A POLYNOMIAL APPROACH TO FAST ALGORITHMS FOR DISCRETE FOURIER-COSINE AND FOURIER-SINE TRANSFORMS

G. STEIDL AND M. TASCHE

Abstract. The discrete Fourier-cosine transform (cos-DFT), the discrete Fourier-sine transform (sin-DFT) and the discrete cosine transform (DCT) are closely related to the discrete Fourier transform (DFT) of real-valued sequences. This paper describes a general method for constructing fast algorithms for the cos-DFT, the sin-DFT and the DCT, which is based on polynomial arithmetic with Chebyshev polynomials and on the Chinese Remainder Theorem.

1. Introduction

In this paper, we use standard notation. By \( \mathbb{N}, \mathbb{Z}, \mathbb{R} \) and \( \mathbb{C} \), we denote the set of positive integers, the ring of integers, the field of reals and the field of complex numbers. For two polynomials \( X, Y \) we let \( X \mod Y \) signify the remainder of \( X \) divided by \( Y \).

One of the most important tools in numerical analysis and digital signal processing is the fast Fourier transform (FFT), which efficiently computes the discrete Fourier transform of length \( N \) (DFT(\( N \))), a mapping of a sequence \( x = (x_0, \ldots, x_{N-1}) \in \mathbb{C}^N \) to its spectrum \( \hat{x} = (\hat{x}_0, \ldots, \hat{x}_{N-1}) \in \mathbb{C}^N \) defined by

\[
\hat{x}_k := \sum_{j=0}^{N-1} x_j w_N^{jk}, \quad w_N := \exp(-2\pi i/N).
\]

Using polynomial arithmetic, the formulation of many FFT-algorithms can be greatly simplified and their derivation seems more natural [1, 3, 10, 16]. Further, the polynomial notation can be utilized for considerations of the computational complexity of FFT's [6, 17].

In order to introduce a polynomial representation of the DFT, we represent the input sequence \( x \in \mathbb{C}^N \) of the DFT(\( N \)) as the polynomial

\[
X(z) := \sum_{j=0}^{N-1} x_j z^j.
\]
Then we have \( \hat{x}_k = X(w_N^k) \) (\( k = 0, \ldots, N - 1 \)), i.e.,

\[
(1.1) \quad \hat{x}_k = X(z) \mod(z - w_N^k) \quad (k = 0, \ldots, N - 1).
\]

Since

\[
z^N - 1 = \prod_{k=0}^{N-1} (z - w_N^k),
\]

we get a fast algorithm for the DFT(\( N \)) by the Chinese Remainder Theorem (CRT) [10, pp. 26–27], if we split \( X(z) \mod(z^N - 1) \) stepwise into equivalent simultaneous remainders, using the successive factorization of \( z^N - 1 \), such that we ultimately obtain the desired simultaneous remainders (1.1). We illustrate this by the radix-2 FFT of Cooley and Tukey (see [1]).

Let \( N = 2^r \) (\( r \in \mathbb{N} \)). Then \( z^N - 1 \) can be decomposed successively as in Figure 1. This factorization is the foundation of the radix-2 FFT, which calculates the DFT(\( N \)) by the recursive reduction of the input polynomial \( X(z) \) modulo the factors of \( z^N - 1 \) in Figure 1. The \( r \)th step of this reduction procedure yields the spectrum \( \hat{x} \in \mathbb{C}^N \).

Taking into account that most DFT's are taken on real data, many fast algorithms for real DFT's were published in recent years. These algorithms exploit directly the symmetries of the real DFT [14] or use transforms, which map a real-valued sequence to a real-valued spectrum as the discrete Hartley transform [13], the DCT, the cos-DFT and the sin-DFT [15]. Although the advantage of the polynomial arithmetic for the FFT is well known, there does not exist a convenient polynomial approach to the DCT, the cos-DFT and the sin-DFT up to now. This indeed is the task of our paper. Using Chebyshev polynomials, we define the DCT, the cos-DFT and the sin-DFT on a polynomial basis. We show that this representation leads to the descriptive derivation of fast algorithms for these transforms.

Section 2, where useful properties of Chebyshev polynomials are collected, has preliminary character. In §3, we suggest a new recursive algorithm for the DCT(2\(^r \)), which works with the same number of real operations as the best-known fast DCT's [7, 8, 15]. Introducing the cos-DFT and the sin-DFT, as well
as their reduced versions, we apply the polynomial arithmetic to decompose the reduced cos-DFT (reduced sin-DFT) of a length $N$ divisible by 4 into a DCT($N/4$) and a reduced cos-DFT (reduced sin-DFT) of length $N/2$ in §5 (cf. [15]). Our fast algorithms can be used to compute the DFT($2^f$) for real- and complex-valued sequences with the same computational complexity as the split-radix algorithm [3].

Although §§3 and 5 contain mainly the special case of fast algorithms for transforms of radix-2 length, the polynomial approach to fast algorithms for the DCT, the cos-DFT, and the sin-DFT of arbitrary highly factorizable lengths will be clear. We illustrate this idea by a fast DCT($3N$)-algorithm. Up to now, there do not exist fast algorithms for DCT's of such lengths. Note that, especially, the DCT has found wide applications in data compression and digital filtering.

