Number-Theoretic Transforms for Implementers: Butterflies, Twisting, Incompleteness, and Good's Trick

Bo-Yin Yang
Institute of Information Science, Academia Sinica

August 16, 2026

Abstract

We develop (mostly) the radix-2 number-theoretic transform (NTT) and its butterflies, the twisting trick and why it never changes the transform, the freedom to use Cooley–Tukey butterflies in both directions, incomplete[1] NTTs, Good's trick, and the ways all of these combine—culminating in the coefficient-bound bookkeeping that motivates the whole toolkit. This note is intended to help implementers of postquantum cryptography, and is compressed from the author's lecture notes and slides in his Postquantum Cryptography class at National Taiwan University (2020–2025). It may be otherwise trivial for FFT experts who know the DIT–DIF equivalence inside out—except that they tend not to ever encounter incomplete NTTs.

1 Introduction

Throughout, $\mathcal{R}$ is a commutative ring in which 2 is invertible—and, in Section 11 alone, 3 as well; in cryptographic applications $\mathcal{R}$ is usually a prime field $\mathbb{F}_q$ with q odd. We want to multiply in quotient rings such as $\mathcal{R}[x]/\langle x^N-1\rangle$ or $\mathcal{R}[x]/\langle x^N+1\rangle$, and the tool is the Chinese Remainder Theorem (CRT) for polynomials: if we can factor the ring polynomial into coprime pieces, multiplication decomposes into independent multiplications modulo the pieces. The number-theoretic transform (NTT) is nothing more than this factorization applied recursively, organized so that each level costs a linear number of operations [17, 31]. This note develops the basic radix-2 transform and its butterflies, the twisting trick and—importantly—why twisting does not change the result, the freedom to use Cooley–Tukey butterflies in both directions, incomplete NTTs (with Kyber's as the running example), Good's trick, and the ways these variations combine; the development culminates in the coefficient-bound bookkeeping that turns these identities into faster code. We suggest [7, 22, 23] for useful perspectives into this widely-used toolkit[2].

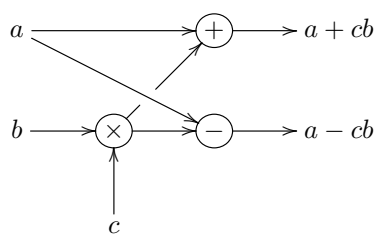
2 One CRT step and its two butterflies

The basic splitting step is the polynomial analogue of $\mathbb{Z}/\langle mn\rangle \cong \mathbb{Z}/\langle m\rangle \times \mathbb{Z}/\langle n\rangle$ for coprime m, n. For any unit $c \in \mathcal{R}^\times$,

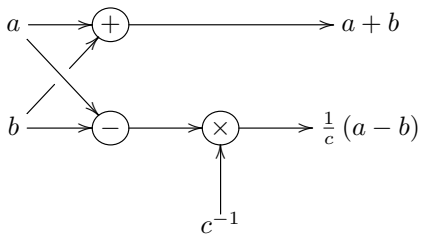
$$\mathcal{R}[x]/\langle x^{2n} - c^2\rangle \;\cong\; \mathcal{R}[x]/\langle x^n - c\rangle \times \mathcal{R}[x]/\langle x^n + c\rangle, \tag{1}$$

[1] To the best of the author's knowledge, he coined this term early in his first semester of class in 2020. Not to be confused with van der Hoeven's *truncated* Fourier transform, which prunes the set of evaluation points to a non-power-of-two count; an incomplete NTT keeps all points but stops the recursion above the linear level.

[2] For a complete mathematical exposition, you are better referred to [23]. This one is for mere mortals.



(a) Cooley–Tukey butterfly: multiply, then add/subtract.



(b) Gentleman–Sande butterfly: add/subtract, then multiply.

Figure 1: The two butterflies of a radix-2 NTT. In the plain NTT the CT butterfly is the forward step and the GS butterfly is the inverse step; Section 6 shows that the assignment can be reversed.

because $\frac{-1}{2c}(x^n - c) + \frac{1}{2c}(x^n + c) = 1$ exhibits the two factors as coprime. Coefficientwise, the forward map takes $a_0 + a_1 x + \cdots + a_{2n-1}x^{2n-1}$ to the pair

$$\Big((a_0 + c\,a_n) + \cdots + (a_{n-1} + c\,a_{2n-1})x^{n-1},\ \ (a_0 - c\,a_n) + \cdots + (a_{n-1} - c\,a_{2n-1})x^{n-1} \Big), \tag{2}$$

and the inverse map reassembles $(f(x), g(x))$ as

$$f(x) \cdot \frac{1}{2c}(x^n + c) + g(x) \cdot \frac{-1}{2c}(x^n - c) \ = \ \frac{f(x) + g(x)}{2} + \frac{f(x) - g(x)}{2c}\, x^n. \tag{3}$$

Each direction costs n additions, n subtractions, and n multiplications by a fixed constant (c forward, c^{-1} backward; the halvings can be deferred and merged into one final scaling). In exchange, one multiplication modulo $x^{2n} - c^2$ becomes two independent half-size multiplications—the divide-and-conquer engine behind everything that follows.

The two coefficientwise maps are the two classical *butterflies*, shown in Figure 1: the forward map (2) multiplies *first* and then adds and subtracts (the *Cooley–Tukey*, or CT, butterfly [17]), while the inverse map (3) adds and subtracts *first* and multiplies afterwards (the *Gentleman–Sande*, or GS, butterfly [20]). Keep this shape distinction in mind: it is the whole point of Sections 4–6.

3 The radix-2 NTT

Iterating (1) gives the radix-2 cyclic NTT. Suppose $\zeta \in \mathcal{R}$ satisfies $\zeta^{2^{k-1}} = -1$, i.e., ζ is a primitive 2^k-th root of unity. Starting from $\mathcal{R}[x]/\langle x^{2^k} - 1\rangle$ and splitting ℓ times, the factor visited t-th (in the natural in-place order) is

$$\mathcal{R}[x]/\big\langle x^{2^{k-\ell}} - \zeta^{2^{k-\ell}\,\mathrm{brv}_\ell(t)} \big\rangle, \qquad t = 0, \ldots, 2^\ell - 1, \tag{4}$$

where $\mathrm{brv}_\ell(t)$ denotes the reversal of t written as ℓ bits. After k levels every factor is linear:

$$\mathcal{R}[x]/\langle x^{2^k} - 1\rangle \ \cong \ \prod_{t=0}^{2^k-1} \mathcal{R}[x]/\big\langle x - \zeta^{\mathrm{brv}_k(t)} \big\rangle \ \cong \ \prod_{t=0}^{2^k-1} \mathcal{R},$$

and coordinate t of the transform is simply the evaluation $f(\zeta^{\mathrm{brv}_k(t)})$. The *negacyclic* transform for $\mathcal{R}[x]/\langle x^{2^k} + 1\rangle$ works identically, now requiring $\zeta^{2^k} = -1$ (a primitive 2^{k+1}-th root of unity); after ℓ levels the factors are $x^{2^{k-\ell}} - \zeta^{\mathrm{brv}_{k+1}(t)}$ for $t = 2^\ell, \ldots, 2^{\ell+1} - 1$, and the leaves are $x - \zeta^{\mathrm{brv}_{k+1}(2^k+t)}$, the odd powers of ζ.

To multiply $f, g \in \mathcal{R}[x]/\langle x^{2^k} - 1\rangle$: map both into $\mathcal{R}^{2^k}$ (k levels of butterflies, each level $3 \cdot 2^{k-1}$ operations, so $O(k\,2^k)$ in total), multiply the two vectors coordinatewise ($O(2^k)$), and apply the inverse mapping. Since coordinate t holds $f(\zeta^{\mathrm{brv}_k(t)})\, g(\zeta^{\mathrm{brv}_k(t)}) = (fg)(\zeta^{\mathrm{brv}_k(t)})$, the inverse transform returns $fg \bmod (x^{2^k} - 1)$.

Example 3.1 (a complete negacyclic NTT)**.** In $\mathbb{Z}_{17}[x]/\langle x^4 + 1\rangle$ we have $2^4 = -1$, and

$$x^4 + 1 = (x^2 - 4)(x^2 + 4) = (x - 2)(x + 2)(x - 8)(x + 8).$$

To multiply $f(x) = 2x^3 + 7x^2 + x + 8$ and $g(x) = 2x^3 + 4x + 8$:

$$\begin{aligned} f(x) &\to (9x + 36,\ -7x - 20) \to (54,\ 18,\ -76,\ 36) = (3, 1, 9, 2),\\ g(x) &\to (12x + 8,\ -4x + 8) \to (32,\ -16,\ -24,\ 40) = (-2, 1, -7, 6). \end{aligned}$$

Coordinatewise multiplication gives $(-6, 1, -63, 12) = (-6, 1, 5, -5)$, and the inverse transform reassembles

$$\begin{aligned} (-6, 1, 5, -5) &\to \tfrac{1}{2}\big(-5 + \tfrac{-7}{2}x,\ \ \tfrac{10}{8}x\big) = \tfrac{1}{2}\big(-5 + 5x,\ \ -3x\big)\\ &\to \tfrac{1}{4}\Big[(-5 + 2x) + \tfrac{-5+8x}{4}\,x^2\Big] = \tfrac{1}{4}\big[2x^3 + 3x^2 + 2x - 5\big] = 9x^3 + 5x^2 + 9x + 3, \end{aligned}$$

which is indeed $f(x)g(x) \bmod (x^4 + 1)$. See below for the decomposition and the ensuing ordering.

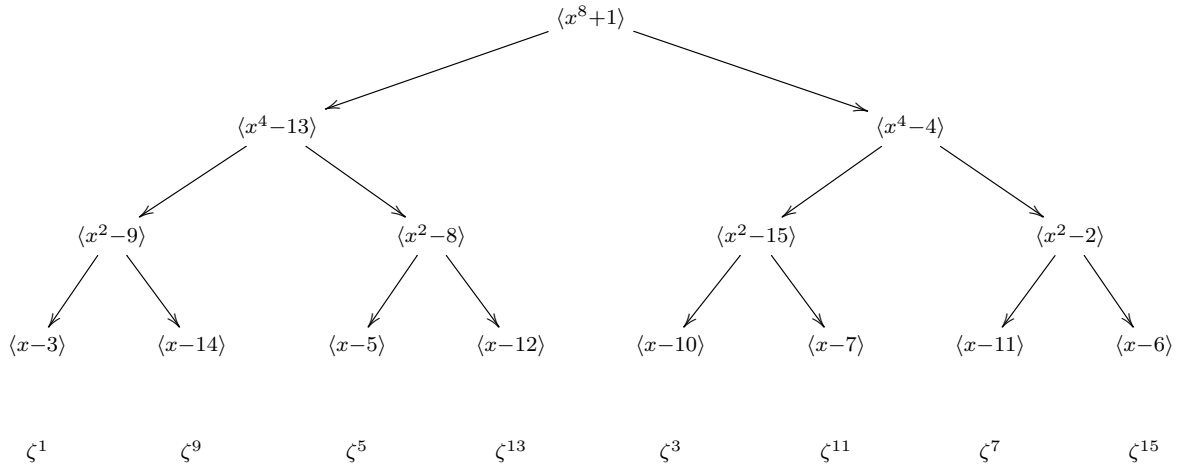


Figure 2: Complete splitting of $\mathbb{F}_{17}[x]/\langle x^8+1\rangle$, with $\zeta = 3$ a primitive 16th root of unity modulo 17, so that $\zeta^8 = -1$. A node $\langle x^m - \zeta^s\rangle$ has children $\langle x^{m/2} - \zeta^{s/2}\rangle$ and $\langle x^{m/2} + \zeta^{s/2}\rangle = \langle x^{m/2} - \zeta^{s/2+8}\rangle$; one layer of Cooley–Tukey butterflies with twiddle $\zeta^{s/2}$ realizes each level. The last row gives the leaf exponents: leaf t is $\langle x - \zeta^{\mathrm{brv}_4(8+t)}\rangle$, and $\mathrm{brv}_4(8+t) = 2\,\mathrm{brv}_3(t)+1$, so the odd powers of ζ arrive in bit-reversed order. A corresponding diagram for a "normal" NTT is Figure 4 in the appendix.

