REAL ARITHMETIC SCHEDULES AND OPERATION COUNTS FOR FASTCD

HARRISON LEMLEY

1. Notation

The notation and results in this technical note are based on the mathematical framework and results of [Lem26].

NOTE: This document will be out-dated as of v3 of the arXiv paper, as the contents will be included in this version. It is kept here anyway for the purpose of maintaining compatibility with v2 and keeping a record of past work and paper organization.

Let $R = \mathbb{R}$ and $S = \mathbb{C}$, and fix nonzero real doubling parameters μ_k for $k \geq 1$. Set $A_0 := \mathbb{R}$ and define $A_n := A_{n-1} \oplus A_{n-1}$ with conjugation and multiplication

$$\overline{(a,b)} = (\bar{a}, -b)\,, \qquad (a,b)(c,d) = \left(ac + \mu_n \bar{d}\, b,\ da + b\bar{c}\right), \tag{1.1}$$

where $\bar{r} = r$ for $r \in \mathbb{R}$. Write $N = 2^n$ and use the recursive standard basis $e_0 = 1, e_1, \ldots, e_{N-1}$. The standard real tower has $\mu_k = -1$ for every k. Put

$$V_n := \{x \in A_n : \bar{x} = -x\}\,, \qquad A_n^{\mathbb{C}} := A_n \otimes_{\mathbb{R}} \mathbb{C}\,, \qquad V_n^{\mathbb{C}} := V_n \otimes_{\mathbb{R}} \mathbb{C}\,.$$

Thus V_n consists of the elements whose e_0 coordinate is zero. Define the bilinear pairing in these coordinates by

$$\langle x, y\rangle_n := \sum_{j=0}^{N-1} w_{n,j} x_j y_j\,, \qquad w_0 = (1)\,, \qquad w_n = (w_{n-1}, -\mu_n w_{n-1})\,. \tag{1.2}$$

Conjugation is extended $\mathbb{C}$-linearly and the pairing $\mathbb{C}$-bilinearly. For $X, Y \in V_n$, write

$$X \times_n Y := \frac{1}{2}(XY - YX)\,,$$

and let C_n denote its $\mathbb{C}$-bilinear extension to $V_n^{\mathbb{C}}$. For $x = \alpha + X$ and $y = \beta + Y$, with $\alpha, \beta \in \mathbb{R}$ and $X, Y \in V_n$, full multiplication is recovered by

$$xy = (\alpha\beta - \langle X, Y\rangle_n) + \alpha Y + \beta X + X \times_n Y\,. \tag{1.3}$$

Date: October 2026.

The measured source snapshot is preserved with the benchmark data at fastCD revision 4954189686, including the archived source patch.

At level $n \geq 2$, write $X = (u, \gamma + s)$ and $Y = (v, \delta + t)$, where $u, v, s, t \in V_{n-1}^{\mathbb{C}}$ and $\gamma, \delta \in \mathbb{C}$, and put

$$\begin{aligned} D &:= \mathsf{C}_{n-1}(u, v) + \mu_n \mathsf{C}_{n-1}(s, t)\,, \\ H &:= \mathsf{C}_{n-1}(u, t) + \mathsf{C}_{n-1}(s, v)\,, \\ \lambda &:= \langle s, v\rangle_{n-1} - \langle t, u\rangle_{n-1}\,. \end{aligned}$$

Expanding (1.1) gives

$$\mathsf{C}_n(X, Y) = \big(D + \mu_n(\delta s - \gamma t),\ \lambda + \delta u - \gamma v - H\big)\,. \tag{1.4}$$

The notation $B_n^{\mathbb{C}}(X, Y)$ below uses the triples (u, γ, s) and (v, δ, t) from this splitting. For real coefficients, define the nonrecursive bilinear map by

$$\begin{aligned} B_n\big((u, \gamma, s), (v, \delta, t)\big) := \big(&\mu_n(\delta s - \gamma t),\ \langle s, v\rangle_{n-1} - \langle t, u\rangle_{n-1}\,, \\ &\delta u - \gamma v\big)\,. \end{aligned} \tag{1.5}$$