2. Chebyshev polynomials

The polynomial approach to fast DCT's, cos-DFT's, and sin-DFT's is mainly based on known properties of Chebyshev polynomials, which we now summarize.

The Chebyshev polynomials of first and second kind can be defined recursively by

\[
T_0(z) := 1, \quad T_1(z) := z, \\
T_n(z) := 2zT_{n-1}(z) - T_{n-2}(z) \quad (n = 2, 3, \ldots), \\
T_n = T_{-n} \quad (n \in \mathbb{Z}),
\]

and by

\[
U_0(z) := 1, \quad U_1(z) := 2z, \\
U_n(z) := 2zU_{n-1}(z) - U_{n-2}(z) \quad (n = 2, 3, \ldots), \\
U_n = -U_{-n-2} \quad (n \in \mathbb{Z}),
\]

respectively [11, pp. 11–12]. From this it follows that

\[
T_n(z) = \cos(n \arccos z) \quad (|z| \leq 1; n \in \mathbb{Z}),
\]

\[
U_n(z) = (1 - z^2)^{-1/2} \sin((n + 1) \arccos z) \quad (|z| < 1; n \in \mathbb{Z}),
\]

and then

\[
T_n(z) = 2^{n-1} \prod_{k=0}^{n-1} (z - \cos(\pi(2k + 1)/2n)) \quad (n \in \mathbb{N}),
\]

\[
U_n(z) = 2^n \prod_{k=1}^{n} (z - \cos(\pi k/(n + 1))) \quad (n \in \mathbb{N}).
\]

We have [11, p. 24; 12, p. 5]

\[
T_{mn} = T_m(T_n) = 2^{m-1} \prod_{k=0}^{m-1} (T_n - \cos(\pi(2k + 1)/2m)) \quad (m, n \in \mathbb{N}).
\]
More generally, setting \( y := n \arccos z \) \((|z| \leq 1)\) in

\[
\cos my - \cos \alpha = 2^{m-1} \prod_{k=0}^{m-1} (\cos y - \cos((\alpha + 2\pi k)/m)) \quad (m \in \mathbb{N}; \alpha \in \mathbb{R})
\]

[5, p. 48], we obtain that

\[
(2.6) \quad T_{mn} - \cos \alpha = 2^{m-1} \prod_{k=0}^{m-1} (T_n - \cos((\alpha + 2\pi k)/m)) \quad (m, n \in \mathbb{N}; \alpha \in \mathbb{R}).
\]

Differentiation of (2.5) yields

\[
U_{mn-1} = U_{m-1}(T_n)U_{n-1}
\]

\[
= 2^{m-1} \prod_{k=1}^{m-1} (T_n - \cos(\pi k/m))U_{n-1} \quad (m, n \in \mathbb{N}).
\]

Furthermore, we shall use the properties [11, p. 24]

\[
(2.8) \quad T_mT_n = (T_{m-n} + T_{n+m})/2,
\]

\[
(2.9) \quad T_m(z)T_n(z) + (1 - z^2)U_{m-1}(z)U_{n-1}(z) = T_{m-n}(z),
\]

\[
(2.10) \quad U_{n-1}T_m + T_nU_{m-1} = U_{n+m-1}.
\]

As in [4], we define the polynomials \( W_n \) \((n \in \mathbb{N})\) by

\[
W_1(z) := z - 2, \quad W_2(z) := z + 2,
\]

\[
W_n(z) := \prod_{k=1}^{[n/2]} (z - (w_n^k + w_n^{-k}))
\]

\[
= \prod_{(k, n) = \gcd(k, n) = 1}^{[n/2]} (z - 2\cos(2\pi k/n)) \quad (n = 3, 4, \ldots),
\]

where \([n/2] := \max\{k \in \mathbb{Z} : k \leq n/2\}\) and where \((k, n)\) signifies the greatest common divisor of \(k\) and \(n\). For further properties of \( W_n \), especially the connection of \( W_n \) with Chebyshev polynomials, see [4]. Finally, let

\[
(2.11) \quad V_{m+1}(z) := \prod_{d|m} W_d(2z) = 2^{m+1} \prod_{k=0}^{m} (z - \cos(2\pi k/n))
\]

with \( m = \lfloor n/2 \rfloor \). If \( n \in \mathbb{N} \) is even, then we have by (2.4) and (2.11) that

\[
(2.12) \quad V_{m+1}(z) = 4(z^2 - 1)U_{m-1}(z).
\]

License or copyright restrictions may apply to redistribution; see http://www.ams.org/journal-terms-of-use
3. Discrete cosine transform

The discrete cosine transform of length $N$ (DCT($N$)) is defined by the following mapping between $x = (x_0, \ldots, x_{N-1}) \in \mathbb{R}^N$ and $\hat{x} = (\hat{x}_0, \ldots, \hat{x}_{N-1}) \in \mathbb{R}^N$:

$$x_k := \sum_{j=0}^{N-1} x_j \cos(\pi(2k+1)j/2N) \quad (k = 0, \ldots, N-1).$$

Note that our version of the DCT($N$) is similar to the inverse DCT in [7, 9, 15].

