),

$$S_n \sim \bigoplus_{d|n} \Psi(d). \tag{5}$$

Here $\Psi(d) = \bigotimes_{p|d, p\in\mathcal{P}} C_{\Phi_{H_d(p)}}$ where $H_d(p)$ is the highest power of p dividing d, and $\mathcal{P}$ is the set of primes. For example, $\Psi(225) = C_{\Phi_9} \otimes C_{\Phi_{25}}$. In this structure, a multidimensional cyclotomic convolution, represented by $\Psi(d)$, replaces each cyclotomic convolution in Winograd's algorithm (represented by C_{Φ_d} in (4)). Indeed, if the product of $b_1, \ldots, b_k$ is d and they are pairwise relatively prime, then $C_{\Phi_d} \sim C_{\Phi_{b_1}} \otimes \cdots \otimes C_{\Phi_{b_k}}$. This gives a method for combining cyclotomic convolutions to compute a longer circular convolution. It is like the Agarwal-Cooley method but requires fewer additions [7].

The Matrix Toom-Cook technique gives a way to perform block convolution. As an illustrative example, consider the following linear convolution. For the time being, we assume no special structure of H_i.

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} H_0 & \\ H_1 & H_0 \\ & H_1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}$$

with $H_i \in \Re^{n\times n}$, $Y_i, X_i \in \Re^n$. Define $Y(s) = Y_0 + sY_1 + s^2 Y_2$, $H(s) = H_0 + sH_1$, and $X(s) = X_0 + sX_1$. Let S_1, S_2, S_3 be three matrices used for evaluation and let $Y(S_i) = Y_0 + S_i Y_1 + S_i^2 Y_2$. That is, take the *left* value of $Y(S_i)$ (only then do the dimensions agree), so that a block Vandermonde matrix gives the relationship

$$\begin{bmatrix} Y(S_1) \\ Y(S_2) \\ Y(S_3) \end{bmatrix} = \begin{bmatrix} I & S_1 & S_1^2 \\ I & S_2 & S_2^2 \\ I & S_3 & S_3^2 \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix}.$$

Then $S_i H_k = H_k S_i \Rightarrow Y(S_i) = H(S_i)X(S_i)$. To see this, note the following sequence of equalities.

$$\begin{aligned} H(S_i)X(S_i) &= (H_0 + S_i H_1)(X_0 + S_i X_1) \\ &= H_0X_0 + S_iH_1X_0 + H_0S_iX_1 + S_iH_1S_iX_1 \\ &= H_0X_0 + S_i(H_1X_0 + H_0X_1) + S_i^2 H_1X_1 \\ &= Y_0 + S_iY_1 + S_i^2 Y_2 \\ &= Y(S_i) \end{aligned}$$

That S_iH_k equals H_kS_i gives the third equality from the second. That is, S_i must be chosen so as to *commute* with each of H_k. In this case

$$\begin{bmatrix} Y(S_1) \\ Y(S_2) \\ Y(S_3) \end{bmatrix} = \begin{bmatrix} H(S_1) \\ H(S_2) \\ H(S_3) \end{bmatrix} * \begin{bmatrix} X(S_1) \\ X(S_2) \\ X(S_3) \end{bmatrix}$$

where $*$ denotes matrix-vector multiplication on the n point vectors. Since

$$\begin{bmatrix} H(S_1) \\ H(S_2) \\ H(S_3) \end{bmatrix} = \begin{bmatrix} I & S_1 \\ I & S_2 \\ I & S_3 \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \text{and} \begin{bmatrix} X(S_1) \\ X(S_2) \\ X(S_3) \end{bmatrix} = \begin{bmatrix} I & S_1 \\ I & S_2 \\ I & S_3 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}$$

one gets the matrix Toom-Cook expression,

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} I & S_1 & S_1^2 \\ I & S_2 & S_2^2 \\ I & S_3 & S_3^2 \end{bmatrix}^{-1} \left\{ \begin{bmatrix} I & S_1 \\ I & S_2 \\ I & S_3 \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} * \begin{bmatrix} I & S_1 \\ I & S_2 \\ I & S_3 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \right\}. \tag{6}$$

In the proposed method, the matrices H_k are of the form $\sum_k a_k \left[C_{\Phi_d} \right]^k$ so that simple evaluation matrices can be taken to be powers of C_{Φ_d}. Then the matrix-vector products denoted by $*$ represent another level of convolutions.