Write $B_n^{\mathbb{C}}$ for its $\mathbb{C}$-bilinear extension and put $(F, \lambda, G) = B_n^{\mathbb{C}}(X, Y)$. For $n \geq 3$, choose $\rho_n \in \mathbb{C}$ with $\rho_n^2 = \mu_n$ and compute

$$Z_{\pm} := \mathsf{C}_{n-1}(u \pm \rho_n s, v \pm \rho_n t)\,.$$

The transformed inputs belong to $V_{n-1}^{\mathbb{C}}$ because that space is closed under complex linear combinations. Expanding by bilinearity gives

$$\begin{aligned} Z_{\pm} &= \mathsf{C}_{n-1}(u, v) \pm \rho_n \mathsf{C}_{n-1}(u, t) \pm \rho_n \mathsf{C}_{n-1}(s, v) + \rho_n^2 \mathsf{C}_{n-1}(s, t) \\ &= D \pm \rho_n H\,. \end{aligned}$$

Since $\mu_n \neq 0$, also $\rho_n \neq 0$, and hence

$$D = \frac{Z_+ + Z_-}{2}\,, \qquad H = \frac{Z_+ - Z_-}{2\rho_n}\,.$$

Substitution into (1.4) gives the recursive coefficient kernel

$$\mathsf{C}_n(X, Y) = \Big(\frac{Z_+ + Z_-}{2} + F,\ \lambda + G - \frac{Z_+ - Z_-}{2\rho_n}\Big)\,. \tag{1.6}$$

Its level-one base case is $\mathsf{C}_1 \equiv 0$, and its level-two base case is the direct formula $(F, \lambda + G)$, with the scalar λ inserted in the unit coordinate of the second component. The standard-tower schedule uses the octonion formula below as its level-three base case. For real inputs, write $\mathsf{R}_n(X, Y) = X \times_n Y$ and use the real-input reductions of Section 2, followed by (1.3) for full multiplication.

Write T_{A_n} for the real multiplication tensor. Its real tensor rank $\mathrm{R}_{\mathbb{R}}(T_{A_n})$ is the least r for which

$$xy = \sum_{j=1}^{r} \alpha_j(x)\beta_j(y)v_j$$

with real linear forms α_j, β_j and fixed $v_j \in A_n$. In this bilinear model, fixed linear coordinate changes have no cost.

2. Real coefficient specialization

Write ι for the central coefficient satisfying $\iota^2 = -1$. The Cayley–Dickson involution is extended $\mathbb{C}$-linearly and therefore fixes ι, whereas coefficientwise complex conjugation sends ι to $-\iota$. Let $\mathrm{Re_c}$ and $\mathrm{Im_c}$ denote the real and imaginary coefficient projections. Complex addition uses 2 real additions, multiplication by ι requires only a coordinate exchange and a sign change, and complex multiplication can be expressed using 3 real products:

$$\begin{aligned} (a+\iota b)(c+\iota d) &= (p-q)+\iota(r-p-q)\,, \\ p = ac\,, \qquad q &= bd\,, \qquad r = (a+b)(c+d)\,. \end{aligned} \tag{2.1}$$

Write $X = a_0 + \iota a_1$ and $Y = b_0 + \iota b_1$, where $a_0, a_1, b_0, b_1 \in V_n$. Applying this multiplication formula to the nonrecursive map $B = B_n$ in (1.5) gives

$$\begin{aligned} \Delta = B(a_0, b_0)\,, \qquad \Omega &= B(a_1, b_1)\,, \\ W = B(a_0 + a_1, b_0 + b_1)\,, & \\ B^{\mathbb{C}}(a_0 + \iota a_1, b_0 + \iota b_1) &= (\Delta - \Omega) + \iota(W - \Delta - \Omega)\,. \end{aligned} \tag{2.2}$$

Thus the complex kernel evaluates its nonrecursive map by three real evaluations of B, followed by reconstruction of the complex output coefficients.