In order to introduce a polynomial notation for the DCT, we represent the $N$-point sequence $x \in \mathbb{R}^N$ as the polynomial

$$X := \sum_{j=0}^{N-1} x_j T_j.$$

Then, by (2.1), we have that (3.1) can be replaced by $\hat{x}_k = X(\cos(\pi(2k+1)/2N))$ ($k = 0, \ldots, N-1$), i.e.,

$$\hat{x}_k = X(z) \mod(z - \cos(\pi(2k+1)/2N)) \quad (k = 0, \ldots, N-1).$$

By (2.3) and by the CRT, we obtain a fast decimation in frequency algorithm for the DCT, if we split $X \mod T_N$ stepwise into the desired simultaneous remainders (3.3) by using polynomial factorizations of $T_N$.

In the following, let $N = 2^r$ ($r \in \mathbb{N}$). In this case, we get a successive factorization of $T_N$ by applying the following

**Lemma.** Let $s \in \mathbb{N}$ ($s > 1$), and let $a \in \mathbb{N}$ be an odd integer with the bit representation

$$a = (a_{s-1}, \ldots, a_1, 1)_2 := 2^{s-1}a_{s-1} + \cdots + 2a_1 + 1 \quad (a_i \in \{0, 1\}).$$

By $\oplus$, we denote the addition modulo 2. Then, for any $n \in \mathbb{N}$, there holds

$$T_{2n} - \cos(\pi a/2^s) = 2(T_n + \cos(\pi a/2^{s+1}))(T_n - \cos(\pi a/2^{s+1})),

T_{2n} - \cos(\pi a/2^s) = 2 \prod_{j=0}^{s-1} (T_n - \cos(\pi(j, j \oplus a_{s-1}, \ldots, j \oplus a_1, 1)_2/2^{s+1})).$$

**Proof.** Setting $m := 2$ and $\alpha := \pi a/2^s$ in (2.6), we get (3.4). The rest of the assertion follows from

$$\cos(\pi a/2^{s+1}) = -\cos(\pi(2^{s+1} - a)/2^{s+1})

= -\cos(\pi(2^s + 2^{s-1}(1 - a_{s-1}) + \cdots + 2(1 - a_1) + 1)/2^{s+1})

= -\cos(\pi(1, 1 - a_{s-1}, \ldots, 1 - a_1, 1)_2/2^{s+1})

= -\cos(\pi(1, 1 \oplus a_{s-1}, \ldots, 1 \oplus a_1, 1)_2/2^{s+1}). \quad \Box$$
The above lemma yields the following recursive factorization of $T_N = T_N - \cos(\pi/2)$:

1. $T_N = 2T^{(0)}T^{(1)}$ with
   
   $$T^{(j_1)} := T_{N/2} - (-1)^{j_1} \sqrt{2}/2 \quad (j_1 = 0, 1).$$

2. $T^{(j_1)} = 2T^{(j_1, 0)}T^{(j_1, 1)}$ for $j_1 = 0, 1$ with
   
   $$T^{(j_1, j_2)} := T_{N/4} - (-1)^{j_2} \cos(\pi(j_1, 1)_2/8) \quad (j_2 = 0, 1).$$

3. $T^{(j_1, \ldots, j_r)} := 2T^{(j_1, \ldots, j_{r-1}, 0)}T^{(j_1, \ldots, j_{r-1}, 1)}$ for $j_1, \ldots, j_{r-1} = 0, 1$ with
   
   $$T^{(j_1, \ldots, j_r)} := T_1 - (-1)^{j_1} \cos(\pi(j_{r-1}, j_r \oplus j_{r-2}, \ldots, j_1, 1)_2/2) \quad (j_r = 0, 1).$$

The $r$th step contains the linear factors $T^{(j_1, \ldots, j_r)}$ of $T_N$. Figure 2 describes the decomposition of $T_N$ by the so-called transform factors $\cos(\pi(j_1, j_1 \oplus j_{s-1}, \ldots, j_1 \oplus \cdots \oplus j_1 1)_2/2^{s+1})$.

For a given input sequence $x \in \mathbb{R}^N$, we consider the polynomial $X$ introduced in (3.2). By the CRT, we have for every $s = 1, \ldots, r$ that $X \mod T_N$ is uniquely determined by its residues $X \mod T^{(j_1, \ldots, j_s)} (j_1, \ldots, j_s = 0, 1)$. This leads to the following recursive DCT($N$)-algorithm:

1. Calculate $X \mod T^{(j_1)}$ for $j_1 = 0, 1$. Observing that by (2.8)
   
   $$T_{N/2+j} = 2T_{N/2}T_j - T_{N/2-j},$$
   
   we obtain
   
   $$X^{(j_1)} := \sum_{j=0}^{N/2-1} x_j^{(j_1)} T_j = X \mod T^{(j_1)}.$$
with
\[
x^{(j_1)}_j := \begin{cases} 
  x_0 + (-1)^j x_{N/2} \sqrt{2}/2 & \text{for } j = 0, \\
  x_j - x_{N-j} + (-1)^j x_{N/2+j} \sqrt{2} & \text{for } j = 1, \ldots, N/2 - 1.
\end{cases}
\]