3 A Winograd Based Approach

The method we propose for circular convolution retains the general structure of Winograd's original approach by converting cyclotomic convolution to *matrix* convolution. The key observation that makes this possible is the following. It uses the notational framework given above to encapsulate precisely the possible matrix convolutions to which a cyclotomic convolution can be converted.

> *If r divides d, then C_{Φ_d}, the companion matrix of the d^{th} cyclotomic polynomial, is similar to the companion matrix of the matrix polynomial, $C_{\Phi_r}^{\phi(q)} \Phi_q(s^{d/rq} C_{\Phi_r}^{-1})$, where $q = \eta(d)/\eta(r)$ and $\eta(d)$ denotes the product of the prime divisors of d.*

For example, taking $d = 45$ and $r = 5$, one gets that the companion matrix of $s^{24} - s^{21} + s^{15} - s^{12} + s^9 - s^3 + 1$ is similar to the companion matrix of $s^6 I_4 + s^3 C_{\Phi_5} + C_{\Phi_5}^2$. Denote the matrix polynomial $C_{\Phi_r}^{\phi(q)} \Phi_q(s^{d/rq} C_{\Phi_r}^{-1})$ by $Q_{d,r}(s)$. Note that the order of $Q_{d,r}(s)$ is $\phi(d)/\phi(r)$ and that the $C_{\Phi_r}^{\phi(q)}$ term is present so that the matrix polynomial is monic. Also note that the companion matrices of both $Q_{d,1}$ and $Q_{d,d}$ are simply C_{Φ_d} again.

Because C_{Φ_d} and $C_{Q_{d,r}}$ are similar, the action of C_{Φ_d} in Winograd's algorithm can be replaced by the action of $C_{Q_{d,r}}$. Then (4) becomes

$$S_n \sim \bigoplus_{d|n} C_{Q_{d,r(d)}}.$$

By (2), the form $C_{Q_{d,r}}$ gives rise to a $\phi(d)/\phi(r)$-point matrix convolution with respect to $Q_{d,r}(s)$. Moreover, the coefficients of $Q_{d,r}(s)$ are scalar multiples of powers of C_{Φ_r}, so that the blocks, H_k, of the associated matrix convolution are of the form $\sum_k a_k \left[C_{\Phi_r} \right]^k$. Since the matrix Toom-Cook technique calls for matrices, S_i, commuting with H_k, they may be chosen to be powers of C_{Φ_r}. Because $C_{\Phi_r}^r$ is $I_{\phi(r)}$, the evaluation of a polynomial at these points may be performed with an FFT type algorithm.

Since the matrices, H_k, in the matrix Toom-Cook method are of the form $\sum_k a_k \left[C_{\Phi_r} \right]^k$, by (2), the operation represented by $*$ in (6) denotes convolution with respect to $\Phi_r(s)$. To compute this cyclotomic convolution, the process just described can be repeated, etc, until d is a prime. When d is a prime, the usual scalar Toom-Cook method of Winograd's original algorithm (or some other method) must be used.

To discuss the selection of the value r in this process when d is fixed, note that the length of the block convolution associated with $Q_{d,r}$ is $\phi(d)/\phi(r)$. The matrix Toom-Cook technique then calls for $2\phi(d)/\phi(r)-1$ simple matrices, S_i, commuting with matrices of the form $\sum_k a_k \left[C_{\Phi_r}\right]^k$. If the length, $\phi(d)/\phi(r)$ is too high, then there will not be enough such simple matrices. At the same time, if r is chosen too high, then the advantage of this approach is lost and the arithmetic cost rises.