Corollary 2.1 (One-call real-input root)**.** *Let $n \geq 3$ and $\mu_n < 0$. Put $a_n = \sqrt{-\mu_n}$ and $\rho_n = a_n\iota$. For $X = (u, \gamma + s)$ and $Y = (v, \delta + t)$ in V_n, define*

$$Z := \mathsf{C}_{n-1}(u + \rho_n s, v + \rho_n t)\,, \qquad \lambda := \langle s, v\rangle_{n-1} - \langle t, u\rangle_{n-1}\,.$$

Then

$$X \times_n Y = \Big(\mathrm{Re_c}\, Z + \mu_n(\delta s - \gamma t),\ \lambda + \delta u - \gamma v - a_n^{-1}\,\mathrm{Im_c}\, Z\Big)\,. \tag{2.3}$$

Proof. For $X, Y \in V_n$, the vectors u, v, s, t have real coefficients. Hence the quantities

$$D = \mathsf{C}_{n-1}(u, v) + \mu_n \mathsf{C}_{n-1}(s, t)$$

and

$$H = \mathsf{C}_{n-1}(u, t) + \mathsf{C}_{n-1}(s, v)$$

also have real coefficients. Expanding Z by bilinearity and using $\rho_n = a_n\iota$ gives

$$Z = D + a_n \iota H\,.$$

Therefore

$$\mathrm{Re_c}\, Z = D\,, \qquad \mathrm{Im_c}\, Z = a_n H\,.$$

Substituting these identities into (1.4) gives (2.3). □

This one-call reduction applies at the real-input root when $\mu_n < 0$, while general complex recursive calls still use the two-call formula. For $\mu_n > 0$, choosing $\rho_n = \sqrt{\mu_n} \in \mathbb{R}$ gives real transformed inputs, so two real-input calls at the preceding level suffice. The real-input schedule uses these reductions for $n \geq 3$.

For Propositions 3.1–4.2, take $\mu_k = -1$ for every k, with $\rho_k = \iota$, so the tower is standard. Write M for input-dependent real multiplications, A for real additions and subtractions, and E for multiplications by fixed scalars other than $0, \pm 1$. Each costs 1 operation, so the total is $T = M + A + E$. Known-zero terms are omitted and sums are initialized from their first term. Preprocessing of both variable operands is included in the total cost, while coordinate extraction, copying, permutations, and sign changes are not counted, and scalar and imaginary parts are obtained directly from the standard-basis coordinates.

3. An octonion base case

When $n = 3$, the complex kernel uses the following formula for the alternating product C_3 on the complexified imaginary octonions, whose elements have 7 complex coordinates. With the one-call real-input root, this base case first occurs in full multiplication in A_4, of real dimension 16.

Write an input to C_3 in the standard coordinates as

$$X = (x_1, x_2, x_3, h, y_1, y_2, y_3) \in V_3^{\mathbb{C}}.$$

For $u, v \in \mathbb{C}^3$, let

$$u \times v := (u_2 v_3 - u_3 v_2,\ u_3 v_1 - u_1 v_3,\ u_1 v_2 - u_2 v_1)$$

be the ordinary three-dimensional cross product, extended bilinearly to complex coordinates. Define the following linear forms in X:

$$p = -\iota h\,, \qquad z = x_1 + p\,,$$
$$b = x_2 + \iota y_2\,, \qquad r = x_2 - \iota y_2\,, \qquad c = x_3 + \iota y_3\,, \qquad s = x_3 - \iota y_3\,.$$

Let $L_j \colon \mathbb{C}^7 \to \mathbb{C}^3$, $1 \le j \le 4$, be the linear maps

$$\begin{aligned} L_1 X &= (x_1, y_1, p)\,, & L_2 X &= (b + s, r - c, p)\,, \\ L_3 X &= (z + \iota y_1, b, c)\,, & L_4 X &= (z - \iota y_1, r, s)\,. \end{aligned} \tag{3.1}$$

Their transposes below are ordinary matrix transposes, without complex conjugation.