A remark on ordering: Implemented in place on naturally ordered coefficients, the transform produces its outputs in the bit-reversed indexing of (4). It is traditional to "bit-reverse" the inputs so that outputs come out in natural order, but for polynomial multiplication the output order is irrelevant: the pointwise stage is order-agnostic, and the index permutation can be absorbed into

the inverse transform. No explicit reversal pass is ever needed. For mixed-radix transforms the same bookkeeping is handled by a digit-reversal function: define recursively $R_p(k) = k$ and

$$R_{p_1,\dots,p_{n-1},p_n}(k) = \Big(k - \Big\lfloor \frac{k}{p_n} \Big\rfloor p_n\Big) \cdot \prod_{i=1}^{n} p_i + R_{p_1,\dots,p_{n-1}}\Big(\Big\lfloor \frac{k}{p_n} \Big\rfloor\Big);$$

then, e.g., a 270-NTT applying one radix-2, three radix-3, and one radix-5 stage outputs in the order $[R_{2,3,3,3,5}(0), \dots, R_{2,3,3,3,5}(269)]$.

4 The Twisting Trick

Splitting is one way to simplify a modulus; *twisting* is the other. If $\xi \in \mathcal{R}^\times$ satisfies $\xi^N = c$, then the substitution $x = \xi y$,

$$\begin{aligned} & & \mathcal{R}[x]/\langle x^N - c\rangle \;&\longrightarrow\; \mathcal{R}[y]/\langle y^N - 1\rangle, \\ f(x) \;\mapsto\; f(\xi y): \quad & (a_0, a_1, a_2, \dots, a_{N-1}) \;&\mapsto\; (a_0,\; a_1\xi,\; a_2\xi^2, \dots,\; a_{N-1}\xi^{N-1}), \end{aligned} \tag{5}$$

is a ring isomorphism: it transforms arithmetic modulo $x^N - c$ into arithmetic modulo $y^N - 1$ at the cost of $N - 1$ multiplications per polynomial (against $N/2$ multiplications for one splitting layer of the same ring). Both the twisted and the untwisted ring eventually split into copies of $\mathcal{R}$, and—as Section 5 makes precise—the resulting transforms are one-to-one identical, so twisting is never a question of correctness, only of operation scheduling. Why pay nearly double the multiplications, then? One important algorithmic reason: with lazy (delayed) modular reductions, array entries grow with each addition, and a twist—whose multiplications are typically performed with reduction—resets every coefficient to a reduced range just before entries would go out of bounds.

Used *inside* the recursion, twisting gives the *twisted FFT* [7]. The base case: if $\zeta^n = -1$, then

$$\mathcal{R}[x]/\langle x^{2n} - 1\rangle \cong \mathcal{R}[x]/\langle x^n - 1\rangle \times \mathcal{R}[x]/\langle x^n + 1\rangle, \qquad \mathcal{R}[x]/\langle x^n + 1\rangle \underset{x\leftrightarrow\zeta y}{\cong} \mathcal{R}[y]/\langle y^n - 1\rangle,$$

since $x^n + 1 \leftrightarrow (\zeta y)^n + 1 = -y^n + 1$. Both children are now copies of the *same* problem at half size, so the recursion is uniform: with ζ a 2^{k-1}-th root of -1,

$$\begin{aligned} \mathcal{R}[x]/\langle x^{2^k} - 1\rangle &\cong \mathcal{R}[x]/\langle x^{2^{k-1}} - 1\rangle \times \mathcal{R}[x]/\langle x^{2^{k-1}} - 1\rangle \quad \{\text{twist by } \zeta\} \\ &\cong \textstyle\prod^2\Big(\mathcal{R}[x]/\langle x^{2^{k-2}} - 1\rangle \times \mathcal{R}[x]/\langle x^{2^{k-2}} - 1\rangle\Big) \quad \{\text{twist by } \zeta^2\} \\ &\cong \cdots \cong \textstyle\prod^{2^k}\mathcal{R}. \end{aligned}$$

Coefficientwise, one twisted level maps

$$\begin{pmatrix} a_0 & \cdots & a_{n-1} \\ a_n & \cdots & a_{2n-1} \end{pmatrix} \longmapsto \begin{pmatrix} (a_0 + a_n) & (a_1 + a_{n+1}) & \cdots & (a_{n-1} + a_{2n-1}) \\ (a_0 - a_n) & (a_1 - a_{n+1})\,\zeta & \cdots & (a_{n-1} - a_{2n-1})\,\zeta^{n-1} \end{pmatrix},$$

i.e., subtract first, multiply afterwards: *the twisted NTT is built from Gentleman–Sande butterflies*, with the multiplications by ζ^i being the twist. Its inverse untwists first and then merges,

$$\begin{pmatrix} b_0 & \cdots & b_{n-1} \\ c_0 & \cdots & c_{n-1} \end{pmatrix} \longmapsto \frac{1}{2}\begin{pmatrix} (b_0 + c_0) & (b_1 + c_1\zeta^{-1}) & \cdots & (b_{n-1} + c_{n-1}\zeta^{-(n-1)}) \\ (b_0 - c_0) & (b_1 - c_1\zeta^{-1}) & \cdots & (b_{n-1} - c_{n-1}\zeta^{-(n-1)}) \end{pmatrix},$$

which multiplies first: a Cooley–Tukey butterfly with twiddle ζ^{-i} (and a deferred halving).

Example 4.1 (a complete twisted NTT)**.** In $\mathbb{Z}_{17}[x]/\langle x^8-1\rangle$, again $2^4=-1$. For $f(x)=1+2x+8x^2+2x^3+5x^4+6x^5+5x^6+x^7$:

$$f \xrightarrow{\sqrt[4]{-1}=2} (6{+}8x{+}13x^2{+}3x^3,\ -4{-}8x{+}12x^2{+}8x^3) \xrightarrow{\sqrt[2]{-1}=4} (19{+}11x,\ -7{+}20x,\ 8,\ -16{-}64x)$$

$$\xrightarrow{\sqrt[1]{-1}=-1} (30,\ 8,\ 13,\ -27,\ 8,\ 8,\ -80,\ 48) \equiv (13,\ 8,\ 13,\ 7,\ 8,\ 8,\ 5,\ 14) \pmod{17}.$$

One checks directly that output t equals $f(2^{\mathrm{brv}_3(t)})$: for instance output 4 is $f(2)\equiv 297\equiv 8$ and output 3 is $f(2^6)=f(-4)=-27\equiv 7$. These are exactly the values, in exactly the order, that the untwisted recursion of Section 3 produces—a concrete instance of Lemma 5.1 below.

For lazy-reduction planning it is instructive to compare one level of the plain and the twisted map: the operation counts per level are essentially equal, but the *placement* of the multiplications differs. In the plain CT level, every output coefficient is an addition or subtraction whose operand is a fresh product; in the twisted level only half the outputs are touched by a multiplication.[3] Since additions performed after multiplications are what push lazily reduced coefficients toward overflow, this proportion—1 for a plain level, $\frac{1}{2}$ for a twisted one, reversed for the corresponding inverses—is a real scheduling resource, and twisting is the knob that adjusts it; Section 10 turns the bookkeeping into concrete word-size budgets.

5 Why the Plain and the Twisted NTT Compute the Same Thing

Everything above yields ring isomorphisms onto products of smaller rings, so any of these transforms *can* be used for multiplication. The stronger and more useful statement is that they all compute the *same* map—at every node of the recursion, not merely at the end: the same residues, and under the canonical choice of constants even in the same order. This is what licenses mixing the variants freely, which Sections 6 and 9 exploit.

Lemma 5.1 (invariance at every node)**.** *Let $c\in\mathcal{R}^\times$, and let 2 and every constant drawn below be units of $\mathcal{R}$; nothing here requires x^N-c to split into linear factors. Consider any tower of maps that starts at $\mathcal{R}[x]/\langle x^N-c\rangle$ and is built from the two moves*

(S) split: $\mathcal{R}[z]/\langle z^{2m}-e^2\rangle \to \mathcal{R}[z]/\langle z^m-e\rangle \times \mathcal{R}[z]/\langle z^m+e\rangle$;

(T) twist: $\mathcal{R}[z]/\langle z^m-d\rangle \to \mathcal{R}[w]/\langle w^m-d\xi^{-m}\rangle$, $g(z)\mapsto g(\xi w)$, *for a unit ξ,*

not *necessarily continued down to linear factors. To a node ν with ring $\mathcal{R}[z]/\langle z^m-d\rangle$, reached through twists whose constants have product Ξ_ν, associate the polynomial $x^m-d\,\Xi_\nu^m$. Then:*

1. *$x^m-d\,\Xi_\nu^m$ divides x^N-c, and the composite map from the root to ν is $f\mapsto f(\Xi_\nu z) \bmod \langle z^m-d\rangle$; under the change of variable $x=\Xi_\nu z$ this is exactly reduction modulo the associated polynomial, $f\mapsto f \bmod \langle x^m-d\,\Xi_\nu^m\rangle$.*

2. *The associated polynomial is invariant under twisting: move* (T) *at ν changes the presentation, $(d,\Xi_\nu)\mapsto(d\xi^{-m},\,\Xi_\nu\xi)$, but not the product $d\,\Xi_\nu^m$; and move* (S) *at ν, with root e of the constant chosen, produces children with associated polynomials $x^m-e\,\Xi_\nu^m$ and $x^m+e\,\Xi_\nu^m$.*

[3] The split-radix FFT [38, 19], which twists only the "inconvenient" quarters of $\mathcal{R}[x]/\langle x^{4n}-1\rangle$, pushes the proportion further down to $\frac{1}{4}$; we do not pursue it here.

