Cyclic convolution of long sequences using number theoretic transform

Wan-Chi Siu, AP(HK), M.Phil., C.Eng., M.I.E.R.E., Mem.I.E.E.E., and

Indexing terms: Signal processing, Number theoretic transforms

Abstract: In this paper, number theoretic transforms (NTT) are examined and expressed in a way that facilitates the computation of the convolution sum of two one-dimensional sequences each of length \( N \) in two-dimensional (or multidimensional) form. The number of multiplications per point is shown to be always less than two when \( N_1 \) and \( N_2 \) are not mutually prime, and this figure is reduced to one multiplication per point when \( N_1 \) and \( N_2 \) are mutually prime and a nested NTT is used. For the convolution of two-dimensional arrays, the number of multiplications per point turns out to be always less than four when \( N_1 \) and \( N_2 \) are not mutually prime, and the number of multiplications remains as one per point when \( N_1 \) and \( N_2 \) are mutually prime.

1 Introduction

One- and two-dimensional cyclic convolutions have found many applications [1] in such areas as digital speech processing [2], image processing [3], radar processing [1] etc. With the development of the fast-Fourier transform (FFT) algorithm, the discrete Fourier transform (DFT) has often been applied to compute the circular convolution when constructing digital signal processors. However, if the DFT is used to effect the implementation, round-off error cannot be avoided, and the computation is rather time consuming (even if the FFT is used), since both DFT and inverse DFT involve complicated multiplications of irrational complex numbers.

Because of its suitability to fast and error-free calculation of cyclic convolution, number theoretic transform (NTT) [4-6] has received growing attention during recent years. This transform may be defined on a ring or field of integers with arithmetic carried out modulo an integer \( M \). With a proper choice of root of unity \( \alpha \), the calculations of the transform and the inverse transform of the NTT can involve only shifts, and no multiplications at all. Of the various versions of the NTT proposed, the Fermat number transform (FNT), which has been thoroughly investigated by Agarwal and Burrus [6], and the Mersenne number transform (MNT), which was introduced by Rader [5], with the efficient hardware structures suggested by Siu and Constantinides [7], are the most promising cases. Recently Siu and Constantinides [8] have also shown that the number of multiplications per point is only one for the computation of the DFT of a real sequence using NTTs, and they have also shown that the FNT [9] can be used very effectively to compute short and long DFTs.

However, for actual implementation, the FNT and MNT are limited by the wordlength and sequence-length constraints [10]. The normal transform length of MNT is \( P \) for modulo base of \((2^P - 1)\), and the MNT transform length can be increased to \( 2P \) when \( \alpha = -2 \). Agarwal and Burrus [6] have pointed out that a more practical choice of Fermat number, the modulo base, is \( F_s \), in which case \( M = F_s = 2^s + 1 \). The corresponding transform length is 128 and \( \alpha = \sqrt{2} \). This length may be too short for certain applications, such as radar or picture processing. Another constraint may be due to the memory size of the computer, especially if a minicomputer or a microcomputer is used to compute the NTT as an aid to convolution. This problem is more serious in two-dimensional filtering. If the size of the main core memory is not large enough to store the whole matrix, one may store part or all data on disc. An obvious method for two-dimensional transformation is to transform the rows of the data matrix, transpose the resulting matrix and then transform this again. However, this method is not too efficient, since matrix transposition of data that involves the disc is very difficult and time consuming. One possible solution, as pointed out by Rader [10], would involve sectioning the data and then for filtering one could use an overlap save or an overlap add technique.

It is the purpose of this paper to give systematic ways of implementing convolutions of long sequences using number theoretic transforms. These methods are useful in computing one-dimensional cyclic convolutions, but they are also suitable, and very efficient, for computing two-dimensional convolutions.

2 Circular shift theorem

Consider the following time-domain sequence:

\[
x(n) = \{x_0, x_1, \ldots, x_{N-1}\}
\]

where \( N \) is the length of the sequence.

Let \( M \) be the base for modulo arithmetic in a number theoretic transform (NTT). The residue of \( \alpha^n \)-modulo \( M \) can be written as \( \langle \alpha^n \rangle_M \), where \( \alpha \) is a root of unity of order \( N \). The NTT of the sequence \( x(n) \) is then defined as [4-6]

\[
T\{x(n)\} = X_k = \sum_{n=0}^{N-1} x_n \alpha^{nk} M
\]

for \( k = 0, 1, \ldots, N - 1 \) (1)

This definition assumes that the sequence is periodically extended with period \( N \) or the indices are evaluated modulo \( N \). Hence,

\[
T\{x(n - l)\} = \sum_{n=0}^{N-1} x_{n-l} \alpha^{nk} M
\]

In this equation, substitute \( r = n - l \), so that

\[
T\{x(n - l)\} = \sum_{r=0}^{N-1} x_r \alpha^{-r} M
\]

(2)

Eqn. 2 shows that a circular shift of the input sequence gives a circular shift of the transformed output and this is actually the shift theorem [6] of cyclic convolution.
Now let us define the z-transform* of the same sequence \[ Z\{x(n)\} = X(z) = \sum_{n=0}^{N-1} x_n z^n \] (3)
Hence,\[ Z\{x(n - l)\} = z^{-l} X(z) \]
(4)
It can be seen from eqns. 2 and 4 that the shift theorem on NTT and the z-transform are related by the following expression:
\[ z^l X(z) \leftrightarrow z^l X_k \]
(5)