2. Calculate \(X^{(j_1)} \mod T^{(j_1, j_2)}\) for \(j_1, j_2 = 0, 1\). Using (3.5), with \(N/4\) instead of \(N/2\), we get
\[
X^{(j_1, j_2)} := \sum_{j=0}^{N/2-1} x^{(j_1, j_2)}_j T_j = X^{(j_1)} \mod T^{(j_1, j_2)}
\]
with
\[
x^{(j_1, j_2)}_j := \begin{cases} 
  x^{(j_1)}_0 + (-1)^j x^{(j_1)}_{N/4} \cos(\pi j_1, 1)/8) & \text{for } j = 0, \\
  x^{(j_1)}_j - x^{(j_1)}_{N/2-j} + (-1)^j x^{(j_1)}_{N/4+j} 2 \cos(\pi j_1, 1)/8) & \text{for } j = 1, \ldots, N/4 - 1.
\end{cases}
\]

3. Calculate \(X^{(j_1, \ldots, j_r)} \mod T^{(j_1, \ldots, j_r)}\) for \(j_1, \ldots, j_r = 0, 1\). This yields the final result
\[
X^{(j_1, \ldots, j_r)} := x^{(j_1, \ldots, j_r)}_0 = X^{(j_1, \ldots, j_r-1)} \mod T^{(j_1, \ldots, j_r)}
\]
with
\[
x^{(j_1, \ldots, j_r)}_0 := x^{(j_1, \ldots, j_r-1)}_0 + (-1)^{j_r} x^{(j_1, \ldots, j_r-1)}_1
\cdot \cos(\pi (j_{r-1} \oplus j_{r-2} \oplus \cdots \oplus j_1, 1)/2N).
\]

By (3.3) and by the decomposition of \(T_N\), we see that \(x^{(j_1, \ldots, j_r)}_0 = \tilde{x}_k\) for the index \(k\) with
\[
k = (j_r, j_r \oplus j_{r-1}, \ldots, j_r \oplus \cdots \oplus j_1)_2.
\]

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{flow_graph.png}
\caption{Flow graph for the DCT(8) with \(\alpha := \sqrt{2}/2\), \(\beta_j := \cos(\pi j/8)\), and \(\gamma_j := \cos(\pi j/16)\)}
\end{figure}
This describes the permutation of the output values. The \((j_1, \ldots, j_r)\)th component of our output sequence is \(\hat{x}_k\) with \(k\) as in (3.6). Figure 3 shows the flow graph of the DCT(8).

Simple considerations yield the computational complexity of our DCT-algorithm. The \(s\)th step of the algorithm requires \(2^{s-1}(N/2^s)\) real multiplications and \(2^{s-1}(3N/2^s - 1)\) real additions. Thus, our DCT\((N)\)-algorithm with \(N = 2^r\ (r \in \mathbb{N})\) works with a total of

\[
\tilde{M}_N = \sum_{s=1}^{r} 2^{s-1}(N/2^s) = (N/2)r
\]

real multiplications and

\[
\tilde{A}_N = \sum_{s=1}^{r} 2^{s-1}(3N/2^s - 1) = (N/2)(3r - 2) + 1
\]

real additions. Hence, our polynomial algorithm has the same computational complexity as the best-known DCT-algorithms [7, 8, 15].

4. Cos-DFT and sin-DFT

The image \(\hat{x} \in \mathbb{C}^N\) of the DFT\((N)\) of a real-valued sequence \(x \in \mathbb{R}^N\) can be obtained by

\[
\hat{x}_{c,k} := \sum_{j=0}^{N-1} x_j \cos(2\pi kj/N) \quad (k = 0, \ldots, N - 1),
\]

\[
\hat{x}_{s,k} := \sum_{j=0}^{N-1} x_j \sin(2\pi kj/N) \quad (k = 0, \ldots, N - 1),
\]

and by

\[
\hat{x}_k = \hat{x}_{c,k} - i\hat{x}_{s,k} \quad (k = 0, \ldots, N - 1).
\]

The mappings defined by (4.1) and (4.2) are called the discrete Fourier-cosine transform of length \(N\) (cos-DFT\((N)\)) and the discrete Fourier-sine transform of length \(N\) (sin-DFT\((N)\)), respectively. In this section, we present a polynomial approach to the cos-DFT and the sin-DFT, which suggests fast algorithms for both transforms.