Consequently, two towers that choose the same roots at corresponding splits—automatic when all constants are drawn as the canonical powers of one fixed root ζ—carry the identical *tree of associated polynomials, at every depth. Truncated to any common shape, they compute the same decomposition of $\mathcal{R}[x]/\langle x^N - c\rangle$ node by node, the twists amounting to no more than the change of variable $x = \Xi_\nu z$ within each component.*

Proof. Maintain the invariant that a node $\mathcal{R}[z]/\langle z^m - d\rangle$ reached through accumulated substitution $x = \Xi z$ (with Ξ the product of the twist constants so far, initially $\Xi = 1$) stores the residue $f(\Xi z) \bmod \langle z^m - d\rangle$. This holds at the root. A split preserves it, replacing one residue by its two reductions; a twist by ξ turns $g(z) = f(\Xi z) \bmod \langle z^m - d\rangle$ into $g(\xi w) = f(\Xi\xi w) \bmod \langle w^m - d\xi^{-m}\rangle$ and updates $\Xi \leftarrow \Xi\xi$. For claim (2): under the twist, $(d\xi^{-m})(\Xi\xi)^m = d\,\Xi^m$, so the associated polynomial is unchanged; under a split at $\mathcal{R}[z]/\langle z^{2m} - e^2\rangle$ the children $\langle z^m \mp e\rangle$ have associated polynomials $x^m \mp e\,\Xi^m$, whose product is the parent's $x^{2m} - e^2\Xi^{2m}$. Divisibility in claim (1) follows by induction down the tree, and the identification of the composite map with reduction modulo the associated polynomial is the invariant read through $x = \Xi z$. For the final claim, induct on depth: at corresponding nodes with equal associated polynomials, a twist changes neither the polynomial nor the underlying decomposition, and splits with the same chosen root produce children with equal associated polynomials in the same order. □

Corollary 5.2 (complete towers)**.** *Suppose $\mathcal{R}$ contains a primitive 2^k-th root of unity ζ (so that $x^{2^k} - 1$ splits into the 2^k distinct linear factors $x - \zeta^t$); fix such a ζ and draw all splitting and twist constants as the canonical powers of ζ (the children of the node with constant ζ^{2s} use $\pm\zeta^s$, and the twist of $x^m + 1$ uses the corresponding root of -1). Then every tower over $\mathcal{R}[x]/\langle x^{2^k} - 1\rangle$ continued down to linear factors—the plain tower, the fully twisted tower, and every mixture—realizes the* identical *map*

$$\Phi :\ \mathcal{R}[x]/\langle x^{2^k} - 1\rangle \to \mathcal{R}^{2^k}, \qquad \Phi(f)_t = f\big(\zeta^{\mathrm{brv}_k(t)}\big) :$$

same values, same order. In particular the pointwise stage cannot tell which tower produced its inputs, and the single inverse map Φ^{-1} may be computed by inverting any *of the towers—the forward transform of one variant may be paired with the inverse of another.*

Proof. At a linear leaf the associated polynomial is $x - \Xi_\nu r$ and the stored value is $f(\Xi_\nu r)$, by the lemma. The leaf polynomials are the 2^k distinct linear factors of $x^{2^k} - 1$, in the order $x - \zeta^{\mathrm{brv}_k(t)}$ of the plain tower; by the lemma this order is shared by every tower with the canonical constants. □

Corollary 5.3 (incomplete towers)**.** *Truncate two towers with canonical constants at the same shape, with leaves of degree d. Both decompose $\mathcal{R}[x]/\langle x^N - c\rangle$ modulo the same list of factors $x^d - c_t$; the plain tower stores the residues $f \bmod \langle x^d - c_t\rangle$ themselves, while a twisted tower stores the same residues in twisted coordinates, as $f(\Xi_t z) \bmod \langle z^d - c_t\Xi_t^{-d}\rangle$. In particular, when the twists act on a substituted variable $y = x^d$ in which the tower is complete—so the retained variable x is never twisted and y is eliminated at the leaves—no twisted coordinates remain, and the plain and twisted incomplete transforms are* equal as maps. *This is exactly the situation of Section 9.*

Proof. The first statement is the lemma read at the truncated leaves. For the second, write $A = \mathcal{R}[x]/\langle x^{dn} - c\rangle$ and $B = \mathcal{R}[y]/\langle y^n - c\rangle$ with $y = x^d$, so that $A = B \oplus Bx \oplus \cdots \oplus Bx^{d-1}$ as a B-module. Any tower on B extends componentwise to A, and the lemma applied to B shows the plain and twisted y-towers are the identical map on B, hence on each component of A. At a linear y-leaf $\langle y - \rho\rangle$ the components reassemble to the residue ring $\mathcal{R}[x]/\langle x^d - \rho\rangle$, with no twist left on x. □

If constants are *not* drawn consistently (say one implementation picks ξ where another effectively picks $-\xi$), the two outputs differ by a known permutation; this permutation is harmless for multiplication and can be folded into twiddle indexing. Example 4.1 shows the coordinate-by-coordinate agreement numerically.

6 Cooley–Tukey butterflies in both directions

Collecting the coefficient maps of Sections 2–4, the four transform variants use butterflies as follows (twiddles shown for one pair at distance n; scalings deferred):

tower	forward	inverse
plain	CT, twiddle ζ^s: $(a, b) \mapsto (a + \zeta^s b,\ a - \zeta^s b)$	GS, twiddle ζ^{-s}: $(u, v) \mapsto (u + v,\ (u - v)\zeta^{-s})$
twisted	GS, twiddle ζ^i: $(a, b) \mapsto (a + b,\ (a - b)\zeta^i)$	CT, twiddle ζ^{-i}: $(u, v) \mapsto (u + \zeta^{-i} v,\ u - \zeta^{-i} v)$

By Corollary 5.2 the four cells are freely interchangeable: since the plain and twisted towers compute the same map Φ, a multiplication pipeline may run the forward transform down the plain tower and invert along the twisted tower, or vice versa. Concretely, this gives implementations that use *Cooley–Tukey butterflies in both directions* (plain forward, twisted inverse) or Gentleman–Sande butterflies in both directions (twisted forward, plain inverse). The freedom is not universally exercised: the CT forward/GS inverse pairing remains the default in production code, perhaps mostly descended from Seiler's [36], including the `mlkem-native` and `mldsa-native` lines, while CT-both-ways appears where the bound profile of Section 10 favors it, as in Neon NTT [6]; Section 9 extends the same pairing to incomplete NTTs such as Kyber's. (In classical FFT terminology the plain recursion is the decimation-in-time network and the twisted recursion is decimation-in-frequency; the equality of the two transforms is exactly the classical DIT/DIF correspondence [20, 31].)

Which combination to choose is an engineering question with three main inputs. First, as noted in Section 3, each cell either consumes natural order and produce bit-reversed order or do the reverse, and the permutation dissolves into indexing. Second, coefficient ranges: per CT level, every output coefficients is an array element added/subtracted by a product (the range increases additively); per GS level, $\frac{1}{2}$ of the coefficients are products and $\frac{1}{2}$ are sums (the range exponentiates); choosing butterfly directions is changing the output ranges. Third, twiddle economics, which becomes visible at odd radixes: both butterflies exist for any radix, e.g. radix 3 (with ω a primitive cube root of unity),

$$\text{CT: } \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \mapsto \begin{pmatrix} a_0 + a_1 c + a_2 c^2 \\ a_0 + a_1 \omega c + a_2 \omega^2 c^2 \\ a_0 + a_1 \omega^2 c + a_2 \omega c^2 \end{pmatrix}, \qquad \text{GS: } \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \mapsto \begin{pmatrix} a_0 + a_1 + a_2 \\ (a_0 + a_1 \omega^1 + a_2 \omega^2)\, c^{-1} \\ (a_0 + a_1 \omega^2 + a_2 \omega^1)\, c^{-2} \end{pmatrix},$$

where the CT form needs the six multiplicands $c, c^2, \omega c, \omega^2 c^2, \omega^2 c, \omega c^2$ while the GS form makes do with the four $\omega, \omega^2, c^{-1}, c^{-2}$—fewer constants to keep in registers or tables. (In either direction a radix-3 step on blocks of size n costs $4n$ multiplications by these constants and $6n$ additions and subtractions, exploiting $1 + \omega + \omega^2 = 0$: two multiplications to form $a_1 c$ and $a_2 c^2$, and two more inside the 3-point transform proper.)

6.1 The same equivalence, read through orthogonality

The above talks CRT-theory with terms like rings and ideals (and towers). It is worth recording the four-line classical Fourier-analytic derivation as well, as that clarifies the rôle of the twist.

$$\sum_{j=0}^{n-1} \zeta_n^{j\ell} \;=\; n\,\delta_{0\ell}, \qquad -n < \ell < n,$$

if $\zeta_n \in \mathcal{R}$ is a principal n-th root of unity, as for $\ell \not\equiv 0$ the geometric sum $(\zeta_n^{n\ell}-1)/(\zeta_n^{\ell}-1) = 0$. Now if $b_k = \sum_j a_j \zeta_n^{jk}$, then an inversion formula can be shown by orthogonality in a double sum:

$$\frac{1}{n}\sum_{k=0}^{n-1} b_k\,\zeta_n^{-kj} = \frac{1}{n}\sum_{k=0}^{n-1} \zeta_n^{-kj} \sum_{\ell=0}^{n-1} a_\ell \zeta_n^{\ell k} \;=\; \frac{1}{n}\sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1} a_\ell \zeta_n^{k(\ell-j)} \;=\; \frac{1}{n}\sum_{\ell=0}^{n-1} n\delta_{\ell-j,0} a_\ell \;=\; a_j.$$

Now the negacyclic case, where $\zeta = \zeta_{2n}$ satisfies $\zeta^n = -1$ and the transform evaluates at the *odd* powers of ζ. The trick is to peel the odd part of the exponent off the index being summed:

$$b_k \;=\; \sum_{j=0}^{n-1} a_j\,\zeta^{\,j(2k+1)} \;=\; \sum_{j=0}^{n-1} \big(a_j\,\zeta^{\,j}\big)\,\zeta_n^{jk}, \qquad \zeta_n := \zeta^2.$$

That *is* the twisting trick of Section 4, and the negacyclic transform is the cyclic transform of the pre-twisted sequence $(a_j\zeta^{\,j})_j$. Applying cyclic inversion to it and dividing the twist back out,

$$a_j\,\zeta^{\,j} \;=\; \frac{1}{n}\sum_{k=0}^{n-1} b_k\,\zeta_n^{-kj}, \qquad \text{that is} \qquad a_j \;=\; \frac{1}{n}\sum_{k=0}^{n-1} b_k\,\zeta^{-(2k+1)j}, \text{ or } \quad \frac{1}{n\zeta^{\,j}}\sum_{k=0}^{n-1} b_k\,\zeta_n^{-kj}.$$

Caution[4]: So the inverse negacyclic transform can be viewed in two ways. The latter form depicts a straight-up cyclic transform with powers of ζ_n^{-1} then a scaling-twist by $\frac{1}{n\zeta^j}$. The other form is more subtle as the two indices switched roles: while in the forward map the factor $(2k+1)$ rides on the free index and j on the summation index, in the inverse it is the reverse. In matrix terms the negacyclic matrix $F_{kj} = \zeta^{\,j(2k+1)}$ is *not* symmetric—unlike the cyclic matrix ζ_n^{jk}, which is—and what the display above says is $F^{-1} = \frac{1}{n}(F')^{\mathsf{T}}$ with F' the same matrix built from ζ^{-1}. Transposing a linear network reverses every arrow in its dataflow graph [12], and reversing a Cooley–Tukey network yields a Gentleman–Sande one. This explains the "normal" pairing of CT forward with GS inverse.