This suggests that it is sensible to take for r the smallest divisor of d for which there are enough matrices, S_i. Here, we will restrict S_i to be 0, ∞, and integer powers of $\pm C_{\Phi_r}$, as polynomial evaluation at these points requires no multiplications. (In this context, the value of a polynomial at ∞ is its leading coefficient.) Note that the powers of C_{Φ_r} supply r matrices, ($C_{\Phi_r}^r = I_{\phi(r)}$). The powers of $-C_{\Phi_r}$ supply an additional r matrices when r is odd, however, when r is even $C_{\Phi_r}^{r/2} = -I_{\phi(r)}$ so that the powers of $-C_{\Phi_r}$ are the same as the powers of C_{Φ_r}. Consequently, one has $r+2$ matrices, S_i, for r even and $2r+1$ for r odd. Then it is sensible to choose for r the smallest divisor of d such that

$$a)\quad 2\phi(d)/\phi(r)-1 \;\le\; r+2 \quad \text{if } r \text{ is even} \tag{7}$$
$$b)\quad 2\phi(d)/\phi(r)-1 \;\le\; 2r+2 \quad \text{if } r \text{ is odd.} \tag{8}$$

For some values of d, table 1 gives r when it is chosen according to this scheme.

Table 1: Selection of r.

d	r	d	r	d	r	d	r	d	r
4	1	14	7	24	3	35	5	49	7
6	1	15	3	25	5	36	9	50	5
8	4	16	8	27	9	40	5	54	9
9	3	18	3	28	7	42	7	56	7
10	5	20	5	30	3	45	5	60	5
12	3	21	7	32	8	48	8	63	7

Next, note that because we already have the similarity $C_{\Phi_d} \sim \Psi(d)$ (recall the Split-Nesting algorithm), it is sufficient here to consider only the transformation accomplishing the similarity $\Psi(d) \sim C_{Q_{d,r}}$. Although we do not give the details here, by combining appropriate transformations one can get the desired transformation for $\Psi(d) \sim C_{Q_{d,r}}$.

This approach appears to us to be the most natural way to extend Winograd's algorithm efficiently. Accordingly, for a variety of lengths, we expect it to require fewer arithmetic operations than previously designed algorithms.

4 A Variation on the Approach

In this section we focus on a variation of the above described algorithm in which the matrix cyclotomic convolutions are especially simple. We have investigated this method more extensively than the more general algorithm given in the previous section.

Here r is chosen so that it not only divides d and satisfies (7,8) but is also a multiple of $\eta(d)$, the product of the prime divisors of d. Because $q = \eta(d)/\eta(r)$ is always 1 when r is chosen in this way, and because $\Phi_1(s) = s-1$, one gets $Q_{d,r}(s) = s^{\phi(d)/\phi(r)} I_{\phi(r)} - C_{\Phi_r}$. The companion matrix of $Q_{d,r}(s)$ can therefore be written as $I_{\phi(d)-\phi(r)} \oslash C_{\Phi_r}$ where $A \oslash B$ denotes

$$\begin{bmatrix} & B \\ A & \end{bmatrix}.$$

Using this observation and recalling that $C_{\Phi_r} \sim \Psi(r)$, this gives $C_{\Phi_d} \sim I_{\phi(d)-\phi(r)} \oslash \Psi(r)$. In particular, taking $r = \eta(d)$, we find that C_{Φ_d} is similar to $I_{\phi(d)-\theta(d)} \oslash \Omega(d)$, the companion matrix of $s^{\phi(d)/\theta(d)} I_{\theta(d)} - \Omega(d)$ where $\theta(d)$ denotes $\phi(\eta(d))$ and $\Omega(d)$ denotes $\Psi(\eta(d))$. Then (4) becomes

$$S_n \sim \bigoplus_{d|n} \begin{bmatrix} & \Omega(d) \\ I_{\phi(d)-\theta(d)} & \end{bmatrix}. \tag{9}$$

After this form is acquired from the circular shift matrix S_n by the appropriate similarity transformation, the arithmetic cost of which is discussed below, the only further similarity transformations required are simply permutations. This is because a permutation gives the similarity $I_{\phi(d)-\theta(d)} \oslash \Omega(d) \sim I_{\phi(d)-\phi(r)} \oslash I_{\phi(r)-\theta(r)} \oslash \Omega(r)$. Using this permutation, one can convert convolution with respect to $s^{\phi(d)/\theta(d)} I_{\theta(d)} - \Omega(d)$ to a convolution with respect to $s^{\phi(d)/\phi(r)} I_{\phi(r)} - \left(I_{\phi(r)-\theta(r)} \oslash \Omega(r)\right)$. The second of which can be computed with the matrix Toom-Cook algorithm in which the blocks, H_k, are of the form $\sum_k a_k \left(I_{\phi(r)-\theta(r)} \oslash \Omega(r)\right)^k$. Then, by (2), the operation represented by $*$ in (6) denotes matrix convolution with respect to $s^{\phi(r)/\theta(r)} I_{\theta(r)} - \Omega(r)$.