3 One-dimensional circular convolution

Consider the length of a sequence to be \( N \), a composite number, and let \( N = N_1 N_2 \). The one-dimensional z-transform of this sequence can map into a two-dimensional form by making use of the following equations:
\[ n = N_2 n_1 + n_2 \] (6)
where \( n_1 = 0, 1, \ldots, N_1 - 1 \)
\[ n_2 = 0, 1, \ldots, N_2 - 1 \]
and
\[ z_1 = z^{N_2} \]
(7)
Under these conditions, eqn. 3 becomes
\[ X(z, z_1) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_2,n_1} z_1^{n_2} z^{n_2} \] (8)
The cyclic convolution of \( x(n) \) and \( h(m) \) can be written as
\[ y_l = \sum_{n=0}^{N-1} x_n h_{l-n} \] (10)
for \( l = 0, 1, \ldots, N - 1 \)
This can be expressed as a two-dimensional z-transform, using eqns. 8 and 9,
\[ X(z, z_1) H(z, z_1) = \left( \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x_{n_2,n_1} z_1^{n_2} z^{n_2} \right) \times \left( \sum_{m_2=0}^{N_2-1} \sum_{m_1=0}^{N_1-1} h_{m_2,m_1} z_1^{m_2} z^{m_2} \right) \mod (z_1^{N_1} - 1) \]
(11)

Since \( n_1 \) and \( m_1 \) are cyclic, the product inside the square bracket actually corresponds to a cyclic convolution. Let us define
\[ X_n z_1 \] \[ H_m z_1 \]
\[ X_n z_1 H_m z_1 = \left( \sum_{n_1=0}^{N_1-1} x_{n_2,n_1} z_1^{n_1} \right) \times \left( \sum_{m_1=0}^{N_1-1} h_{m_2,m_1} z_1^{m_1} \right) \mod (z_1^{N_1} - 1) \]
(12)

Hence one can write
\[ X(z, z_1) H(z, z_1) = \sum_{n_2=0}^{N_2-1} \sum_{m_2=0}^{N_2-1} X_{n_2} z_1 H_{m_2} z_1 \] (13)
Evaluating eqn. 12 at \( z_1 = z^{k_1} \) (hence this gives the number theoretic transforms of the sequences \( \{x_{n_2,n_1}, n_2 = 0, 1, \ldots, N_2 \} \) and \( \{h_{m_2,m_1}, m_2 = 0, 1, \ldots, N_2 \} \) with respect to \( n_1 \) and \( m_1 \) and substituting into eqn. 13, we obtain
\[ X_{n_2,k_1} H_{m_2,k_1} = \left( \sum_{n_1=0}^{N_1-1} x_{n_2,n_1} z^{k_1} \right) \times \left( \sum_{m_1=0}^{N_1-1} h_{m_2,m_1} z^{k_1} \right) \mod (z_1^{N_1} - 1) \]
for \( k_1 = 0, 1, \ldots, N_1 - 1 \)
(14)
\[ X_{n_2,k_1} H_{m_2,k_1} = \left( \sum_{n_1=0}^{N_1-1} x_{n_2,n_1} z^{k_1} \right) \mod (z_1^{N_1} - 1) \]
(15)
\[ H_{m_2,k_1} = \left( \sum_{m_1=0}^{N_1-1} h_{m_2,m_1} z^{k_1} \right) \mod (z_1^{N_1} - 1) \]
(16)
At this juncture eqn. 14 needs to be examined carefully. It should be noted that this expression is not cyclic in \( n_2 \) and \( m_2 \), and the contribution given by \( z z_1^{m_2} \) has not been taken into account. Usually \( z_1^{m_2} \) can be expressed as \( z_1^{m_2} = z_1^{m_2 N_2} \mod (z_1^{N_1} - 1) \) and \( \beta_{n_2,m_2} \) is defined as the integral part of \( (n_2 + m_2)/N_2 \). The cyclic convolution of \( X_{n_2,k_1} \) and \( H_{m_2,k_1} \) or indeed that the transformed product \( X_{n_2,k_1} H_{m_2,k_1} \) should be multiplied by \( \beta_{n_2,m_2} z_1^{m_2} \). However, this modification could be done at a later stage or after the expansion of the polynomial eqn. 14 has been computed.

Eqn. 14 is actually a linear convolution sum of the sequences \( \{X_{n_2,k_1}, n_2 = 0, 1, \ldots, N_2 - 1\} \) and \( \{H_{m_2,k_1}, m_1 = 0, 1, \ldots, N_2 - 1\} \) or
\[ X_{n_2,k_1} H_{m_2,k_1} = \left( \sum_{n_2=0}^{N_2-1} X_{n_2,k_1} z^{k_2} \right) \left( \sum_{m_2=0}^{N_2-1} H_{m_2,k_1} z^{m_2} \right) \mod (z_1^{N_1} - 1) \]
\[ \triangleq W_{0,k_1} + W_{1,k_1} z^{k_2} + \cdots + W_{N_2-2,k_1} z^{2N_2-2} \] (17)
where \( k_1 = 0, 1, \ldots, (N_1 - 1) \)
A direct implementation of eqn. 14 requires \( N_2^2 \) multiplications for \( N_2 \) points. However, we can implement this with a much lower number of multiplications using eqn. 17. Two ways for doing so are now shown.