So Cooley–Tukey in the negacyclic iNTT is simply an NTT from bitreversed positions scaled by $n^{-1}\zeta^{-j}$. The exponential-versus-additive bound profiles of Section 10 are the practical reason to want it. *However, if you are interested in formal verification and proofs, the proof of inverse twisted NTT follows the proof as the forward NTT except for the bitreversal and the final scaling.*

7 Incomplete NTT

A *complete* NTT splits the ring polynomial all the way into linear factors. Whether this is possible is a question of root supply. For $x^{2^k}-1$ over $\mathbb{F}_q$ we need a primitive 2^k-th root of unity; since $\mathbb{F}_q^\times$ is cyclic of order $q-1$, one exists if and only if $2^k \mid (q-1)$. Most often the ring polynomial is a cyclotomic polynomial $\Phi_n(x)$ (the monic factor of x^n-1 dividing no earlier x^m-1), and Φ_n splits completely over $\mathbb{F}_q$ if and only if x^n-1 does, i.e., iff $n \mid (q-1)$. Examples of complete NTTs:

[4]Also note: This derivation is not the DIT/DIF equivalence. That equivalence concerns two *factorizations of one matrix*—split the input index and you get decimation in time, split the output index and you get decimation in frequency—and it is what Lemma 5.1 and Corollary 5.2 establish. What orthogonality gives is the complementary fact about *one matrix and its inverse*. Both are needed for the freedom claimed in this section.

- NewHope [1], with $\mathbb{F}_{12289}[x]/\langle x^{1024}+1\rangle$: the ring polynomial is $\Phi_{2048}(x)$ and $2048 \mid 12288$.
- Dilithium [18], with $\mathbb{F}_q[x]/\langle x^{256}+1\rangle$, $q = 2^{23} - 2^{13} + 1$: the ring polynomial is $\Phi_{512}(x)$ and $512 \mid q-1$.
- First-round Kyber [11], with $\mathbb{F}_{7681}[x]/\langle x^{256}+1\rangle$: again the ring polynomial is $\Phi_{512}(x)$, and $q - 1 = 7680 = 15 \cdot 512$, so $512 \mid q-1$ and the transform runs all the way down. This is the cleanest small example to keep in mind—the same ring as Kyber's today, but with the modulus chosen so that the last layer still exists.

When the supply runs short, the ring polynomial still splits—just not down to linear factors—and we speak of an *incomplete NTT*: stop the recursion at degree $d > 1$, and multiply the small residues by schoolbook (or Karatsuba) at the leaves.

- Kyber from the second round onward [4]—the modulus was reduced from 7681 to 3329 between rounds—with $\mathbb{F}_{3329}[x]/\langle x^{256}+1\rangle$ and the same $x^{256}+1 = \Phi_{512}(x)$: now $256 \mid 3328$ but $512 \nmid 3328$, so the splitting stops one level early,

$$\mathbb{F}_{3329}[x]/\langle x^{256}+1\rangle \cong \bigoplus_{j=0}^{127} \frac{\mathbb{F}_{3329}[x]}{\langle x^2 - \omega_{256}^{2j+1}\rangle},$$

  with ω_{256} a primitive 256th root of unity.
- NTTRU [28], $\mathbb{F}_{7681}[x]/\langle x^{768} - x^{384} + 1\rangle$ with $x^{768} - x^{384} + 1 = \Phi_{2304}(x)$: here $768 \mid 7680$ but $2304 \nmid 7680$, and the ring splits as

$$\bigoplus_{j=0}^{127} \frac{\mathbb{F}_{7681}[x]}{\langle x^3 - \beta_j\rangle} \;\oplus\; \bigoplus_{j=0}^{127} \frac{\mathbb{F}_{7681}[x]}{\langle x^3 - \beta'_j\rangle},$$

  where the β_j and β'_j are the 128th roots of 685 and -684, which are in turn the primitive sixth roots of unity modulo 7681. *Note the initial split is not Cooley–Tukey (cf. Section 11).*

Incompleteness is not merely a fallback: even when completion is possible, stopping early can be the better trade. Consider degree-2 leaves. The base multiplication

$$(a + bx)(c + dx) \equiv (ac + bd\,\omega_j) + (ad + bc)\,x \pmod{x^2 - \omega_j}$$

costs 5 multiplications and 2 additions. Completing the last level instead would spend three 2-point transforms (two forward, one inverse; each 1 multiplication and 2 additions) plus two pointwise multiplications: 5 multiplications and 6 additions. Same multiplication count, strictly more additions—so the last splitting level never pays for itself at degree 2. At degree 3 the effect is stronger: schoolbook

$$\begin{aligned}(a_0 + a_1x + a_2x^2)&(b_0 + b_1x + b_2x^2)\\ &\equiv \big(a_0b_0 + \omega_j(a_1b_2 + a_2b_1)\big) + \big(a_0b_1 + a_1b_0 + \omega_j a_2b_2\big)x + \big(a_0b_2 + a_1b_1 + a_2b_0\big)x^2\end{aligned}$$

modulo $x^3 - \omega_j$ takes 11 multiplications and 6 additions, while completing the split would spend three 3-point transforms (two forward, one inverse, each 4 multiplications and 6 additions by Section 6) plus three pointwise products: 15 multiplications and 18 additions. Here the schoolbook wins on both counts, and decisively. Incomplete NTTs thus serve two purposes at once: they rescue NTT multiplication in fields without enough roots, and they shave additions even in fields with plenty.

8 The Kyber NTT: a Complete NTT in $y = x^2$

The incomplete NTT of Kyber's later rounds [4] deserves a closer look, both because of its ubiquity and because the right viewpoint on it unlocks everything in Sections 4–6 for incomplete transforms. Rather than "a 256-point NTT stopped one level early," regard it as a complete 128-point NTT in disguise: set $x^2 = y$ and present the ring as

$$\frac{\mathbb{F}_{3329}[x]}{\langle x^{256}+1\rangle} \;=\; \frac{\mathbb{F}_{3329}[x,y]}{\langle y^{128}+1,\; x^2-y\rangle}, \qquad f=\sum_{i=0}^{255} a_i x^i = \sum_{i=0}^{127} (a_{2i}+a_{2i+1}x)\, y^i.$$

In the variable y the ring polynomial $y^{128}+1$ is *complete* for Kyber's root supply: with $\omega = \omega_{256}$ a primitive 256th root of unity, $\omega^{128} = -1$, and the plain recursion of Section 3 runs in y with no obstruction,

$$\langle y^{128}+1\rangle \;\to\; \langle y^{64}-\omega^{64}\rangle \oplus \langle y^{64}+\omega^{64}\rangle \;\to\; \langle y^{32}-\omega^{32}\rangle \oplus \langle y^{32}+\omega^{32}\rangle \oplus \langle y^{32}-\omega^{96}\rangle \oplus \langle y^{32}+\omega^{96}\rangle \to\; \cdots \;\to \bigoplus_{t=0}^{127} \big\langle y-\omega^{\mathrm{brv}_8(128+t)}\big\rangle,$$

each arrow one layer of Cooley–Tukey butterflies with the displayed square roots as twiddles. Substituting $y = x^2$ back into the leaves recovers exactly the decomposition of Section 7,

$$\frac{\mathbb{F}_{3329}[x]}{\langle x^{256}+1\rangle} \;\cong\; \bigoplus_{t=0}^{127} \frac{\mathbb{F}_{3329}[x]}{\big\langle x^2-\omega^{\mathrm{brv}_8(128+t)}\big\rangle},$$

the odd powers ω^{2j+1} arriving in bit-reversed order; the Kyber specification [4] writes the exponent as $2\,\mathrm{brv}_7(i)+1$, and indeed $\mathrm{brv}_8(128+t) = 2\,\mathrm{brv}_7(t)+1$.

Because the transform acts on the y-digits and never touches x, every butterfly operates componentwise on the coefficient pair (a_{2i}, a_{2i+1}). The Kyber NTT is therefore *two size*-128 *negacyclic NTTs running side by side*—one on the even-indexed and one on the odd-indexed coefficients—sharing all twiddles and all control flow; the two strands only meet again in the base multiplications, the 128 products modulo $x^2 - \omega^{\mathrm{brv}_8(128+t)}$. This is how implementations lay the transform out in memory and registers, and it is the viewpoint under which the twisting machinery applies to Kyber, as Section 9 will show.

9 Incomplete Twisted NTT

So, how does the twisting trick of Section 4 interact with incompleteness? A first attempt fails instructively. An incomplete tower leaves residues modulo $x^d - c_t$, and twisting that variable—$x = \xi y$ with $\xi^d = c_t$, aiming for $y^d - 1$—requires a d-th root of each leaf constant. For Kyber this would be a primitive 512th root of unity modulo 3329: precisely the element whose absence made the NTT incomplete in the first place. You can't twist like that. The conclusion, however, is not that the twisted NTT is unavailable; it is that the inert variable x is the wrong variable to twist. An incomplete NTT is a *complete* NTT in the substituted variable $y = x^d$ (Section 8), and the twisting trick applies verbatim there.

Concretely for Kyber, return to the presentation of Section 8: the ring is $\mathbb{F}_{3329}[x,y]/\langle y^{128}+1,\; x^2-y\rangle$, the transform acts on the 128 coefficient pairs and never touches x, and as a modulus in y, $y^{128}+1$ is complete for the available roots. So the whole of Section 4 runs in the variable y. Twist first, via $y = \omega_{256}\, z_0$; then alternate multiplication-free splittings with twists by $\omega_{128} = \omega_{256}^2$,