Since this is exactly the operation with which we began, with r in place of d, we can repeat this procedure on this shorter convolution, etc. When d is finally square free, $d = \eta(d)$, we can go no further and the algorithm of the previous section or a nesting algorithm can be used. This gives a way of mixing the algorithm of the last section with others.

The arithmetic cost associated with carrying out similarity (9) is important and we have found an arithmetically reasonable method for this. Our method for effecting (9) incorporates carrying out similarity (5). However, we find it necessary to include other operations the details of which we do not cover here. We have no simple formula giving the number of extra additions taken, so we have listed in table 2 the number of additions associated with (5) and the extra additions we take for (9). To be precise, these are the number of additions associated with T where $S_n = T^{-1}(\oplus X_i)T$. Because we anticipate the use of the matrix exchange property [4], we ignore the arithmetic cost of T^{-1}. It is replaced by T^t which is arithmetically equivalent to T for similarities (5) and (9). It turns out that the number of extra additions we take for similarity (9) is nonzero only if n is either divisible by two distinct odd squares or divisible by 4 and a number of the form $p_1^2 p_2$ where p_1 and p_2 are distinct odd primes. The table entries are for the first such numbers n. For these n, notice that the number of extra additions needed for (9) is very small. We note here that for the main similarity of the Split-Nesting algorithm (5), T can be carried out in $2n\left(k - \sum_{i=1}^{k} 1/p_i^{e_i}\right)$ additions where $n = p_1^{e_1} \cdots p_k^{e_k}$ is a prime factorization of n. No multiplications are required. Using these observations, one gets the following algorithm.

Table 2: The number of additions for effecting similarities (5) and (9) are given by a and $a+b$ respectively.

n	a	b	n	a	b	n	a	b
180	878	12	441	1648	12	600	3002	18
225	832	9	450	2114	18	612	3158	60
252	1258	20	468	2398	44	675	2596	36
300	1426	6	504	2642	60	684	3538	68
360	1846	36	540	2714	48	700	3594	40
396	2018	36	588	2818	8	720	3782	84

A Circular Convolution Algorithm If A_p, C_p, and G_p are matrices such that $G_p C_p \{A_p u * A_p v\}$ gives the p^{th} cyclotomic convolution of u and v for all primes p dividing n, then the n-point circular convolution of h and x is given by

$$h \circ x = T^{-1} GC \{ATh * ATx\} \tag{10}$$

where

1. T is a matrix such that

$$S_n = T^{-1} \left(\bigoplus_{d|n} \begin{bmatrix} & \Omega(d) \\ I_{\phi(d)-\theta(d)} & \end{bmatrix} \right) T$$

2. A, C, and G are given by the direct sums

$$A = \bigoplus_{d|n} A_d, \quad C = \bigoplus_{d|n} C_d, \quad G = \bigoplus_{d|n} G_d,$$

with, for d square free,

$$A_d = \bigotimes_{p|d, p\in\mathcal{P}} A_p, \quad C_d = \bigotimes_{p|d, p\in\mathcal{P}} C_p, \quad G_d = \bigotimes_{p|d, p\in\mathcal{P}} G_p$$

and for d divisible by a square,

$$\begin{aligned} A_d &= \left(I_{2d/r(d)-1} \otimes A_{r(d)}\right) V_d P_d \\ C_d &= P_d^t U_d \left(I_{2d/r(d)-1} \otimes G_{r(d)} C_{r(d)}\right) \\ G_d &= \left[I_{\phi(d)} \mid E_{d/r(d)} \otimes \left(I_{\phi(r(d))-\theta(d)} \oslash \Omega(d)\right)\right] \\ P_d &= S(d/\eta(d), d/r(d)) \otimes I_{\theta(d)} \end{aligned}$$

where E_k denotes the first $k-1$ columns of I_k.

3. $r(d)$ satisfies $\eta(d)|r(d)$ and $r(d)|d$.