Proposition 3.1 (Octonion leaf)**.** *For $X, Y \in V_3^{\mathbb{C}}$,*

$$\mathsf{C}_3(X, Y) = -\iota L_1^{\mathsf{T}}(L_1 X \times L_1 Y) + \frac{1}{2} \sum_{j=2}^{4} L_j^{\mathsf{T}}(L_j X \times L_j Y)\,. \tag{3.2}$$

Under the arithmetic convention above, this formula has an evaluation using 72 *input-dependent real multiplications,* 156 *real additions and subtractions, and* 15 *real halvings, for a total of* 243 *operations.*

Proof. To verify the identity, express the alternating trilinear form associated with C_3 as a sum of determinants. For complex linear forms f, g, ℓ on $\mathbb{C}^7$, write $[f\, g\, \ell](X, Y, Z)$ for the determinant whose columns are $(f(X), g(X), \ell(X))$, $(f(Y), g(Y), \ell(Y))$, and $(f(Z), g(Z), \ell(Z))$. The bracket is multilinear and

alternating in f, g, ℓ and in X, Y, Z, and its arguments will be suppressed below. In particular, expansion in the first two forms gives

$$\begin{aligned}[f + \iota g \;\; f - \iota g \;\; \ell] &= -\iota[f\, g\, \ell] + \iota[g\, f\, \ell] \\ &= -2\iota[f\, g\, \ell]\,.\end{aligned}$$

Let

$$\Phi(X, Y, Z) := \langle \mathsf{C}_3(X, Y), Z\rangle_3\,.$$

Since the parameters are standard, the pairing in this expression is the complex-bilinear dot product. In the standard coordinates of $V_2^{\mathbb{C}}$, the alternating product is the ordinary cross product and the pairing is $u \cdot v := \sum_{j=1}^{3} u_j v_j$. Write $X = (x, h, y)$, $Y = (x', h', y')$, and $Z = (x'', h'', y'')$, where $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$, with primes indicating the corresponding coordinates of Y and Z. Formula (1.4) gives

$$\begin{aligned}\mathsf{C}_3(X, Y) = \Big(&x \times x' - y \times y' + hy' - h'y, \\ &y \cdot x' - y' \cdot x,\; h'x - hx' - x \times y' - y \times x'\Big)\,.\end{aligned}$$

Pairing with Z therefore yields

$$\begin{aligned}\Phi = {}&(x \times x' - y \times y' + hy' - h'y) \cdot x'' \\ &+ (y \cdot x' - y' \cdot x)h'' \\ &+ (h'x - hx' - x \times y' - y \times x') \cdot y''\,.\end{aligned}$$

For each index j, the terms involving h, h', h'' are

$$\begin{aligned}&(hy_j' - h'y_j)x_j'' + (y_jx_j' - y_j'x_j)h'' + (h'x_j - hx_j')y_j'' \\ &\quad = [x_j\, h\, y_j](X, Y, Z)\,.\end{aligned}$$

The remaining dot and cross products expand into the four determinants in the first line below. Consequently,

$$\begin{aligned}\Phi = {}&[x_1\, x_2\, x_3] - [x_1\, y_2\, y_3] + [x_2\, y_1\, y_3] - [x_3\, y_1\, y_2] \\ &+ [x_1\, h\, y_1] + [x_2\, h\, y_2] + [x_3\, h\, y_3]\,.\end{aligned} \tag{3.3}$$

Here the coordinate symbols in brackets denote the coordinate linear forms, rather than their values at a particular input.

Put $a = x_1 + \iota y_1$ and $q = x_1 - \iota y_1$. Expanding $[b + s \;\; r - c \;\; p]$ gives

$$[b\, r\, p] - [b\, c\, p] + [s\, r\, p] - [s\, c\, p]\,.$$

The term $-[b\, c\, p]$ cancels $[p\, b\, c]$ from $[a + p \;\; b \;\; c]$, since a cyclic permutation of three forms preserves the bracket. Likewise, $[s\, r\, p] = -[p\, r\, s]$ cancels the corresponding term from $[p + q \;\; r \;\; s]$. The term $-[s\, c\, p]$ is $[c\, s\, p]$, so the remaining terms give

$$\begin{aligned}&[a\, q\, p] + [b + s \;\; r - c \;\; p] + [a + p \;\; b \;\; c] + [p + q \;\; r \;\; s] \\ &\quad = [a\, q\, p] + [b\, r\, p] + [c\, s\, p] + [a\, b\, c] + [q\, r\, s]\,.\end{aligned}$$