$\omega_{64} = \omega_{256}^4$, …, introducing a fresh variable z_ℓ at each level (the untwisted branch is relabeled by the identity substitution $z_{\ell-1} = z_\ell$):

$$\begin{aligned}
\frac{\mathbb{F}_{3329}[x,y]}{\langle y^{128}+1,\ x^2-y\rangle} &= \frac{\mathbb{F}_{3329}[x,y,z_0]}{\langle z_0^{128}-1,\ x^2-y,\ y-\omega_{256}z_0\rangle} \\
&= \frac{\mathbb{F}_{3329}[x,y,z_0]}{\langle z_0^{64}-1,\ x^2-y,\ y-\omega_{256}z_0\rangle} \oplus \frac{\mathbb{F}_{3329}[x,y,z_0]}{\langle z_0^{64}+1,\ x^2-y,\ y-\omega_{256}z_0\rangle} \\
&= \frac{\mathbb{F}_{3329}[x,y,z_0,z_1]}{\langle z_1^{64}-1,\ x^2-y,\ y-\omega_{256}z_0,\ z_0-z_1\rangle} \oplus \frac{\mathbb{F}_{3329}[x,y,z_0,z_1]}{\langle z_1^{64}-1,\ x^2-y,\ y-\omega_{256}z_0,\ z_0-\omega_{128}z_1\rangle} \\
&= \frac{\mathbb{F}_{3329}[\ldots]}{\langle z_1^{32}-1,\ \ldots,\ z_0-z_1\rangle} \oplus \frac{\mathbb{F}_{3329}[\ldots]}{\langle z_1^{32}+1,\ \ldots,\ z_0-z_1\rangle} \\
&\quad \oplus \frac{\mathbb{F}_{3329}[\ldots]}{\langle z_1^{32}-1,\ \ldots,\ z_0-\omega_{128}z_1\rangle} \oplus \frac{\mathbb{F}_{3329}[\ldots]}{\langle z_1^{32}+1,\ \ldots,\ z_0-\omega_{128}z_1\rangle} \\
&= \cdots = \bigoplus_{(b_1,\ldots,b_7)\in\{0,1\}^7} \frac{\mathbb{F}_{3329}[x,y,z_0,\ldots,z_7]}{\langle z_7-1,\ x^2-y,\ y-\omega_{256}z_0,\ z_0-\omega_{128}^{b_1}z_1,\ \ldots,\ z_6-\omega_2^{b_7}z_7\rangle},
\end{aligned}$$

the final twists using $\omega_2 = -1$. At each leaf, $z_7 = 1$ and the chain of substitutions composes to $y = \omega_{256}\,\omega_{128}^{b_1}\cdots\omega_2^{b_7} = \omega_{256}^{1+2b_1+4b_2+\cdots+128b_7}$; enumerated in the natural in-place order, this exponent at output slot t is exactly $\mathrm{brv}_8(128+t)$. The result is

$$\frac{\mathbb{F}_{3329}[x]}{\langle x^{256}+1\rangle} \cong \bigoplus_{t=0}^{127} \frac{\mathbb{F}_{3329}[x]}{\left\langle x^2-\omega_{256}^{\mathrm{brv}_8(128+t)}\right\rangle}$$

—the same 128 quadratic factors as the straight Cooley–Tukey incomplete NTT of Section 7 (there listed by odd exponents $2j+1$, here arriving in the bit-reversed order in which the in-place plain tower also produces them). This is Corollary 5.3 in action, and it is worth pausing on what was *not* needed: no primitive 512th root of unity ever appears, because every twist constant is a power of ω_{256} acting on y, never on x.

And this is how Cooley–Tukey butterflies both ways (Section 6) extend to incomplete NTTs. The plain and the twisted y-towers realize the identical map onto $\bigoplus_t \mathbb{F}_{3329}[x]/\langle x^2-\omega_{256}^{\mathrm{brv}_8(128+t)}\rangle$, so a Kyber multiplication may run the forward transform down the plain tower (Cooley–Tukey butterflies, acting componentwise on the coefficient pairs), perform the 128 quadratic base multiplications as its pointwise stage, and invert along the twisted tower (again Cooley–Tukey butterflies, now with inverted twiddles and a deferred scaling-twist by $128^{-1}\omega_{256}^{-j}$); dually, Gentleman–Sande butterflies may be used in both directions. In the layout of Section 8, both directions act componentwise on the even and odd strands [6].

The pattern is general: for leaf degree d, substitute $y = x^d$ and run the size-(N/d) transform in y—plain, twisted, or any mixture of the two, the root requirements being exactly those of the y-transform, that is, of the incomplete NTT itself. Twisting the inert variable directly remains possible in fields with surplus roots (a d-th root of every leaf constant is exactly the supply that would have completed the split), where it can normalize all leaves to the single ring $\mathcal{R}[y]/\langle y^d-1\rangle$ and turn the base multiplication into a constant-free cyclic convolution—an efficiency choice, never a necessity. The mid-tower motivations of Section 4 carry over unchanged: a twist to constant 1 buys a multiplication-free splitting layer, and resets lazily reduced coefficients just before they would overflow (Section 10 quantifies both effects). And since Lemma 5.1 is indifferent to which variable carries the tower, all of this composes freely with incompleteness and with Good's trick.

10 Coefficient Bounds and When to Twist

What does all of this buy an implementer? In production code the twiddle multiplications are Montgomery [29], Barrett [5], or Plantard [32] multiplications, in unsigned or (more usually today) signed variants, and each comes with precise input and output bounds: Seiler's note [36] develops the signed-Montgomery bookkeeping behind the AVX2 Kyber/Dilithium code, "Neon NTT" [6] does the same for Armv8 vector lanes (adding Barrett multiplication for one-known-factor products), and Aoki et al. [3] give the signed Plantard multiplication with its tighter output range. Under lazy reduction, correctness of a transform is then nothing but interval arithmetic over the butterfly network—and the two butterflies of Figure 1 behave completely differently under it.

A Cooley–Tukey butterfly computes $a \pm \zeta^s b$, where the product $\zeta^s b$ has just passed through a reducing multiplication: whatever bound the multiplier guarantees (say $|\zeta^s b| < q$), each CT layer adds that fixed amount to the running bound. Bounds under CT layers therefore grow at a regular, additive pace: inputs below q stay below $(\ell + 1)\, q$ after ℓ layers. A Gentleman–Sande butterfly computes $(a + b,\ (a - b)\,\zeta^s)$: the second output is freshly reduced, but the first is a bare sum—it *doubles* per layer, so bounds under GS layers grow exponentially, like $2^\ell q$. This is the operational content of the 1-versus-$\frac{1}{2}$ proportion of Section 4. One caveat to the per-layer accounting: it presumes the two lanes of a butterfly share a bound, which the trinomial butterfly of the next section does not (Remark 11.2)—rings with a trinomial first layer must track the two lanes separately through that layer.

Twisting is the tool that manages both failure modes, and Lemma 5.1 is the license to use it: the computed values never change, only the bounds bookkeeping.

1. Even with Cooley–Tukey butterflies, enough layers eventually push *every* coefficient toward the top of its word. A twist multiplies every coefficient by a twiddle—every lane passes through a reducing multiplication—so a single twist layer brings the entire array back under control at the same time. This is the quantitative version of the range-control remark in Section 4: twist at the level where $(\ell + 1)\, q$ is about to cross the word bound.

2. With Gentleman–Sande butterflies—typically in the inverse transform—*some* coefficients go out of bounds after only a few layers, because of the doubling. Here, switch to the twisted inverse NTT: by Section 6 the inverse of the twisted tower consists of Cooley–Tukey butterflies (with inverted twiddles), so the exponential growth becomes additive, and it works without extra reductions until $(\ell + 1)\, q$ is about to cross the word bound; and by Corollary 5.3 the computed result is invariant, complete or no.

Dilithium is instructive precisely because it *escapes* the penalty: its modulus is small enough relative to the word. With $q = 2^{23} - 2^{13} + 1 < 2^{23}$ [18] and signed 32-bit coefficients, $2^{31}/q > 2^8 = 256$, and there are > 8 bits of headroom: i.e., $256\, q = 2\,145\,386\,752 < 2^{31}$. Since the complete 256-point transform has exactly 8 layers, even the exponential Gentleman–Sande growth—doubling from below q at every one of the 8 layers—never leaves a 32-bit word, and Dilithium can run either butterfly in either direction with no intermediate reduction at all. Contrast Kyber [4]: $q = 3329$ in signed 16-bit lanes leaves $2^{15}/q < 2^{3.3}$, barely three doublings of headroom, against 7 layers. The additive Cooley–Tukey growth still fits—$8q = 26\,632 < 2^{15} = 32\,768$, so either the forward or the inverse transform using CT needs no mid-transform reduction— but an inverse using Gentleman–Sande doublings does not ($2^7 q \gg 2^{15}$): the inverse must either interleave reductions or twist halfway (option 2 above). That difference in headroom, and nothing deeper, is why bound management dominates Kyber implementation papers and barely appears in Dilithium ones [36, 6, 3].

11 Sub-Radix-3 Butterflies

Everything so far suffices for ML-KEM and ML-DSA; the remaining sections extend the toolkit to ring polynomials beyond $x^N \pm 1$ and to root-poor moduli. Some schemes have a ring polynomial that is not of the form $x^N \pm 1$ at all. Notably NTTRU [28], mentioned in Section 7, and the to-be-standardized NTRU+ [26, 27], work in $\mathcal{R}[x]/\langle \Phi_{6n}(x) \rangle$ with

$$\Phi_{6n}(x) \;=\; x^{2n} - x^n + 1,$$

which is the cyclotomic polynomial of conductor $6n$ precisely when n is 3-smooth (for NTTRU, $n = 384 = 2^7 \cdot 3$ and $6n = 2304$). Because the conductor is divisible by three the ring polynomial is a trinomial, and the first splitting layer is not a Cooley–Tukey butterfly. We call it a *sub-radix-3* butterfly: it is a radix-3 butterfly pruned at *both* ends.

Indeed $x^{3n} + 1 = (x^n + 1)(x^{2n} - x^n + 1)$, and the radix-3 Cooley–Tukey step on $x^{3n} - (-1)$ separates the three cube roots $\zeta \in \{-1, \omega_6, \omega_6^5\}$ of -1, where ω_6 is a primitive sixth root of unity, into the factors $x^n - \zeta$. Now regard our input $f = a + b\,x^n$, with a, b of degree below n, as an element of $\mathcal{R}[x]/\langle x^{3n} + 1 \rangle$ whose third block is zero. The radix-3 butterfly returns $a + \zeta b$ at each of the three ζ, and we discard the $\zeta = -1$ leg, which belongs to the factor $x^n + 1$ that is not part of our ring. Two inputs because the third block is structurally zero, two outputs because the third leg is discarded — and the result resembles an ordinary CT/GS butterfly if one does not look carefully.

Concretely, with $\omega_6^{-1} = \omega_6^5 = 1 - \omega_6$, the split $x^{2n} - x^n + 1 = (x^n - \omega_6)(x^n - \omega_6^5)$ gives

$$\begin{pmatrix} A \\ B \end{pmatrix} = M \begin{pmatrix} a \\ b \end{pmatrix}, \qquad M = \begin{pmatrix} 1 & \omega_6 \\ 1 & 1 - \omega_6 \end{pmatrix}, \qquad \det M = 1 - 2\omega_6 = -\delta,$$

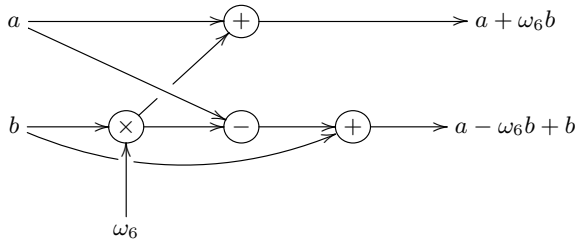
where $\delta := 2\omega_6 - 1 = \omega_6 - \omega_6^5$ is a square root of -3. The butterfly is invertible exactly when 3 is a unit, which is automatic in the regime of interest: ω_6 exists only if $6 \mid q - 1$, and then $3 \nmid q$. What distinguishes M from the usual Cooley–Tukey matrix $\left(\begin{smallmatrix} 1 & \zeta^s \\ 1 & -\zeta^s \end{smallmatrix}\right)$ is the extra $+b$ in the bottom row, so one butterfly costs one multiplication and three additions, against one multiplication and two additions for a plain one. Over the whole layer, on blocks of size n, that is n multiplications and $3n$ additions—against the $4n$ multiplications and $6n$ additions of the full radix-3 butterfly of Section 6. The entire overhead of the trinomial ring is those n extra additions, in one layer, once per transform. Also, inverting is straightforward:

$$M^{-1} = \frac{1}{1 - 2\omega_6} \begin{pmatrix} 1 - \omega_6 & -\omega_6 \\ -1 & 1 \end{pmatrix} = \frac{\delta}{3} \begin{pmatrix} 1 - \omega_6 & -\omega_6 \\ -1 & 1 \end{pmatrix},$$

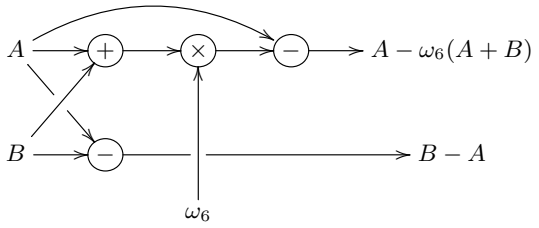
so from the forward outputs A, B we form $\tilde{a} = A - \omega_6(A + B)$ and $\tilde{b} = B - A$—again one multiplication and three additions—whence $a = \frac{\delta}{3}\tilde{a}$ and $b = \frac{\delta}{3}\tilde{b}$. Indeed $A + B = 2a + b$ gives $\tilde{a} = (1 - 2\omega_6)a = -\delta a$ and likewise $\tilde{b} = -\delta b$, and $\frac{\delta}{3} \cdot (-\delta) = 1$.