(i) Lagrange interpolating formula

Eqn. 17 is a product of two polynomials and indeed the actual answers to the polynomial multiplication problem are the coefficients \( W_{0,k_1}, W_{1,k_1}, \ldots, W_{N_2-2,k_1} \) of the resulting product polynomial. Hence these \( (2N_2 - 1) \) coefficients of the \( (2N_2 - 2) \)th order polynomial resulting from

the product can be interpolated exactly through \((2N_2 - 1)\) points using the Lagrange interpolation formula (or, more precisely, the Cook-Toom algorithm [11, 13]). By Lagrange interpolation formula, it is meant that

\[
Y_k(z) = W_0, k_i + W_1, k_i z^2 + \cdots + W_{2N_2 - 3}, k_i z^{2N_2 - 2}
\]

\[
= \sum_{i=0}^{2N_2-2} m_{k_i,i} L(z)
\]

where

(a) the interpolating polynomials are given by

\[
L(z) = \prod_{j \neq i} \frac{z - z_j}{z_i - z_j}
\]

and

(b) The interpolating constants are given by

\[
m_{k_i,i} = X_{k_i}(z)H_{k_i}(z)
\]

In this expression \(\{z_0, z_1, \ldots, z_{2N_2 - 2}\}\) are distinct numbers for interpolation. As shown in eqn. 20, this method requires only \((2N_2 - 1)\) multiplications for the calculation of \(N_2\) points, or \((2 - 1/N_2)\) per point. This method is extremely useful for small values of \(N_2\), but for relatively large \(N_2\) the interpolating polynomials become more and more complicated, thus making it difficult thereby to apply the Lagrange interpolation formula to find the result.

(ii) A second number theoretic transform

As already pointed out, eqn. 17 is in fact a linear convolution sum of two sequences each of length \(N_2\). The result of this linear convolution can be found by a length-(\(2N_2 - 1\)) cyclic convolution [1, 14] of two \((2N_2 - 1)\) sequences formed by appending \((N_2 - 1)\) zeros at the end of the original sequences. For example, the linear convolution of \(\{x_0, x_1, x_2\}\) and \(\{h_0, h_1, h_2\}\) can be found by the cyclic convolution of \(\{x_0, x_1, x_2, 0, 0\}\) and \(\{h_0, h_1, h_2, 0, 0\}\). In general,

\[
\begin{align*}
\{X_{0, k_i}, X_{1, k_i}, \ldots, X_{N_2 - 1, k_i}, 0, \ldots, 0\} & \star \{H_{0, k_i}, H_{1, k_i}, \ldots, H_{N_2 - 1, k_i}, 0, \ldots, 0\} \\
& \sim \{H_{0, k_i}, H_{1, k_i}, \ldots, H_{N_2 - 1, k_i}, 0, \ldots, 0\}
\end{align*}
\]

where \(\star\) means cyclic convolution. This cyclic convolution can be computed via the Winograd’s short convolution algorithms [15] or the number theoretic transform again. The former method requires \((2N_2 - 1) - K\) multiplications per \(N_2\) points whereas the latter method requires \((2N_2 - 1)\) multiplications per \(N_2\) points, where \(K\) is the number of divisors of \((2N_2 - 1)\) including one and \(N_2\). In the case of the number theoretic transform, the best transform length is \((2N_2 - 1)\), which is an odd number, and hence Mersenne number transforms, pseudo Mersenne transforms [16] and pseudo Fermat transforms [17] can be used.

\[
X_{k_2', k_1} = \left\langle \sum_{n_2=0}^{2N_2 - 2} X_{n_2, k_1} \delta^{k_1 n_2} \right\rangle_{M'}
\]

where

\[
k_2' = 0, 1, \ldots, 2N_2 - 2
\]

\[
M' = \text{base for modulo arithmetic}
\]

\[
\delta = \text{the root of unity of the transform}
\]

Since the results of the second NTT are the sum of the products \(X_{n_2, k_1} H_{n_2, k_1}\), the wordlength for \(M'\) should be at least twice as large as \(M\). This is the major requirement that has to be satisfied in order to reduce the number of multiplications involved in the convolution. However, this particular requirement is eliminated if \(M'\) is chosen to be \(M\). This is possible since the evaluation of eqn. 17 is to be done modulo \(M\). This reduction in the modulo length requirement is of great importance in actual applications. This does not only reduce the wordlength for the multiplication, but it simplifies some other processes involved in the transformation. For example, the results of the second NTT have to be reduced to \(M\) before they can be applied to eqn. 17:

\[
W_{n_2', k_1} = \left\langle \sum_{k_2=0}^{2N_2 - 2} X_{k_2', k_1} H_{k_2', k_1} \delta^{-k_1 n_2'} \right\rangle_{M'}
\]

However, if \(M' = M\), eqn. 22 reduces to

\[
W_{n_2', k_1} = \left\langle \sum_{k_2=0}^{2N_2 - 2} X_{k_2', k_1} H_{k_2', k_1} \delta^{-k_1 n_2'} \right\rangle_{M}
\]

Only one reduction is required for this step in using eqn. 23 instead of eqn. 22. Simplicity in programming the software implementation and also simplicity in the construction of hardware implementations are a consequence of choosing both moduli to be the same. For example, the cyclic convolution of two length 496 \((= 31 \times 16)\) sequences can be implemented by two length 31 cyclic convolution using MNT with a modulo base of \(M = 2^{31} - 1\), \(a = 2\) and \(N = 31\). The number of multiplications per output point is always less than two.