Let \(M := \lfloor N/2 \rfloor\). By

\[
\cos(2\pi kj(N - j)/N) = \cos(2\pi kj/N) \quad (j, k \in \mathbb{Z}),
\]

(4.1) can be rewritten as

\[
\hat{c}_k := \hat{x}_{c,k} = \hat{x}_{c,N-k} = \sum_{j=0}^{M} c_j \cos(2\pi kj/N) \quad (k = 0, \ldots, M)
\]

with

\[
c_j := \begin{cases} x_j & \text{for } j = 0 \text{ and } j = M \text{ if } 2|N, \\ x_j + x_{N-j} & \text{otherwise} \end{cases} \quad (j = 0, \ldots, M).
\]
We call (4.3) the reduced \textit{cos-DFT}(N). Similarly, by
\[
\sin(2\pi k(N - j)/N) = -\sin(2\pi j/N) \quad (j, k \in \mathbb{Z}),
\]
it follows from (4.2) that
\[
(4.4) \quad \hat{s}_k := \hat{x}_s, k = -\hat{x}_{s, N-k} = \sum_{j=1}^{M} s_j \sin(2\pi j/N) \quad (k = 1, \ldots, M)
\]
with
\[
s_j := \begin{cases} 
0 & \text{for } j = M \text{ if } 2|N, \\
x_j - x_{N-j} & \text{otherwise} 
\end{cases} \quad (j = 1, \ldots, M).
\]
Then (4.4) is said to be the reduced \textit{sin-DFT}(N). Hence we can calculate the \textit{cos-DFT}(N) (\textit{sin-DFT}(N)) by \(\lceil N/2 \rceil - 1\) additions and by the reduced \textit{cos-DFT}(N) (\textit{sin-DFT}(N)). Here, \(\lceil N/2 \rceil := \min\{k \in \mathbb{Z} : k \geq N/2\}\). In the following, we deal with these reduced transforms.

In order to introduce a polynomial notation of the reduced cos-DFT, we represent the input sequence \(c = (c_0, \ldots, c_M) \in \mathbb{R}^{M+1}\) as the polynomial
\[
C := \sum_{j=0}^{M} c_j T_j.
\]
Then we have by (2.1) that
\[
(4.5) \quad \hat{c}_k = C(\cos(2\pi k/N)) \quad (k = 0, \ldots, M), \quad \text{i.e.,}
\]
\[
(4.6) \quad \hat{c}_k = \text{mod}(z - \cos(2\pi k/N)) \quad (k = 0, \ldots, M).
\]
By (2.11), we obtain a fast algorithm for the reduced \textit{cos-DFT}(N) if we split \(C \text{ mod } V_{M+1}\) stepwise into equivalent simultaneous remainders by using successive factorization of \(V_{M+1}\) together with the CRT, such that we get (4.5) in the last step.

Analogously, we represent the input sequence \(s = (s_1, \ldots, s_M) \in \mathbb{R}^M\) of the reduced sin-DFT as the polynomial
\[
S := \sum_{j=1}^{M} s_j U_{j-1}.
\]
Then we see by (2.2) that (4.4) can be expressed as
\[
(4.7) \quad \hat{s}_k = \sin(2\pi k/N) S(\cos(2\pi k/N)) \quad (k = 1, \ldots, M), \quad \text{i.e.,}
\]
\[
(4.8) \quad \hat{s}_k = \sin(2\pi k/N) S(z) \text{ mod}(z - \cos(2\pi k/N)) \quad (k = 1, \ldots, M).
\]
It follows from (2.11) and from the CRT that we can deduce a fast \textit{sin-DFT}(N) if we reduce \(S(z) \text{ mod}(V_{M+1}(z)/2(z-1))\) successively into equivalent simultaneous residues by applying polynomial factorizations of \(V_{M+1}(z)/2(z-1)\), such that we ultimately obtain (4.7).

For even \(N \in \mathbb{N}\), (4.6) and (4.7) can be simplified to
\[
(4.9) \quad S = \sum_{j=1}^{M-1} s_j U_{j-1},
\]
(4.9) \[ \hat{s}_k = \sin(2\pi k/N)S(z) \mod(z - \cos(2\pi k/N)) \quad (k = 1, \ldots, M - 1), \]
since \( \hat{s}_M = 0 \). By (2.4), we get a fast algorithm for the reduced sin-DFT(N)
with even \( N \in \mathbb{N} \) by splitting \( S \mod U_{M-1} \) stepwise, such that we obtain (4.9)
in the end.

5. Fast algorithms for reduced cos-DFT and sin-DFT

In this section, we assume \( N \in \mathbb{N} \) divisible by 4. Set \( M := N/2 \). Based on
the factorization

(5.1) \[ U_{M-1} = U_1(T_{M/2})U_{M/2-1} = 2T_{M/2}U_{M/2-1}, \]
which follows immediately from (2.7), we show that the reduced cos-DFT can be
decomposed into the reduced cos-DFT\( (N/2) \) and the DCT\( (N/4) \). The reduced
sin-DFT can be handled analogously. This verifies a result in [15] from the
polynomial point of view.

By (2.12), (5.1), and by the CRT, \( C \mod V_{M+1} \) is completely determined by
its residues \( C \mod T_{M/2} \) and \( C \mod V_{M/2+1} \). First we evaluate \( C \mod T_{M/2} \)
by polynomial reductions. By \( T_j = -T_{M-j} \mod T_{M/2} \) \((j = 0, \ldots, M/2 - 1)\), we
verify that

(5.2) \[ C^{(1)} := \sum_{j=0}^{M/2-1} c_j^{(1)}T_j = C \mod T_{M/2} \]
with \( c_j^{(1)} := c_j - c_{M-j} \) \((j = 0, \ldots, M/2 - 1)\). Since by (2.12), (5.1) and (2.3),
\[ \hat{c}_{2k+1} = \left( (C \mod V_{M+1}) \mod T_{M/2} \right) \mod(T_1 - \cos(2\pi(2k+1)/N)) \]
\[ = C^{(1)} \mod(T_1 - \cos(\pi(2k+1)/M)) \quad (k = 0, \ldots, M/2 - 1), \]
the output values with odd indices of the reduced cos-DFT\( (N) \) can be calculated
by \( M/2 \) additions and by the DCT\( (M/2) \) given by (5.2).