Both outputs carry the same constant $\frac{\delta}{3}$, and the sub-radix-3 butterfly occurs only in this first layer: below it every modulus has the form $x^m - c$, so the splitting layers are ordinary radix-2 (for NTTRU, seven of them, down to the cubic leaves of Section 7). The scaling can therefore be deferred and folded into the final scaling of the inverse transform, exactly as the factors of $\frac{1}{2}$ are, and costs nothing extra. Figure 3 shows one possible dataflow realization of the two directions.[5]

[5] The companion trinomial $x^{2n} + x^n + 1 = (x^n - \omega_3)(x^n - \omega_3^{-1})$, Φ_{3n} for 3-power n, also splits as a multiply plus 3 adds and a deferred $\delta/3$ with only two sign-changes. Not a coincidence: $\Phi_{6n}(x) = \Phi_{3n}(-x)$ for odd n. It's not in use due to $n = 3^k$—degrees $2n \in \{6, 18, 54, 162, 486, 1458\}$ (cf. Section 7) — and only radix-3 layers below the split.



(a) Forward ω_6: multiplies first, CT-like



(b) Back ω_6: adds, multiplies, then subtracts; deferred scaling by $\delta/3$; GS-like

Figure 3: Sub-radix-3 butterflies for cyclotomic splits of conductor divisible by three. Panel (b) consumes the outputs of panel (a) and returns $-\delta a$ and $-\delta b$, and a deferred $\delta/3$ restores the inputs.

Remark 11.1 (scope of Lemma 5.1). The invariance lemma is built from the binomial split *(S)* and the twist *(T)*; the trinomial split is neither, so the lemma says nothing *about* the sub-radix-3 layer. It applies to everything *below* it, which is all one needs: the layer occurs once, at the root, and each of its two outputs is a plain $\mathcal{R}[x]/\langle x^n - \omega_6^{\pm 1}\rangle$, on which Sections 4–6 apply verbatim.

Remark 11.2 (bounds, and a symmetric variant). The two outputs do not grow alike under lazy reduction. If the multiplication by ω_6 reduces, then $A = a + \omega_6 b$ is one addition after a reduced product and is bounded by $|a| + q$, while $B = a - \omega_6 b + b$ is two and is bounded by $|a| + q + |b|$: the bottom lane carries the unreduced b through. This is unlike the common Cooley-Tukey butterfly whose two lanes share a bound.[6]

12 Good's Trick (the Good–Thomas NTT)

Good proposed a method [21, 37] to perform a size-$(p_0 p_1)$ NTT, for *coprime* p_0 and p_1, as independent size-p_0 and size-p_1 NTTs. The CRT on exponents gives a ring isomorphism

$$\frac{\mathbb{F}_q[x]}{\langle x^{p_0 p_1} - 1\rangle} \;\cong\; \frac{\mathbb{F}_q[y,z]}{\langle y^{p_0} - 1,\; z^{p_1} - 1\rangle}, \qquad x = yz, \tag{6}$$

under which $x^i = y^{i_0} z^{i_1}$ with $i_0 = i \bmod p_0$ and $i_1 = i \bmod p_1$; the inverse index calculation is the CRT reconstruction ("Ruritanian permutation")

$$i = \big((p_1)^{-1} \bmod p_0\big) \cdot p_1 \cdot i_0 + \big((p_0)^{-1} \bmod p_1\big) \cdot p_0 \cdot i_1 \pmod{p_0 p_1}.$$

The advantage: after the coefficient permutation implied by (6), we can run a y-transform and a z-transform *independently*, and both are transforms modulo $y^{p_0} - 1$ and $z^{p_1} - 1$—plain cyclic transforms with no inter-axis twiddle multiplications, which makes the code simpler and more repetitive than a mixed-radix decomposition of the same length.

In coordinates, Good's isomorphism is a permutation: coefficient a_i is stored at the two-dimensional position (i_0, i_1). Flattening that position y-major (slot $p_1 i_0 + i_1$) or z-major (slot $p_0 i_1 + i_0$) gives two index permutations $\tilde{\imath}$ and $\hat{\imath}$, related to each other by transposition of the $p_0 \times p_1$

[6] An implementer can opt to precompute ω_6^5 and take $B = a + \omega_6^5 b$ directly—two multiplications and two additions, both lanes bounded by $|a| + q$—trading one extra multiplication in a single layer for simpler bookkeeping, cf. Section 10.

i	0	1	2	3	4	5
i_0	0	1	2	0	1	2
i_1	0	1	0	1	0	1
$\hat{\imath}$	0	4	2	3	1	5
$\tilde{\imath}$	0	3	4	1	2	5

i	0	1	2	3	4	5	6	7	8	9	10	11
i_0	0	1	2	0	1	2	0	1	2	0	1	2
i_1	0	1	2	3	0	1	2	3	0	1	2	3
$\hat{\imath}$	0	4	8	9	1	5	6	10	2	3	7	11
$\tilde{\imath}$	0	9	6	3	4	1	10	7	8	5	2	11

Table 1: Good's permutations for sizes $6 = 3 \times 2$ and $12 = 3 \times 4$: slot j of the permuted array holds coefficient $a_{\hat{\imath}_j}$ (z-major layout) or $a_{\tilde{\imath}_j}$ (y-major layout).

array rather than by inversion; Table 1 lists both for sizes $6 = 3 \times 2$ and $12 = 3 \times 4$. For example, for $p_0 = 3$, $p_1 = 2$ the y-major layout of a (zero-padded) polynomial of degree 5 reads

$$a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \;=\; (a_5z + a_2)y^2 + (a_1z + a_4)y + (a_3z + a_0).$$

In practice one rarely performs the permutation as a separate pass: we can permute on the fly, operate the NTT, and redeposit the entries in the correct locations, merging the permutation with the first (and last) butterfly rounds. Where a set of coefficients goes depends only on the residue class of its leading location (modulo p_0, say), plus residual cases from padding zeros. The price of all this is code size: Good's trick tends to inflate the number of distinct code paths, and a code generator makes it much less painful.

As a worked pipeline, take a size-1536 cyclic multiplication with $1536 = 3 \times 512$ (as used, e.g., to multiply polynomials with 761 nonzero coefficients in NTRU Prime [8], cf. Section 14):

1. apply Good's permutation to both multiplicands, landing in $F[y, z]/\langle y^3 - 1,\, z^{512} - 1\rangle$;
2. run a 512-NTT on the z-axis for each of the three y-coefficients of each multiplicand, reaching $\bigoplus_{i=0}^{511} F[y]/\langle y^3 - 1,\, z - \zeta_i\rangle$;
3. do the pointwise "multiplications"—each one a schoolbook 3-convolution in $F[y]/\langle y^3 - 1\rangle$;
4. run the inverse 512-NTTs, returning to $F[y, z]/\langle y^3 - 1,\, z^{512} - 1\rangle$;
5. reverse Good's permutation.

Three remarks. Steps 1 and 5 are frequently merged with neighboring stages, as above. The schoolbook 3-convolution (9 multiplications) is no slower than going through 3-NTTs (≥ 2 multiplications for each of three transforms, plus the pointwise stage). And—consequently—this pipeline *does not need a cube root of unity at all*: the only roots required are those of the 512-NTT. Good's trick thus relaxes root-supply requirements axis by axis, in addition to simplifying the transforms.

13 Incomplete Good's Trick

Good's trick composes with incompleteness, and the combinations multiply. As an example over $\mathbb{F}_{769}$ (note $q - 1 = 768 = 4 \cdot 3 \cdot 64$, so all roots below exist), write $u = x^4$ and apply Good's trick to u, with $u = yz$, $y^3 = 1$, $z^{64} = 1$:

$$\frac{\mathbb{F}_{769}[x]}{\langle x^{768} - 1\rangle} \;\to\; \frac{\mathbb{F}_{769}[x, y, z]}{\langle x^4 - yz,\; y^3 - 1,\; z^{64} - 1\rangle}.$$

Now split the z-axis by the radix-2 recursion of Section 3,

$$\to \frac{\mathbb{F}_{769}[x,y,z]}{\langle x^4-yz,\, y^3-1,\, z^{32}-1\rangle} \oplus \frac{\mathbb{F}_{769}[x,y,z]}{\langle x^4-yz,\, y^3-1,\, z^{32}+1\rangle} \to \cdots \to \bigoplus_{j=0}^{63} \frac{\mathbb{F}_{769}[x,y,z]}{\langle x^4-yz,\, y^3-1,\, z-\omega_{64}^{\mathrm{brv}_6(j)}\rangle},$$

then split the y-axis by a 3-NTT:

$$\to \bigoplus_{j=0}^{63} \bigoplus_{k=0}^{2} \frac{\mathbb{F}_{769}[x,y,z]}{\langle x^4-\omega_3^k\,\omega_{64}^{\mathrm{brv}_6(j)},\; y-\omega_3^k,\; z-\omega_{64}^{\mathrm{brv}_6(j)}\rangle}.$$

We performed a 3-NTT on y and an incomplete 256-point transform (splitting only 64 ways) on z, and are left with 192 residues of degree less than 4, multiplied modulo the quartics $x^4-\omega_3^k\,\omega_{64}^{\mathrm{brv}_6(j)}$. Every axis contributes its own dial—how far to split z, whether to transform or convolve on y, what leaf degree to keep in x—so "incomplete Good" describes a whole design space of decompositions to search for the cheapest one on a given platform.

14 NTT-Unfriendly Rings: Saber and NTRU Prime

Everything so far assumed the scheme hands us a ring in which the transform exists. Often it does not, and the two standard escapes [16, 2] are worth describing, because both reduce to choosing a ring *we* like and paying for the detour in coefficient size.