4. V_d is the degree $d/r(d)-1$ Vandermonde matrix of $2d/r(d)-1$ points commuting with $I_{\phi(r(d))-\theta(d)} \oslash \Omega(d)$.

5. U_d is the inverse of the degree $2d/r(d)-2$ Vandermonde matrix of same the points used in prescribing V_d.

Here $S(\cdot, \cdot)$ is the stride permutation.

5 Applications of the Algorithm

The central application and motivation we have for this approach is the use of convolution algorithms in prime length FFTs and Prime Factor FFT modules. Short prime length FFTs are generally computed using Rader's permutation [8] and Winograd's short convolution algorithm. However, for primes > 23, the length of the cyclotomic convolutions to which this gives rise are too long for the scalar Toom-Cook technique to be efficient. To keep the arithmetic complexity low for these lengths, it is necessary to modify the approach for short lengths as we have done here.

The convolution algorithm proposed above can also be employed in PFA modules. Moreover, since the approach above gives a bilinear form, it can be used in the Winograd Fourier transform algorithm or incorporated into the dynamic programming technique for designing optimal composite length FFT algorithms [3].

6 Summary

We have given a similarity that makes explicit the equivalence of scalar and matrix cyclotomic convolutions. By employing this similarity we replace long cyclotomic convolutions with shorter matrix convolutions, thereby reducing the overall arithmetic complexity for the circular convolution of sequences of certain lengths.

In this way, the only scalar cyclotomic convolutions required are those of prime index. We also describe an algorithm that allows one to combine this approach with nesting and other algorithms to maximize efficiency and flexibility. The approach given here is applicable to arbitrary composite lengths by appropriate similarity transformations.

A Notation

We summarize some of the notation used.

$\phi(d)$: The Euler totient function.
$\eta(d)$: The product of the prime divisors of d.
$\theta(d)$: $\phi(\eta(d))$.
$H_d(p)$: The highest power of p dividing d.
$\mathcal{P}$: The set of primes.
C_M : The companion matrix of $M(s)$.
$\Phi_d(s)$: The d^{th} cyclotomic polynomial.
S_n : The circular shift matrix of size n.
$Q_{d,r}$: $C_{\Phi_r}^{\phi(q)}\Phi_q(s^{d/rq}C_{\Phi_r}^{-1})$.
$\Psi(d)$: $\bigotimes_{p|d,p\in\mathcal{P}} C_{\Phi_{H_d(p)}}$.
$\Omega(d)$: $\Psi(\eta(d))$.

References

[1] R. C. Agarwal and J. W. Cooley. New algorithms for digital convolution. *IEEE Trans. Acoust., Speech, Signal Proc.*, 25(5):392–410, October 1977.

[2] R. E. Blahut. *Fast Algorithms for Digital Signal Processing.* Addison-Wesley, 1985.

[3] H. W. Johnson and C. S. Burrus. The design of optimal DFT algorithms using dynamic programming. *IEEE Trans. Acoust., Speech, Signal Proc.*, 31(2):378–387, April 1983.

[4] H. W. Johnson and C. S. Burrus. On the structure of efficient DFT algorithms. *IEEE Trans. Acoust., Speech, Signal Proc.*, 33(1):248–254, February 1985.

[5] J. H. McClellan and C. M. Rader. *Number Theory in Digital Signal Processing.* Prentice Hall, 1979.

[6] H. J. Nussbaumer. Fast polynomial transform algorithms for digital convolution. *IEEE Trans. Acoust., Speech, Signal Proc.*, 28(2):205–215, April 1980.

[7] H. J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms.* Sringer-Verlag, 1982.

[8] C. M. Rader. Discrete Fourier transform when the number of data samples is prime. *Proc. IEEE*, 56(6):1107–1108, June 1968.

[9] I. W. Selesnick and C. S. Burrus. Multidimensional mapping techniques for convolution. In *Proc. of 1993 ICASSP*, 1993.

[10] R. Stasinski. Easy generation of small-n discrete Fourier transform algorithms. *IEE Proceedings, part G*, 133(3):133–139, June 1986.

[11] S. Winograd. Some bilinear forms whose multiplicative complexity depends on the field of constants. *Mathematical Systems Theory*, 10:169–180, 1977.

[12] S. Winograd. *Arithmetic Complexity of Computations.* SIAM, 1980.