For each of the first three brackets on the right, the identity above and $p = -\iota h$ give

$$\begin{aligned}[x_j + \iota y_j \;\; x_j - \iota y_j \;\; p] &= -2\iota[x_j\, y_j \; - \iota h] \\ &= -2[x_j\, y_j\, h] = 2[x_j\, h\, y_j]\,.\end{aligned}$$

In $[a\,b\,c] + [q\,r\,s]$, terms containing an odd number of y coordinates have opposite coefficients and cancel. Terms containing no y coordinates have coefficient 2, and terms containing two have coefficient $2\iota^2 = -2$. Reordering the forms in the latter terms gives

$$\begin{aligned}[a\,b\,c] + [q\,r\,s] = 2\big([x_1\, x_2\, x_3] &- [x_1\, y_2\, y_3] \\ &+ [x_2\, y_1\, y_3] - [x_3\, y_1\, y_2]\big)\,.\end{aligned}$$

Comparison with (3.3) now gives

$$2\Phi = [a\,q\,p] + [b + s \;\; r - c \;\; p] + [a + p \;\; b \;\; c] + [p + q \;\; r \;\; s]\,.$$

After division by 2, the first bracket is $-\iota[x_1\, y_1\, p]$. For any linear map L with rows f, g, ℓ, ordinary matrix transposition gives

$$\begin{aligned}\langle L^{\mathsf{T}}(LX \times LY), Z\rangle_3 &= (LX \times LY) \cdot LZ \\ &= [f\; g\; \ell](X, Y, Z)\,.\end{aligned}$$

The four brackets are therefore the pairings with Z of the four terms on the right of (3.2). Their sum equals $\langle \mathsf{C}_3(X, Y), Z\rangle_3$ for every Z. Pairing the difference with each standard basis vector shows that all its coordinates vanish, which proves (3.2).

For the arithmetic count, evaluate the input linear maps, the four complex cross products, and the output reconstruction in order.

Input linear maps. For each operand, forming z costs 1 complex addition; forming b, r, c, s costs 4; forming $b + s, r - c$ costs 2; and forming $z \pm \iota y_1$ costs 2 more. Thus the input maps require 9 complex additions, or 18 real additions, per operand. Multiplication by ι only exchanges coordinates and changes a sign, so both input maps together cost 36 real additions.

Cross products. Write

$$(\Xi_j, \Psi_j, \Gamma_j) := L_j X \times L_j Y \qquad (1 \le j \le 4)\,.$$

Each complex cross product can be evaluated by three real cross products. For $u = u_0 + \iota u_1$ and $v = v_0 + \iota v_1$, form

$$\Delta = u_0 \times v_0\,, \qquad \Omega = u_1 \times v_1\,, \qquad W = (u_0 + u_1) \times (v_0 + v_1)\,.$$

Then $u \times v = (\Delta - \Omega) + \iota(W - \Delta - \Omega)$. The four complex cross products therefore require twelve real cross products, each using 6 multiplications and 3 subtractions, for a total of 72 multiplications and 36 additions. The sums $u_0 + u_1$ and $v_0 + v_1$ would cost 6 additions per complex cross product, or 24 in total. The third coordinate of both L_1 and L_2 is p, so the sum of the real and imaginary parts of p is computed once and reused in each operand. This saves 2 additions, reducing the input-sum count to 22.

For the jth complex cross product, denote its three real cross products by Δ_j, Ω_j, W_j. The coordinates Γ_1, Γ_2 will be combined directly when computing $\widehat{h}$, so only the other ten complex output coordinates are reconstructed separately. Each uses 1 subtraction for its real part and 2 for its imaginary part, costing 30 additions in total. This stage therefore uses 72 multiplications and $36 + 22 + 30 = 88$ additions.