The common device is to lift to $\mathbb{Z}$. A product in $\mathbb{Z}_q[x]/\langle g(x)\rangle$ is the reduction of the product computed in $\mathbb{Z}[x]$, so if we recover the integer answer exactly we may reduce afterwards—and we are free to compute it modulo any prime p, or product of primes, chosen for the transform's convenience rather than the scheme's. The sizing criterion is elementary. If the coefficients of the two multiplicands lie in centered intervals of lengths A and B, and at most N products contribute to any one output coefficient, then each output lies in an interval of length

$$\frac{N\cdot A\cdot B}{2},$$

and it suffices that the modulus—or the product of the moduli, recombined by CRT—exceed this. The factor $\frac{1}{2}$ is exactly the saving from signed arithmetic: with representatives centered at 0 each multiplicand contributes $\pm A/2$ rather than A, and only the final range doubles back.

Saber: friendly ring, unfriendly modulus. Saber's ring [13] is $\mathbb{Z}_q[x]/\langle x^{256}+1\rangle$—the same cyclotomic as Kyber's, and perfectly well suited to Section 8—but $q=2^{13}$. A power of two has no roots of unity to speak of (its unit group is a 2-group), so no transform exists over $\mathbb{Z}_q$ at all. Here lifting is all that is needed: pick $p\equiv 1 \pmod{256}$, which supplies the 256th roots for the seven layers down to quadratic leaves exactly as in Section 8 (or $p\equiv 1 \pmod{512}$ for a complete transform), run the whole machinery of this note over $\mathbb{F}_p$, and reduce modulo 2^{13} at the end—a mask, essentially free. Sizing: two arbitrary Saber-range operands would need $256\cdot(2^{13})^2/2=2^{33}$, beyond a single 32-bit prime, but the products that actually occur pair a full-range polynomial with a small secret, and A or B collapses accordingly; a single 32-bit prime, or a pair of smaller ones, is comfortable. Nothing else changes—same butterflies, same twisting freedom, same bound analysis as Section 10. (NTRU proper [14] combines both obstructions, power-of-two moduli *and* rings $\mathbb{Z}_q[x]/\langle x^n-1\rangle$ with n prime, and is handled the same way at a larger convolution size [16].)

NTRU Prime: unfriendly ring. NTRU Prime is the harder case, and the modulus is not the problem: $\mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$ with $p = 761$ and $q = 4591$ prime for `sntrup4591761` [8, 9]. The ring polynomial $x^{761} - x - 1$ is irreducible, deliberately, so there is *no* CRT decomposition to exploit, complete, incomplete, or otherwise. The escape is to enlarge rather than to lift: a product of two polynomials with at most 761 terms has degree at most 1520, hence is determined by its image in any cyclic ring of size at least 1521—roughly twice the original. Take $1536 = 3 \times 512$ and run precisely the Good's pipeline of Section 12 [2], which as noted there does not even require a cube root of unity. Sizing by the criterion above with $N = 761$: two full-range operands would need $761 \cdot 4591^2/2 \approx 8.0 \times 10^9$, and again the small operand in the products that matter brings this well within one convenient prime.

Two reductions then remain, and their order matters. Reduce the *polynomial* first, modulo $x^{761} - x - 1$: since $x^{761} \equiv x + 1$, each coefficient above degree 760 folds down into two lower ones, collapsing 1536 coefficients to 761, and that many reductions of the coefficients modulo q instead of 1536. Doing it the other way round costs twice the modular arithmetic for the same answer.

Acknowledgments

The author thanks his students over the years for sitting through these lectures. He thanks Claude Opus 5 for spell- and grammar-checking these notes while overriding most of its suggested edits [ergo blame for any bad English is directly on the author].

References

[1] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe. Post-quantum key exchange – a new hope. *USENIX Security Symposium 2016*, 327–343.

[2] Erdem Alkim, Dean Yun-Li Cheng, Chi-Ming Marvin Chung, Hülya Evkan, Leo Wei-Lun Huang, Vincent Hwang, Ching-Lin Trista Li, Ruben Niederhagen, Cheng-Jhih Shih, Julian Wälde, Bo-Yin Yang. Polynomial multiplication in NTRU Prime: comparison of optimization strategies on Cortex-M4. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021(1): 217–238 (2021). DOI 10.46586/tches.v2021.i1.217-238.

[3] Daichi Aoki, Kazuhiko Minematsu, Toshihiko Okamura, Tsuyoshi Takagi. Efficient word size modular multiplication over signed integers. *ARITH 2022* (29th IEEE Symposium on Computer Arithmetic), 94–101.

[4] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé. CRYSTALS-Kyber: algorithm specifications and supporting documentation (version 3.02). Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 4 August 2021. `https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf`

[5] Paul Barrett. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. *CRYPTO '86*, LNCS 263, 311–323. DOI 10.1007/3-540-47721-7_24.

[6] Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Bo-Yin Yang, Shang-Yi Yang. Neon NTT: faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2022(1): 221–244 (2022). DOI 10.46586/tches.v2022.i1.221-244.

[7] Daniel J. Bernstein. Fast multiplication and its applications. In J. P. Buhler, P. Stevenhagen (eds.), *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*, MSRI Publications 44, Cambridge University Press, 2008, 325–384. DOI 10.1017/9781139049801.011.

[8] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal. NTRU Prime: reducing attack surface at low cost. *SAC 2017*, LNCS 10719, 235–260. DOI 10.1007/978-3-319-72565-9_12.

[9] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, Bo-Yin Yang. NTRU Prime: round 3. Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 7 October 2020. `https://ntruprime.cr.yp.to/nist.html`

[10] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Nicola Tuveri. OpenSSLNTRU: faster post-quantum TLS key exchange. *USENIX Security Symposium 2022*, 845–862.

[11] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé. CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. *EuroS&P 2018*, 353–367. DOI 10.1109/EuroSP.2018.00032. (First-round design, $q = 7681$.)

[12] Alin Bostan, Grégoire Lecerf, Éric Schost. Tellegen's principle into practice. *ISSAC 2003,* 37-44. DOI 10.1145/860854.860870.

[13] Andrea Basso, Jose Maria Bermudo Mera, Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Michiel Van Beirendonck, Frederik Vercauteren. SABER: Mod-LWR based KEM. Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 2020. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`

[14] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, Tsunekazu Saito, John M. Schanck, Peter Schwabe, William Whyte, Keita Xagawa, Takashi Yamakawa, Zhenfei Zhang. NTRU: algorithm specifications and supporting documentation. Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 2020. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`

[15] Georg Bruun. z-transform DFT filters and FFTs. *IEEE Trans. Acoust. Speech Signal Process.* 26(1) (1978), 56–63. DOI 10.1109/TASSP.1978.1163036.

[16] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, Bo-Yin Yang. NTT multiplication for NTT-unfriendly rings: new speed records for Saber and NTRU on Cortex-M4 and AVX2. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021(2): 159–188 (2021). DOI 10.46586/tches.v2021.i2.159-188.

[17] James W. Cooley, John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* 19(90) (1965), 297–301. DOI 10.1090/S0025-5718-1965-0178586-1.

[18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, Damien Stehlé. CRYSTALS-Dilithium: a lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018(1): 238–268 (2018). DOI 10.13154/tches.v2018.i1.238-268.

[19] Pierre Duhamel, Henk Hollmann. Split radix FFT algorithm. *Electronics Letters* 20(1) (1984), 14–16.

[20] W. Morven Gentleman, G. Sande. Fast Fourier transforms: for fun and profit. *AFIPS Fall Joint Computing Conference 1966*, AFIPS Conference Proceedings 29, 563–578. DOI 10.1145/1464291.1464352.

[21] I. J. Good. The interaction algorithm and practical Fourier analysis. *Journal of the Royal Statistical Society, Series B* 20(2) (1958), 361–372. DOI 10.1111/j.2517-6161.1958.tb00300.x. Addendum: *ibid.* 22(2) (1960), 372–375.

[22] Vincent Hwang. Software Implementations of Polynomial Multiplications for Lattice-Based Cryptosystems, Ph.D. Thesis, Radboud University, 2026, `doi.org/10.54195/9789465152516`

[23] Vincent Hwang. A Survey of Polynomial Multiplications for Lattice-Based Cryptosystems, *IACR Commun. Cryptol.* 1(2) (2024), 1–63. DOI 10.62056/A0IVR-10K.

[24] Vincent Hwang. Pushing the Limit of Vectorized Polynomial Multiplications for NTRU Prime, *ACISP 2024*, LNCS 14896, 84–102, `doi.org/10.1007/978-981-97-5028-3_5`.

[25] Vincent Hwang, Chi-Ting Liu, Bo-Yin Yang. Algorithmic views of vectorized polynomial multipliers – NTRU Prime. *ACNS 2024*, LNCS 14584, 24–46. DOI 10.1007/978-3-031-54773-7_2.

[26] J. Kim and J. H. Park, "NTRU+: Compact construction of NTRU using simple encoding method," submission to the Korean Post-Quantum Cryptography (KpqC) Competition, Round 1, v1.0, Oct. 2022. `https://kpqc.or.kr/images/pdf/NTRU+.pdf`.

[27] J. Kim and J. H. Park, "NTRU+: Compact construction of NTRU using simple encoding method," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4760–4774, 2023. `https://doi.org/10.1109/TIFS.2023.3299172`.

[28] Vadim Lyubashevsky, Gregor Seiler. NTTRU: truly fast NTRU using NTT. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2019(3): 180–201 (2019). DOI 10.13154/tches.v2019.i3.180-201.

[29] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation* 44(170) (1985), 519–521. DOI 10.1090/S0025-5718-1985-0777282-X.

[30] Henri J. Nussbaumer. Fast polynomial transform algorithms for digital convolution. *IEEE Trans. Acoust. Speech Signal Process.* 28(2) (1980), 205–215.

[31] Henri J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms*, 2nd ed., Springer, 1982.

[32] Thomas Plantard. Efficient word size modular arithmetic. *IEEE Trans. Emerg. Top. Comput.* 9(3) (2021), 1506–1518. DOI 10.1109/TETC.2021.3073475. (Conference version: *ARITH 2021*, 139.)

[33] Charles M. Rader. Discrete Fourier transforms when the number of data samples is prime. *Proceedings of the IEEE* 56(6) (1968), 1107–1108.

[34] Arnold Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Informatica* 7 (1977), 395–398.

[35] Arnold Schönhage, Volker Strassen. Schnelle Multiplikation großer Zahlen. *Computing* 7 (1971), 281–292. DOI 10.1007/BF02242355.

[36] Gregor Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. Cryptology ePrint Archive, Paper 2018/039. https://eprint.iacr.org/2018/039

[37] L. H. Thomas. Using a computer to solve problems in physics. In *Applications of Digital Computers*, Ginn, 1963.

[38] R. Yavne. An economical method for calculating the discrete Fourier transform. *AFIPS Fall Joint Computer Conference 1968, Part I*, AFIPS Conference Proceedings 33, 115–125. DOI 10.1145/1476589.1476610.

A Appendix: Rader's Trick

Good's trick (Section 12) needs a transform of prime size p; Rader's trick [33] supplies one, turning it into a convolution of the composite length $p-1$.