To make the ideas contained here clearer an example is included in Appendix 8.1.

4 Two-dimensional convolution

In this Section we examine the two-dimensional case of the cyclic convolution and derive results parallel to the previous case. The basic concepts remain the same and the saving in the number of multiplications achievable is considerable.

The \(z\)-transform of the two-dimensional sequences \(x(n, r) = \{x(n, r); n = 0, 1, \ldots, N - 1\}\) and \(h(m, s) = \{h(m, s); m = 0, 1, \ldots, N - 1\}\) can be written as

\[
X(z, q) = \sum_{n=0}^{N-1} \sum_{r=0}^{N-1} x(n, r) q^n z^n
\]

\[
H(z, q) = \sum_{m=0}^{N-1} \sum_{s=0}^{N-1} h(m, s) q^m z^m
\]

Now if the length \(N\) is a composite number, then let \(N = N_1 N_2\) and

\[
\begin{align*}
n &= N_1 n_1 + N_2 \\
r &= N_1 r_1 + N_2 \\
m &= N_1 m_1 + m_2 \\
s &= N_2 s_1 + s_2
\end{align*}
\]

where

\[
\begin{align*}
n_1, r_1, m_1, s_1 &= 0, 1, \ldots, N_1 - 1 \\
n_2, r_2, m_2, s_2 &= 0, 1, \ldots, N_2 - 1
\end{align*}
\]

Also, let

\[
\begin{align*}
z_1 &= q^{N_2} \\
q_1 &= q^{N_1}
\end{align*}
\]
Eqs. 24 and 25 can now be written as

\[
X(z, z_1, q, q_1) = \sum_{n_2=0}^{N_2-1} \sum_{r_2=0}^{N_2-1} X_{n_2, r_2, 1,1} q_2^{n_2} z_2^{r_2} z^{n_2} z_2^{r_2} z^{n_2} \tag{26}
\]

\[
H(z, z_1, q, q_1) = \sum_{m_2=0}^{N_2-1} \sum_{s_2=0}^{N_2-1} H_{m_2, s_2, 1,1} q_2^{m_2} z_2^{s_2} z^{m_2} z_2^{s_2} z^{n_2} \tag{27}
\]

On taking the number theoretic transform with respect to \((n_2, r_2)\) and \((m_2, s_2)\), respectively, we obtain

\[
X_{k_1, i_1}(z, q) = \sum_{n_2=0}^{N_2-1} \sum_{r_2=0}^{N_2-1} X_{n_2, r_2, k_1, i_1} q_2^{n_2} z_2^{r_2} z^{n_2} z_2^{r_2} z^{n_2} \tag{28}
\]

\[
H_{k_1, i_1}(z, q) = \sum_{m_2=0}^{N_2-1} \sum_{s_2=0}^{N_2-1} H_{m_2, s_2, k_1, i_1} q_2^{m_2} z_2^{s_2} z^{m_2} z_2^{s_2} z^{n_2} \tag{29}
\]

where

\[
X_{n_2, r_2, k_1, i_1} = \sum_{n_2=0}^{N_2-1} \sum_{r_2=0}^{N_2-1} X_{n_2, r_2, 1,1} \beta_i^{n_2} \beta_i^{r_2} \beta_i^{k_1} \beta_i^{i_1} \tag{30}
\]

\[
H_{m_2, s_2, k_1, i_1} = \sum_{m_2=0}^{N_2-1} \sum_{s_2=0}^{N_2-1} H_{m_2, s_2, 1,1} \beta_i^{m_2} \beta_i^{s_2} \beta_i^{k_1} \beta_i^{i_1} \tag{31}
\]

In these equations \(\beta\) and \(\alpha\) are the roots of unity of the transform and usually \(\beta = \alpha\). The NTT of the convolution of the sequences \(x(n, r)\) and \(h(m, s)\) can now be written as

\[
X_{k_1, i_1}(z, q) H_{k_1, i_1}(z, q) = \sum_{n_2=0}^{N_2-1} \sum_{r_2=0}^{N_2-1} X_{n_2, r_2, k_1, i_1} H_{n_2, r_2, k_1, i_1} q_2^{n_2} z_2^{r_2} z^{n_2} z_2^{r_2} z^{n_2} \tag{32}
\]

This equation is similar to eqn. 17 for the one-dimensional convolution. Direct computation of eqn. 30 requires \([N_2^2]^2 = N_2^4\) multiplications for \(N_2^2\) points. However the computation of the two-dimensional Lagrange interpolation formula as before requires \((2N^2 - 2)^2\) multiplications for \(N_2^2\) points only. The number of multiplications per point is then \((2 - (1/N_2))^2\), which is always less than four.