On the other hand, by (2.9), we have \( T_j = T_{M-j} \mod V_{M/2+1} \) \((j = 0, \ldots, M/2 - 1)\), so that \( C \mod V_{M/2+1} \) is obtained by

(5.3) \[ C^{(2)} := \sum_{j=0}^{M/2} c_j^{(2)}T_j = C \mod V_{M/2+1} \]
with \( c_j^{(2)} := c_j + c_{M-j} \) \((j = 0, \ldots, M/2 - 1)\), \( c_{M/2}^{(2)} := c_{M/2} \). Then we have by
(2.12), (5.1), and (2.4) that
\[ \hat{c}_{2k} = \left( (C \mod V_{M+1}) \mod V_{M/2+1} \right) \mod(T_1 - \cos(2\pi(2k)/N)) \]
\[ = C^{(2)} \mod(T_1 - \cos(2\pi k/M)) \quad (k = 0, \ldots, M/2). \]
Hence, we get the output values with even indices of the reduced cos-DFT\( (N) \)
by \( M/2 \) additions and by the reduced cos-DFT\( (M) \) determined in (5.3).
Turning to the reduced sin-DFT, we take a similar approach. By (5.1) and
by the CRT, \( S \mod U_{M-1} \) is completely determined by its residues \( S \mod T_{M/2} \)
and \( S \mod U_{M/2-1} \). Considering that, by (2.10), \( U_{j-1} = U_{M-j-1} \mod T_{M/2} \)
\((j = 1, \ldots, M/2)\), we find that \( S \mod T_{M/2} \) is given by

\[
S^{(1)} := \sum_{j=1}^{M/2} s_j^{(1)} U_{j-1} = S \mod T_{M/2}
\]

with \( s_j^{(1)} := s_j + s_{M-j} \) \((j = 1, \ldots, M/2 - 1)\), \( s_{M/2}^{(1)} := s_{M/2} \). Instead of \( S^{(1)} \)
we consider

\[
s^{(1)}_j := \sum_{j=0}^{M/2-1} s_j^{(1)} T_j
\]

with \( s_j^{(1)} := s_{M/2-j} \) \((j = 0, \ldots, M/2 - 1)\). Then from

\[
\sin(2\pi(2k + 1)/N) U_{j-1} \cos(\pi(2k + 1)/M) = \sin(2\pi(2k + 1)j/N)
\]

\[
= (-1)^k \cos(2\pi(2k + 1)(M/2 - j)/N) = (-1)^k T_{M/2-j} \cos(\pi(2k + 1)/M))
\]

one verifies that

\[
\hat{s}_{2k+1} = \sin(2\pi(2k + 1)/N) ((S \mod U_{M-1}) \mod T_{M/2})
\]

\[
= \sin(2\pi(2k + 1)/N) S^{(1)} \mod (T_1 - \cos(2\pi(2k + 1)/N))
\]

\[
= (-1)^k \hat{S}^{(1)} \mod (T_1 - \cos(2\pi(2k + 1)/M)) \quad (k = 0, \ldots, M/2 - 1).
\]

Consequently, we obtain the output values with odd indices of the reduced
sin-DFT(\(N\)) by \(M/2 - 1\) additions and by the DCT(\(M/2\)) given by (5.4),
where we have to change the sign of the output values with indices congruent 3
modulo 4.

Next, by (2.10), we have \( U_{j-1} = -U_{M-j-1} \mod U_{M/2-1} \) \((j = 1, \ldots, M/2 - 1)\). Using this property, we form

\[
S^{(2)} := \sum_{j=1}^{M/2-1} s_j^{(2)} U_{j-1} = S \mod U_{M/2-1}
\]

with \( s_j^{(2)} := s_j - s_{M-j} \) \((j = 1, \ldots, M/2 - 1)\). Now we conclude from

\[
\hat{s}_{2k} = \sin(2\pi(2k)/N) ((S \mod U_{M-1}) \mod U_{M/2-1}) \mod (T_1 - \cos(2\pi(2k)/N))
\]

\[
= \sin(2\pi k/M) S^{(2)} \mod (T_1 - \cos(2\pi k/M)) \quad (k = 1, \ldots, M/2 - 1),
\]

that the output values with even indices of the reduced sin-DFT(\(N\)) can be
evaluated by \(M/2 - 1\) additions and by the reduced sin-DFT(\(M\)) determined
in (5.5).
Let \( N = 2^r \) \((r \geq 2)\). Then we can use the above reduction successively for the reduced \( \text{cos-DFT}(2^s) \) \((\text{reduced \sin-DFT}(2^s))\) with \( s = r, \ldots, 2 \). This results in the computation of the \( \text{cos-DFT}(N) \) \((\text{sin-DFT}(N))\) using only DCT's and some additions. See Figure 4 with \( u := 1 \).