Let ψ be a p-th root of unity, so that $\mathbb{Z}_q[x]/\langle x^p-1\rangle$ splits completely into the $\langle x-\psi^j\rangle$, and write the transform as $F_k=\sum_{i=0}^{p-1} f_i\psi^{ik}$. The index 0 is the obstruction to any group structure, so set it aside: F_0 is a plain sum of coefficients, and f_0 can be added back at the end, leaving

$$\hat{F}_k \;=\; F_k - f_0 \;=\; \sum_{i=1}^{p-1} f_i\psi^{ik}, \qquad k\in\{1,\dots,p-1\}.$$

What remains is indexed by the *nonzero* residues, and those form a cyclic group of order $p-1$. Take a primitive root g modulo p and pass to discrete logarithms, $i=g^{\hat{\imath}}$ and $j=g^{p-\hat{\jmath}}$; then $ij=g^{\hat{\imath}+p-\hat{\jmath}}$ depends on $\hat{\imath}$ and $\hat{\jmath}$ only through their difference. With $a_n=f_{g^n}$ and $b_n=\psi^{g^{p-n}}$ the whole system collapses to one cyclic convolution of length $p-1$,

$$\hat{F}_{g^{p-\hat{\jmath}}} \;=\; \sum_{\hat{\imath}=1}^{p-1} a_{\hat{\imath}}\, b_{\hat{\jmath}-\hat{\imath}},$$

whose outputs arrive indexed by $g^{-\hat{\jmath}}$ rather than in natural order. A prime-length transform is therefore a composite-length convolution—for the sizes that arise in practice, a power-of-two-friendly one—plus a linear number of additions.

Two riders. *Truncated* Rader NTT: on $(x^p-1)/(x-1)=\prod_{j=1}^{p-1}(x-\psi^j)$, of degree exactly $p-1$, the same convolution[7] does everything with no separate F_0, and the inverse is the identity

$$\left[\psi^{g^{\hat{\imath}+\hat{\jmath}}}\right]_{0\le\hat{\imath},\hat{\jmath}<p-1}\cdot\left[\psi^{-g^{\hat{\imath}+\hat{\jmath}}}-1\right]_{0\le\hat{\imath},\hat{\jmath}<p-1} \;=\; p\cdot\mathrm{Id}.$$

Prime powers Rader: the same device applies one level at a time—for $p^2=9$, two 3-point NTTs, one 6-point convolution, and some additions.

B Appendix: Schönhage's and Nussbaumer's FFTs

Every transform in this note draws its roots of unity from the coefficient ring. Schönhage's and Nussbaumer's FFTs are for when it has none: they build an FFT-friendly environment in which the roots *come from the variable*. Both start by splitting the ring into two variables, keeping only part of the modulus, and then imposing a second, redundant modulus that the answer does not depend on.

[7] What drugs am I on to introduce this – how can this be useful? The answer (Vincent Hwang [24]) is NTRU Prime sntrup761 where $q=4591=17\cdot 270+1$, and a truncated 17-Rader NTT is exactly 16-long, lovely for AVX2.

Schönhage [35, 34]. Take $\mathcal{R}[x]/\langle x^{mn}+1\rangle$ and put $y=x^m$, so that f becomes a two-variable $F(x,y)$ with $\deg_x < m$. Ignore part of the modulus and work only modulo y^n+1, that is, in $\mathcal{R}[x][y]/\langle y^n+1\rangle$. A product of two such polynomials has $\deg_x \le 2m-2$, so for *any* $nk > 2m-2$ we may redundantly reduce modulo $x^{nk}+1$ as well, landing in $\mathcal{R}'[y]/\langle y^n+1\rangle$ with $\mathcal{R}'=\mathcal{R}[x]/\langle x^{nk}+1\rangle$. Now $\mathcal{R}'$ has what we need: x^k is an n-th root of -1. Run the FFT in y over $\mathcal{R}'$; multiplying by a power of x is merely a shift of $\mathcal{R}$-coefficients, so the twiddles cost no scalar multiplications at all. At the end the redundant modulus is discarded and $y\mapsto x^m$ recovers the product.

Nussbaumer [30, 31]. The same device with the roles of the two variables exchanged. From $\mathcal{R}[x]/\langle x^{mnk}+1\rangle$, again put $y=x^m$, and this time keep only $y^{nk}+1$, so that $\mathcal{R}'=\mathcal{R}[y]/\langle y^{nk}+1\rangle$ and the surviving variable is x. Since the product still has $\deg_x \le 2m-2$, we may redundantly reduce modulo $x^{2n}-1$ whenever $2n > 2m-2$—a *cyclic* redundant modulus, where Schönhage's was negacyclic—and correspondingly y^k is a $2n$-th root of 1 in $\mathcal{R}'$. The FFT then runs in x over $\mathcal{R}'$, with powers of y as free twiddles.

Cost, and use in cryptography. Switching the polynomial modulus to get an FFT at all already expands the degree by a factor of about two; Schönhage and Nussbaumer expand it by a further factor of two. What is bought is that no scalar multiplications are needed—but the small polynomials at the leaves still have positive degree, and Good's trick (Section 12) is unavailable, since the principal roots it needs are exactly what one does not have here. So these are worth reaching for only at quite large degrees, and in the hostile-ring setting of Section 14: "interpose rings with roots of unity (Schönhage/Nussbaumer)" is the first of the three options listed in [2], and the OpenSSLNTRU software [10] took it for `sntrup761`. To the author's knowledge that is the only place a Schönhage/Nussbaumer FFT has ever been used in deployed cryptographic software; later NTRU Prime multipliers dropped it in favour of Good's trick together with Rader's and Bruun's FFTs, which avoid the extra doubling [25].

C Appendix: Bruun's FFT

Bruun's FFT [15] splits a ring without any root of unity at all, by working with trinomials instead of binomials. The prototype is

$$x^4+x^2+1=(x^2+x+1)(x^2-x+1),$$

and in general

$$x^{2n}+ax^n+b^2=\big(x^n+\sqrt{2b-a}\,x^{n/2}+b\big)\big(x^n-\sqrt{2b-a}\,x^{n/2}+b\big),$$

which asks only for a square root of $2b-a$—no root of unity is needed. This is what lets negacyclic rings split over fields where -1 is a nonresidue. Over $\mathbb{F}_q$ with $q\equiv 3 \pmod 4$, write $2^v \parallel q+1$; then $x^{2^k}+1$ factors into irreducible quadratics $x^2+\gamma x+1$ when $k\le v-1$, and into 2^{v-1} irreducible trinomials $x^{2^{k-v+1}}+\gamma\,x^{2^{k-v}}-1$ when $k\ge v$.

The radix-2 Bruun butterfly is the corresponding CRT step,

$$\mathrm{Bruun}_{\alpha,\beta}:\ \frac{R[x]}{\langle x^4+(2\beta-\alpha^2)x^2+\beta^2\rangle}\ \to\ \frac{R[x]}{\langle x^2+\alpha x+\beta\rangle}\times\frac{R[x]}{\langle x^2-\alpha x+\beta\rangle},$$

with

$$(\hat a_0,\hat a_1)=\big(a_0-\beta a_2+\alpha\beta a_3,\ a_1-\alpha a_2+(\alpha^2-\beta)a_3\big),$$
$$(\hat a_2,\hat a_3)=\big(a_0-\beta a_2-\alpha\beta a_3,\ a_1+\alpha a_2+(\alpha^2-\beta)a_3\big),$$

and inverse, up to the usual factor of 2,

$$2a_0 = \hat{a}_0 + \hat{a}_2 + \beta\,(\hat{a}_3 - \hat{a}_1)\,\alpha^{-1}, \qquad 2a_2 = (\hat{a}_3 - \hat{a}_1)\,\alpha^{-1},$$
$$2a_1 = \hat{a}_1 + \hat{a}_3 - (\alpha^2 - \beta)(\hat{a}_0 - \hat{a}_2)\,\alpha^{-1}\beta^{-1}, \qquad 2a_3 = (\hat{a}_0 - \hat{a}_2)\,\alpha^{-1}\beta^{-1}.$$

Each direction is a handful of multiplications by the fixed constants $\alpha^{\pm 1}, \beta, \beta^{-1}, \alpha^2 - \beta$ plus additions and subtractions.

Rader's and Bruun's transforms are complementary, and `sntrup761` shows why: $q = 4591$ has $q - 1 = 2 \cdot 3^3 \cdot 5 \cdot 17$, whose Fermat-prime factor 17 feeds Rader's FFT, while $q + 1 = 2^4 \cdot 7 \cdot 41$ supplies the power of two that feeds Bruun's—so a modulus with almost no usable roots of unity still admits a radix-2-friendly decomposition, without the degree doubling of Appendix B [25].

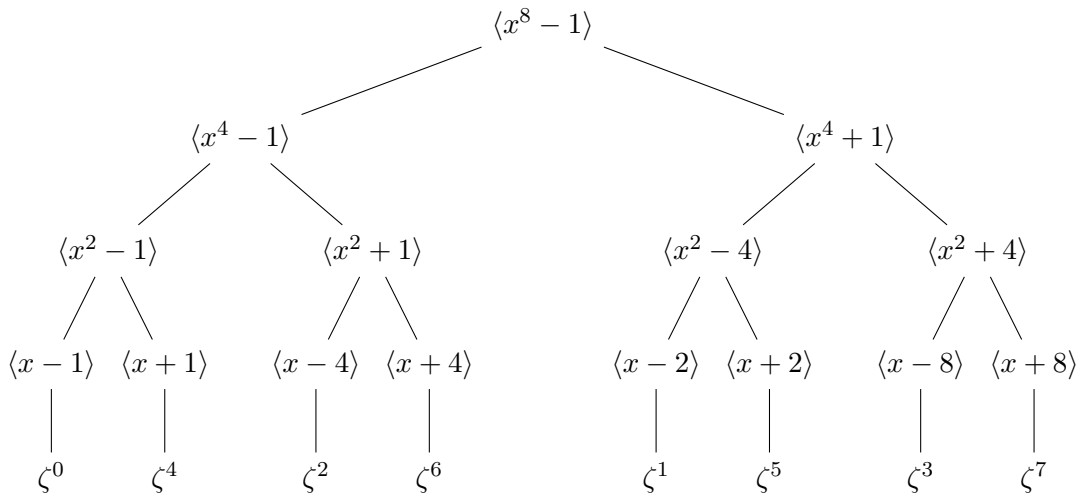


Figure 4: Complete splitting of $\mathbb{F}_{17}[x]/\langle x^8 - 1\rangle$, with $\zeta = 2$ a primitive 8th root of unity modulo 17, so that $\zeta^4 = -1$. A node $\langle x^m - \zeta^s\rangle$ has children $\langle x^{m/2} - \zeta^{s/2}\rangle$ and $\langle x^{m/2} + \zeta^{s/2}\rangle = \langle x^{m/2} - \zeta^{s/2+4}\rangle$; one layer of Cooley-Tukey butterflies with twiddle $\zeta^{s/2}$ realizes each level. The last row gives the leaf exponents: leaf t is $\langle x - \zeta^{\mathrm{brv}_3(t)}\rangle$.