For example, if \(N = 256\) then we can let \(N_2 = 2, N_1 = 128\) and use Fermat number \(F_5\) at the base for modulo arithmetic; i.e. \(M = F_5 = 2^{32} + 1\). The cyclic convolution of the sequences \(\{x_n, s; n = 0, 1, \ldots, 255\}\) and \(\{h_m, s; m = 0, 1, \ldots, 255\}\) can then be written as

\[
y_{u, v} = \sum_{n=0}^{255} \sum_{r=0}^{255} x_{u, r} h_{v, u-v-r} \quad \text{for} \quad u, v = 0, 1, \ldots, 255
\]

Hence, eqn. 30 becomes

\[
X_{k_1, i_1}(z, q) H_{k_1, i_1}(z, q) = \left[ \sum_{n_2=0}^{N_2-1} \sum_{r_2=0}^{N_2-1} X_{n_2, r_2, k_1, i_1} q_2^{n_2} z_2^{r_2} z^{n_2} z_2^{r_2} z^{n_2} \right] \times \left[ \sum_{m_2=0}^{N_2-1} \sum_{s_2=0}^{N_2-1} H_{m_2, s_2, k_1, i_1} q_2^{m_2} z_2^{s_2} z^{m_2} z_2^{s_2} z^{n_2} \right] \tag{33}
\]
Using eqn. 5, eqn. 31 can be written as

\[ Y_{n_i, l_i}(z, q) = \left[ W_{0, 0} \cdot 0, 1, l_i + W_{0, 2, 2, k_i, 1} \cdot \beta^{n_i} 
+ W_{2, 0} \cdot 2, 2, k_i, 1 \cdot \beta^{2 \cdot z + q} 
+ (W_{0, 1} \cdot 0, 1, l_i + W_{2, 2, 2, k_i, 1} \cdot \beta^{2 \cdot z + q}) \right] M 
+ (W_{0, 1} \cdot 0, 1, l_i + W_{1, 2, 2, k_i, 1} \cdot \beta^{0 \cdot z + q}) z 
+ (W_{1, 1} \cdot 1, 1, l_i) z q \]

and

\[ Y_{n_i, l_i}(z, q) = \left[ W_{0, 0} \cdot 0, 1, l_i + W_{0, 2, 2, k_i, 1} \cdot \beta^{n_i} 
+ W_{2, 0} \cdot 2, 2, k_i, 1 \cdot \beta^{2 \cdot z + q} \right] M 
+ (W_{0, 1} \cdot 0, 1, l_i + W_{2, 2, 2, k_i, 1} \cdot \beta^{2 \cdot z + q}) \]

The result \( Y_{n_2, r_2, n_1, r_1} \) can then be found by four length-128 inverse FNT:

\[ \sum_{k_i=0}^{127} \sum_{l_i=0}^{127} Y_{n_2, r_2, n_1, r_1} \mathbf{x}^{-\left(n_1 \cdot k_i + r_1 \cdot l_i\right)} \]

where \( n_1, r_1 = 0, 1, \ldots, 127 \)

\( n_2, r_2 = 0, 1 \)

From eqn. 35, it is clear that the number of multiplications per point is 2.25. This example is similar to the one given by Kraats and Venetsanopoulos [18], but our derivation is much more general and systematic. It is also interesting to point out that the calculation of \( \{X_0, 0, 0, k_i, l_i, X_0, 2, 2, k_i, l_i, X_2, 0, k_i, l_i, X_2, 2, 2, k_i, l_i\} \) actually corresponds to a length-4 Hadamard transform, and hence the fast-Hadamard-transform algorithm [19] can be applied in this computation, producing as intermediate results the required values \( \{X_0, 0, 0, k_i, l_i, X_0, 2, 2, k_i, l_i\} \).

It should be noted that, as in the one-dimensional case, if \( N_2 \) is relatively large, a second two-dimensional NTT can be used instead of the two-dimensional Lagrange interpolation formula. If the same modulo base is also employed (i.e. \( M' = M \) then \( N_1 \) is usually equal to \( C(2N_2 - 1) \), where \( C \) is an integer or the reciprocal of an integer. The number of multiplication per point is then always less than four.

5 Nested number theoretic transform

If the factors \( N_1 \) and \( N_2 \) of a given sequence length are relatively prime, then one may map the one-dimensional cyclic convolution into a two-dimensional cyclic convolution. The map could be written in a linear form as follows:

\[ n = K_1 \cdot n_1 + K_2 \cdot n_2 \mod N \]

Bur-rus [9] shows that the map is cyclic in \( n_1 \) and \( n_2 \) if and only if \( K_1 = CN_2 \) and \( K_2 = DN_1 \). For a unique map, \( C \) and \( N_1 \) (also \( D \) and \( N_2 \)) have to be relatively prime.
Let 
\[ n = \langle N_2 n_1 + N_1 n_2 \rangle_N \]
for \( n_1 = 0, 1, \ldots, N_1 - 1 \)
\[ n_2 = 0, 1, \ldots, N_2 - 1 \]

Hence eqn. 8 becomes
\[ X(z, z_1) = \sum_{n_2=0}^{N_2-1} \sum_{n_1=0}^{N_1-1} x_{n_2, n_1} z_1^{n_1} z_2^{n_2} \mod (z_1^{N_1} - 1) \mod (z_2^{N_2} - 1) \]
we can also make the same arrangement for \( h(n) \). Eqsns. 15 and 16 remain the same as before. But eqn. 17 becomes
\[ X_{h_1}(z)H_{h_1}(z) = \left( \sum_{n_2=0}^{N_2-1} X_{n_2, k_1} z_1^{n_2} \right) \left( \sum_{m_2=0}^{N_2-1} H_{m_2, k_1} z_2^{m_2} \right) \mod (z_1^{N_2} - 1) \mod (z_2^{N_2} - 1) \]
Eqn. 37 is the cyclic convolution of the sequences \( \{X_{0, k_1}, X_{1, k_1}, \ldots, X_{N_2-1, k_1}\} \) and \( \{H_{0, k_1}, H_{1, k_1}, \ldots, H_{N_2-1, k_1}\} \). This cyclic convolution sum can be found by any fast cyclic convolution algorithm, for example Winograd’s short convolution algorithms [15]. However, we suggest here that this can be done by a second number theoretic transform. Since \( N_1 \) and \( N_2 \) are relatively prime, only one FNT can be employed for two NTTs. The other NTT can be a Mersenne transform, a pseudo Mersenne transform, a pseud Fermat transform or any NTT with an odd transform length. In Appendix B.2 the same set of data as in Appendix 8.1 is used to calculate the convolution sum of two sequences. It can be seen that no multiplication is required for the first NTT, and \( N_2 \times N_1 (= N) \) multiplications are required for the computation of the second NTT. This method can easily be generalised to compute convolution of multidimensional sequences. The number of multiplications remains as one per output point irrespective of the dimension of the sequence.