The numbers \( M^c_N \) and \( M^s_N \) of real multiplications and the numbers \( A^c_N \) and \( A^s_N \) of real additions to perform the \( \text{cos-DFT}(N) \) and the \( \text{sin-DFT}(N) \), respectively, follow directly from Figure 4, (3.7) and (3.8). For \( N = 2^r \) \((r \geq 2)\), one obtains

\[
M^c_N = M^s_N = \sum_{s=1}^{r-2} \widetilde{M}_{2^s} = \frac{N}{4}(r - 3) + 1,
\]

\[
A^c_N = N/2 - 1 + \sum_{s=0}^{r-2} 2^s + \sum_{s=1}^{r-2} \widetilde{A}_{2^s} + 2 = \frac{N}{4}(3r - 5) + r + 2,
\]

\[
A^s_N = N/2 - 1 + \sum_{s=0}^{r-2} (2^s - 1) + \sum_{s=1}^{r-2} \widetilde{A}_{2^s} = \frac{N}{4}(3r - 5) - r + 2.
\]

Consequently, the \( \text{DFT}(N) \) of a real-valued sequence computed by our method requires

\[
M^r_N = M^c_N + M^s_N = \frac{N}{2}(r - 3) + 2,
\]

\[
A^r_N = A^c_N + A^s_N = \frac{N}{2}(3r - 5) + 4.
\]
real multiplications and real additions, respectively. Further, by
\[ \hat{x}_k = \sum_{j=0}^{N-1} \text{Re}(x_j) \cos(2\pi k j/N) - \sum_{j=1}^{N-1} \text{Im}(x_j) \sin(2\pi k j/N) \]
\[ + i \left( \sum_{j=0}^{N-1} \text{Im}(x_j) \cos(2\pi k j/N) + \sum_{j=1}^{N-1} \text{Re}(x_j) \sin(2\pi k j/N) \right) \quad (k = 0, \ldots, N-1), \]
the number of real operations of the DFT(N) of a complex-valued sequence is given by
\[ M_N = 2M'_N = N(r - 3) + 4, \quad A_N = 2A'_N + 2(N - 2) = 3N(r - 1) + 4. \]
Compared with other DFT-algorithms, we conclude that our polynomial algorithm works with the same computational complexity as the algorithm in [15] and the split-radix algorithm [3, 14].

6. Fast algorithm for the DCT(3N)

The polynomial representations of the DCT(N), the cos-DFT(N), and the sin-DFT(N) in §§3 and 4 open new possibilities for the derivation of fast algorithms for these transforms for various lengths N by applying the CRT in combination with the factorizations (2.5), (2.6), and (2.7) of Chebyshev polynomials, or in combination with the factorization (2.11). The reductions of polynomials of the form (3.2) or (4.8) modulo Chebyshev polynomials in such algorithms can be performed only by (2.8), (2.9), and (2.10). In order to illustrate these general considerations, we suggest a new polynomial algorithm for the DCT(3N).

We consider the DCT(3N) (N \in \mathbb{N}), i.e., for given
\[ X := \sum_{j=0}^{3N-1} x_j T_j \mod T_{3N}, \]
we have to evaluate
\[ \hat{x}_k = X(z) \mod(z - \cos(\pi(2k + 1)/6N)) \quad (k = 0, \ldots, 3N - 1). \]
By (2.5), \( T_{3N} \) factors as
\[ T_{3N} = 4(T_N - \sqrt{3}/2)T_N(T_N + \sqrt{3}/2), \]
so that \( X \mod T_{3N} \) is completely determined by the residues \( X \mod T_N \) and \( X \mod(T_N \pm \sqrt{3}/2) \). Considering that by (2.5) and (2.8)
\[ T_{2N} = 2T_N^2 - 1, \]
\[ T_{N+j} = 2T_NT_j - T_{N-j}, \]
\[ T_{2N+j} = (4T_N^2 - 1)T_j - 2T_NT_{N-j} \quad (j = 1, \ldots, N - 1), \]
we obtain the following recursive algorithm.
First, we calculate $X \mod T_N$ by

\[
X^{(0)} := \sum_{j=0}^{N-1} x_j^{(0)} T_j = X \mod T_N
\]

with

\[
x_j^{(0)} := \begin{cases} x_0 - x_{2N} & \text{for } j = 0, \\ x_j - x_{2N-j} - x_{2N+j} & \text{for } j = 1, \ldots, N - 1. \end{cases}
\]

Since by (2.3), $T_N$ has the roots $\cos(\pi(2k+1)/2N) = \cos(\pi(2(3k+1)+1)/6N)$ $(k = 0, \ldots, N - 1)$, we obtain

\[
\tilde{x}_{3k+1} = X^{(0)} \mod (T_i - \cos(\pi(2k+1)/2N)) \quad (k = 0, \ldots, N - 1).
\]