6 Conclusion
We have seen that the computation of the cyclic convolution sum of two length-\( N \) sequences using the number theoretic transform can always be done by two-dimensional (or multidimensional) techniques. The NTT can always be applied to the dimension with cyclic rearrangement. If the factors \( (N_1, N_2) \) of \( N \) are not mutually prime, the Lagrange interpolation formula or a second NTT can be used to find the convolution sum. This requires \( (2 - 1/N) \) multiplications per point, where \( i = 1, \) or 2. However, if \( N_1 \) and \( N_2 \) are mutually prime, a nested NTT approach can be used to compute the cyclic convolution sum. This requires one multiplication per point. For two-dimensional convolutions, a two-dimensional Lagrange interpolation formula or a second NTT can be used in the computation to give less than four multiplications per point if \( N_1 \) and \( N_2 \) are not mutually prime. However, if \( N_1 \) and \( N_2 \) are mutually prime, one multiplication per point is enough for the computation.

7 References

8 Appendixes
8.1 Consider the circular convolution of the sequences \( x(n) = \{4, 0, 1, 2, 3, 1\} \) and \( h(n) = \{4, 2, 5, 0, 0, 1\} \). Eqn. 10 can be written as
\[ y = \sum_{n=0}^{5} x_n h_{i-n} \]
for \( i = 0, 1, \ldots, 5 \)
A direct evaluation of this equation gives \( y(n) = \{33, 14, 26, 13, 22, 24\} \).
To compute the cyclic convolution of these two sequences by eqns. 14-16, let \( N_2 = 2, N_1 = 3 \) and
\[ n = n_2 + 2n_1 \]
where \( n_2 = 0, 1 \) (noncyclic)
\[ n_1 = 0, 1, 2 \) (cyclic)
From eqn. 7,
\[ z_1 = z^2 \]
hence,
\[ x_{0, 0} = x_0 = 4 \]
\[ h_{0, 0} = h_0 = 4 \]
\[ x_{0, 1} = x_1 = 2 \]
\[ h_{0, 1} = h_2 = 5 \]
\[ x_{0, 2} = x_2 = 3 \]
\[ h_{0, 2} = h_4 = 0 \]
\[ x_{1, 0} = x_1 = 0 \]
\[ h_{1, 0} = h_1 = 2 \]
\[ x_{1, 1} = x_3 = 2 \]
\[ h_{1, 1} = h_3 = 0 \]
\[ x_{1, 2} = x_5 = 1 \]
\[ h_{1, 2} = h_5 = 1 \]
Using a pseudo Mersenne transform, with
\[ M = \frac{2^9 - 1}{2^3 - 1} = 73 \]
\[ n = 2^3 \]
\[ N_1 = 3 \]
eqn. 15 becomes
\[
\begin{bmatrix}
X_{n_2,0} \\
X_{n_2,1} \\
X_{n_2,2}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
1 & 2^3 & 2^6 \\
1 & 2^6 & 2^{12}
\end{bmatrix}
\begin{bmatrix}
x_{n_2,0} \\
x_{n_2,1} \\
x_{n_2,2}
\end{bmatrix}
\text{mod 73}
\]
and
\[
\begin{bmatrix}
X_{0,0} \\
X_{0,1} \\
X_{0,2}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
1 & 2^3 & 2^6 \\
1 & 2^6 & 2^3
\end{bmatrix}
\begin{bmatrix}
4 & 1 & 58 \\
1 & 19 & 0
\end{bmatrix}
\]
Similarly,
\[
\begin{bmatrix}
X_{1,0} \\
X_{1,1} \\
X_{1,2}
\end{bmatrix}
= \begin{bmatrix}
3 & 7 & 63 \\
9 & 44 & 32 \\
3 & 66 & 10
\end{bmatrix}
\]
Eqn. 17 becomes
\[
X_k(z)H_k(z) = \left(\sum_{n_2=0}^1 X_{n_2,k} z^{n_2}\right)\left(\sum_{m_2=0}^1 H_{m_2,k} z^{m_2}\right)
= W_{0,k_1} + W_{1,k_1} z + W_{2,k_1} z^2
= (W_{0,k_1} + z_1 W_{2,k_1}) + W_{1,k_1} z
\]
By eqn. 5
\[
X_k(z)H_k(z) = (W_{0,k_1} + 2^{3k_1} W_{2,k_1}) + W_{1,k_1} z
\]  (38)
(i) By Lagrange interpolating formula
Note that the procedures are mainly used to illustrate the steps of implementation and no attempt has been given to optimise the number of additions. Let \( z_i = -1, 0 \) and 1 for \( i = 0, 1, 2 \) in eqn. 20. Hence,