Next, we form $X \mod (T_N \pm \sqrt{3}/2)$ as follows:

\[
X^{(1)} := \sum_{j=0}^{N-1} x_j^{(1)} T_j = X \mod (T_N - \sqrt{3}/2)
\]

with

\[
x_j^{(1)} := \begin{cases} x_0 + x_{2N}/2 + x_N\sqrt{3}/2 & \text{for } j = 0, \\ x_j - x_{2N-j} + 2x_{2N+j} + \sqrt{3}(x_{N+j} - x_{3N-j}) & \text{for } j = 1, \ldots, N - 1, \end{cases}
\]

and

\[
X^{(2)} := \sum_{j=0}^{N-1} x_j^{(2)} T_j = X \mod (T_N + \sqrt{3}/2)
\]

with

\[
x_j^{(2)} := \begin{cases} x_0 + x_{2N}/2 - x_N\sqrt{3}/2, & \text{for } j = 0, \\ x_j - x_{2N-j} + 2x_{2N+j} - \sqrt{3}(x_{N+j} - x_{3N-j}) & \text{for } j = 1, \ldots, N - 1. \end{cases}
\]

By (2.6), the zeros of $T_N - \sqrt{3}/2$ and of $T_N + \sqrt{3}/2$ are given by

\[
\{\cos((\pi/6 + 2\pi k)/N) : k = 0, \ldots, N - 1\} = \{\cos(\pi(2(6k') + 1)/6N), \cos(\pi(2(6k' + 5) + 1)/6N) : k = 0, \ldots, M; k' = 0, \ldots, M' \}
\]

and by

\[
\{\cos(5\pi/6 + 2\pi k/ N) : k = 0, \ldots, N - 1\} = \{\cos(\pi(2(6k + 2) + 1)/6N), \cos(\pi(2(6k' + 3) + 1)/6N) : k = 0, \ldots, M; k' = 0, \ldots, M' \},
\]
respectively, where \( M := \lceil N/2 \rceil - 1 \) and \( M' := \lceil N/2 \rceil - 1 \). Hence, we get by (6.2) and (6.3) that

\[
\begin{align*}
\hat{x}_{6k} &= X^{(1)} \mod(T_1 - \cos(\pi(12k + 1)/6N)) \quad (k = 0, \ldots, M), \\
\hat{x}_{6k+5} &= X^{(1)} \mod(T_1 - \cos(\pi(12k + 11)/6N)) \quad (k = 0, \ldots, M'), \\
\hat{x}_{6k+2} &= X^{(2)} \mod(T_1 - \cos(\pi(12k + 5)/6N)) \quad (k = 0, \ldots, M), \\
\hat{x}_{6k+3} &= X^{(2)} \mod(T_1 - \cos(\pi(12k + 7)/6N)) \quad (k = 0, \ldots, M').
\end{align*}
\]

As a result, we have decomposed the DCT(3N) into the DCT(N) given by (6.1) and into the modified DCT(N)'s determined by (6.2) and (6.3), with a total of 2N multiplications and 6N - 2 additions. Obviously, using (2.6) instead of (2.5), these modified DCT's can be handled similarly as the usual DCT. We have only to change the transform factors in the multiplications.

We now combine this idea with the developments of the previous sections. Let \( N = 2^r \) \( (r \geq 2) \). Then by (3.7) and (3.8), we can perform the DCT(3N) with

\[
\begin{align*}
\tilde{M}_{3N} &= 3\tilde{M}_N + 2N = (N/2)(3r + 4), \\
\tilde{A}_{3N} &= 3\tilde{A}_n + 6N - 2 = (N/2)(9r + 6) + 1,
\end{align*}
\]

real operations. Using the decompositions in §5 (see Figure 4), we obtain the following computational complexity for the cos-DFT(3N) and for the sin-DFT(3N):

\[
M^c_{3N} = M^t_{3N} = \sum_{s=0}^{r-2} \tilde{M}_{3 \cdot 2^s} + 2 = (N/4)(3r - 5) + 3, \\
A^c_{3N} = 3N/2 - 1 + 2 \sum_{s=0}^{r-2} 3 \cdot 2^s + \sum_{s=0}^{r-2} \tilde{A}_{3 \cdot 2^s} + 8 = (N/4)(9r - 3) + r + 6, \\
A^t_{3N} = 3N/2 - 1 + 2 \sum_{s=0}^{r-2} (3 \cdot 2^s - 1) + \sum_{s=0}^{r-2} \tilde{A}_{3 \cdot 2^s} + 2 = (N/4)(9r - 3) - r + 2.
\]

Finally, we see that the DFT(3N) requires

\[
M^c_{3N} = (N/2)(3r - 5) + 6, \quad M^t_{3N} = N(3r - 5) + 12
\]

real multiplications and

\[
A^c_{3N} = (N/2)(9r - 3) + 8, \quad A^t_{3N} = N(9r + 3) + 12
\]

real additions. This coincides with the number of real operations for the computation of the DFT(3 \cdot 2^r) \( (r \geq 2) \) by combining the prime factor algorithm, the split-radix algorithm and the Rader algorithm [1, 14].

**Bibliography**


Fachbereich Mathematik, Universität Rostock, Universitätsplatz 1, 0-2500 Rostock, Germany