\[
m_{k_1,0} = (X_{0,k_1} - X_{1,k_1})(H_{0,k_1} - H_{1,k_1})
\]
\[
m_{k_1,1} = (X_{0,k_1} - H_{0,k_1})
\]
\[
m_{k_1,2} = (X_{0,k_1} + X_{1,k_1})(H_{0,k_1} + H_{1,k_1})
\]
and
\[
\begin{bmatrix}
m_{0,0} \\
m_{0,1} \\
m_{0,2}
\end{bmatrix}
= \begin{bmatrix}
8 & 3 & 7 \\
8 & 9 & 72 \\
8 & 3 & 132
\end{bmatrix}
\begin{bmatrix}
30 \\
72 \\
132
\end{bmatrix}
\]
Similarly,
\[
\begin{bmatrix}
m_{1,0} \\
m_{1,1} \\
m_{1,2}
\end{bmatrix}
= \begin{bmatrix}
-1122 & 2552 & 7150 \\
2552 & 608 & 3444 \\
7150 & 608 & 3444
\end{bmatrix}
\]
The interpolating polynomials becomes (by eqn. 19)
\[
L_0(z) = \left(\frac{z - 0}{-1 - 1}\right)\left(\frac{z - 1}{-1 - 1}\right) = \frac{1}{2}(z^2 - z)
\]
Similarly,
\[
L_1(z) = -(z^2 - 1)
\]
and
\[
L_2(z) = 1/2(z^2 + z)
\]
Hence, by eqn. 18,
\[
\begin{bmatrix}
W_{0,k_1} \\
W_{1,k_1} \\
W_{2,k_1}
\end{bmatrix}
= \frac{1}{2}\begin{bmatrix}
0 & 2 & 0 \\
1 & 0 & 1 \\
1 & -2 & 1
\end{bmatrix}
\begin{bmatrix}
m_{k_1,0} \\
m_{k_1,1} \\
m_{k_1,2}
\end{bmatrix}
\]
This gives
\[
\begin{bmatrix}
W_{0,0} \\
W_{1,0} \\
W_{2,0}
\end{bmatrix}
= \begin{bmatrix}
72 & 51 & 9 \\
12 & 4136 & 462 \\
608 & 2206 & 630
\end{bmatrix}
\]
and
\[
\begin{bmatrix}
W_{0,1} \\
W_{1,1} \\
W_{2,1}
\end{bmatrix}
= \begin{bmatrix}
70 & 48 & 24 \\
12 & 46 & 46 \\
12 & 36 & 24
\end{bmatrix}
\]
mod 73
(ii) By a second number theoretic transform
Let us use the same \( M, \alpha \) and \( N \) for the second NTT.
\[
\begin{bmatrix}
X'_{0,k_1} \\
X'_{1,k_1} \\
X'_{2,k_1}
\end{bmatrix}
= \begin{bmatrix}
1 & 1 & 1 \\
1 & 2^3 & 2^6 \\
1 & 2^6 & 2^3
\end{bmatrix}
\begin{bmatrix}
X_{0,k_1} \\
X_{1,k_1} \\
X_{2,k_1}
\end{bmatrix}
\text{mod 73}
\]
Hence,
\[
\begin{bmatrix}
X'_{0,0} \\
X'_{1,0} \\
X'_{2,0}
\end{bmatrix}
= \begin{bmatrix}
11 & 32 & 54 \\
12 & 41 & 68 \\
12 & 36 & 68
\end{bmatrix}
\begin{bmatrix}
X_{0,0} \\
X_{1,0} \\
X_{2,0}
\end{bmatrix}
\text{mod 73}
\]
and similarly,
\[
\begin{bmatrix}
H'_{0,0} \\
H'_{1,0} \\
H'_{2,0}
\end{bmatrix}
= \begin{bmatrix}
12 & 33 & 55 \\
37 & 61 & 34 \\
42 & 39 & 15
\end{bmatrix}
\begin{bmatrix}
H_{0,0} \\
H_{1,0} \\
H_{2,0}
\end{bmatrix}
\text{mod 73}
\]
Let
\[
W'_{k_2,k_1} = X'_{k_2,k_1} H'_{k_2,k_1} \text{ mod 73}
\]
Therefore,
\[
\begin{bmatrix}
W'_{0,0} \\
W'_{1,0} \\
W'_{2,0}
\end{bmatrix}
= \begin{bmatrix}
59 & 34 & 50 \\
69 & 19 & 49 \\
13 & 30 & 29
\end{bmatrix}
\begin{bmatrix}
W_{0,0} \\
W_{1,0} \\
W_{2,0}
\end{bmatrix}
\text{mod 73}
\]
and
\[
\begin{bmatrix}
W'_{0,1} \\
W'_{1,1} \\
W'_{2,1}
\end{bmatrix}
= \frac{1}{3}\begin{bmatrix}
1 & 1 & 1 \\
2^{-3} & 2^{-6} & 2^{-12} \\
1 & 2^{-6} & 2^{-12}
\end{bmatrix}
\begin{bmatrix}
W'_{0,1} \\
W'_{1,1} \\
W'_{2,1}
\end{bmatrix}
\text{mod 73}
\]
Hence,

\[
\begin{align*}
W_{0,0} &= 72 \\
W_{1,0} &= 51 \\
W_{2,0} &= 9
\end{align*}
\]

\[
\begin{align*}
W_{0,1} &= 70 \\
W_{1,1} &= 48 \\
W_{2,1} &= 24
\end{align*}
\]

\[
\begin{align*}
W_{0,2} &= 16 \\
W_{1,2} &= 46 \\
W_{2,2} &= 46
\end{align*}
\]

These results are exactly the same as those we obtained in (i).

Eqn. 38 can now be written as

\[
Y_{0,0} = <W_{0,0} + 2^{31}W_{2,0},>_73
\]

This gives

\[
\begin{align*}
Y_{0,0} &= 8 \\
Y_{1,0} &= 51 \\
Y_{0,1} &= 43
\end{align*}
\]

The final results can be obtained by

\[
\begin{align*}
Y_{0,0} &= 1/3 \\
Y_{1,0} &= 2 \mod 73
\end{align*}
\]

Hence,

\[
\begin{align*}
y_{0,0} &= 33 \\
y_{0,1} &= 26 \\
y_{0,2} &= 22
\end{align*}
\]

and

\[
\begin{align*}
y_{1,0} &= 14 \\
y_{1,1} &= 13 \\
y_{1,2} &= 24
\end{align*}
\]

8.2

Let us now find the convolution sum of the sequences

\[
x(n) = \{4, 0, 1, 2, 3, 1\}
\]

and

\[
h(n) = \{4, 2, 5, 0, 0, 1\}
\]

of Appendix 8.1 by nested NTT. Let \(N_2 = 2\) and \(N_1 = 3\) and

\[
n = 3n_2 + 2n_1
\]

Hence,

\[
\begin{align*}
x_{0,0} &= 4 \\
x_{0,1} &= 1 \\
x_{0,2} &= 3 \\
x_{1,0} &= 2 \\
x_{1,1} &= 1 \\
x_{1,2} &= 0
\end{align*}
\]

Again, let us use the pseudo Mersenne transform with

\[
M = (2^9 - 1)/(2^3 - 1), \quad n = 3
\]

This gives

\[
\begin{align*}
X_{0,0} &= 8 \\
X_{0,1} &= 58 \\
X_{0,2} &= 19
\end{align*}
\]

\[
\begin{align*}
X_{1,0} &= 3 \\
X_{1,1} &= 10 \\
X_{1,2} &= 66
\end{align*}
\]

\[
\begin{align*}
H_{0,0} &= 9 \\
H_{0,1} &= 44 \\
H_{0,2} &= 32
\end{align*}
\]

\[
\begin{align*}
H_{1,0} &= 3 \\
H_{1,1} &= 63 \\
H_{1,2} &= 7
\end{align*}
\]

Since \(N_2 = 2, (z^2 - 1) = 0\) and eqn. 37 becomes

\[
Y_{0,0}(z) = H_{1,0}(z)X_{1,0}(z)
\]

\[
= (X_{0,0}H_{0,0} + X_{1,0}H_{1,0}) + (X_{0,1}H_{0,1} + X_{1,1}H_{1,1})z
\]

\[
= Y_{0,0} + Y_{1,0}z
\]

The second transformation can be done by FNT with

\[
M = F_4 = 2^{16} + 1, N_2 = 2 \quad \text{and} \quad M = 2^{16}
\]

This gives

\[
\begin{align*}
X_{0,0} &= [11] \\
X_{1,0} &= [8] \\
X_{0,1} &= [43] \\
X_{1,1} &= [6]
\end{align*}
\]

Similarly,

\[
\begin{align*}
H_{0,0} &= [12] \\
H_{1,0} &= [8] \\
H_{0,1} &= [107] \\
H_{1,1} &= [65518]
\end{align*}
\]

\[
\begin{align*}
Y_{0,0} &= [8] \\
Y_{1,0} &= [51] \\
Y_{0,1} &= [43] \\
Y_{1,1} &= [6]
\end{align*}
\]

The inverse transform is

\[
\begin{align*}
Y_{0,0} &= 33 \\
Y_{1,0} &= 26 \\
Y_{0,1} &= 22
\end{align*}
\]

Hence,

\[
\begin{align*}
y_{0,0} &= 33 \\
y_{0,1} &= 14 \\
y_{0,2} &= 26
\end{align*}
\]

8.2

Let us now find the convolution sum of the sequences

\[
x(n) = \{4, 0, 1, 2, 3, 1\}
\]

and \(h(n) = \{4, 2, 5, 0, 0, 1\}\) of Appendix 8.1 by nested NTT. Let \(N_2 = 2\) and \(N_1 = 3\) and

\[
n = 3n_2 + 2n_1
\]

Hence,

\[
\begin{align*}
x_{0,0} &= 4 \\
x_{0,1} &= 1 \\
x_{0,2} &= 3 \\
x_{1,0} &= 2 \\
x_{1,1} &= 1 \\
x_{1,2} &= 0
\end{align*}
\]

Making the inverse transform of base 73, we get

\[
\begin{align*}
y_{0,0} &= 33 \\
y_{0,1} &= 14 \\
y_{0,2} &= 22
\end{align*}
\]

This gives the same set of results as those given in Appendix 8.1. As shown in eqn. 39, the total number of multiplications is 6. In general the nested number theoretic transform requires only one multiplication per output.