Output reconstruction. Form the six complex intermediate quantities

$$\begin{aligned} \Theta &= \Xi_3 + \Xi_4\,, & U &= \Xi_3 - \Xi_4\,, \\ d_1 &= \Xi_2 + \Psi_3\,, & d_2 &= \Psi_2 + \Psi_4\,, \\ e_1 &= \Gamma_3 - \Psi_2\,, & e_2 &= \Xi_2 + \Gamma_4\,. \end{aligned}$$

Write the output in the same order as X as $(\widehat{x}_1, \widehat{x}_2, \widehat{x}_3, \widehat{h}, \widehat{y}_1, \widehat{y}_2, \widehat{y}_3)$. The sparse transpose reconstruction gives

$$\begin{aligned} \widehat{x}_1 &= \frac{\Theta}{2} - \iota\Xi_1\,, & \widehat{y}_1 &= \iota\left(\frac{U}{2} - \Psi_1\right)\,, \\ \widehat{x}_2 &= \frac{d_1 + d_2}{2}\,, & \widehat{y}_2 &= \frac{\iota(d_1 - d_2)}{2}\,, \\ \widehat{x}_3 &= \frac{e_1 + e_2}{2}\,, & \widehat{y}_3 &= \frac{\iota(e_1 - e_2)}{2}\,, \\ \widehat{h} &= -\frac{\iota(\Gamma_2 + \Theta)}{2} - \Gamma_1\,. \end{aligned} \tag{3.4}$$

To obtain the last coordinate without reconstructing Γ_1, Γ_2, let p_j, η_j, ω_j be the third coordinates of Δ_j, Ω_j, W_j for $j = 1, 2$. Then

$$\Gamma_j = (p_j - \eta_j) + \iota(\omega_j - p_j - \eta_j)\,.$$

Retain $\theta_R := \mathrm{Re_c}(\Theta/2)$ and $\theta_I := \mathrm{Im_c}(\Theta/2)$ from the computation of $\widehat{x}_1$. Taking real and imaginary parts of the formula for $\widehat{h}$ gives

$$\begin{aligned} \mathrm{Re_c}\,\widehat{h} &= \frac{\omega_2 - p_2 - \eta_2}{2} + \theta_I - p_1 + \eta_1\,, \\ \mathrm{Im_c}\,\widehat{h} &= -\frac{p_2 - \eta_2}{2} - \theta_R - \omega_1 + p_1 + \eta_1\,. \end{aligned}$$

Define

$$e := p_1 + \frac{\eta_2}{2}\,, \qquad f := \eta_1 - \frac{p_2}{2}\,.$$

Collecting the terms involving p_1, η_1, p_2, η_2 then yields

$$\mathrm{Re_c}\,\widehat{h} = f - e + \frac{\omega_2}{2} + \theta_I\,, \qquad \mathrm{Im_c}\,\widehat{h} = e + f - \omega_1 - \theta_R\,.$$

These formulas use the retained product coordinates directly and agree with (3.4).

The six intermediate quantities require 6 complex additions, and the six output coordinates other than $\widehat{h}$ require another 6, giving 24 real additions. Computing e, f costs 2 real additions, and each coefficient of $\widehat{h}$ costs 3, giving 8 more. Halving Θ and U costs 4 real halvings, and the four outputs $\widehat{x}_2, \widehat{y}_2, \widehat{x}_3, \widehat{y}_3$ cost another 8. The last coordinate uses 3 additional halvings

for $\eta_2/2, p_2/2, \omega_2/2$; the retained coefficients of $\Theta/2$ require none. Thus the three stages have costs

	M	A	E
Input linear maps	0	36	0
Four complex cross products	72	88	0
Output reconstruction	0	32	15
Total	72	156	15

and the total number of real operations is 243. □

4. Arithmetic counts

To obtain the arithmetic counts for the recursion, first consider the evaluation of $B_k^{\mathbb{C}}$ in real arithmetic by (2.2). The linear input programs form the factors of the elementary products, and the linear output program reconstructs the result from their values. Here a linear program consists of binary additions or subtractions and fixed-scalar multiplications, with copying and reuse allowed.

Proposition 4.1 (Addition count for the nonrecursive map)**.** *Let $k \geq 2$ and $m = 2^{k-1} - 1$. Consider the evaluation of $B_k^{\mathbb{C}}$ by (2.2), using the six coordinate products in (1.5) for each real evaluation of B_k. For the given input coordinates and prescribed product formulas, with the product values treated as independent inputs to the output program, the two linear input programs and the linear output program require at least $22m+2$ real additions in total, even when fixed real scalar multiplications are free. The specified evaluation attains this bound and uses $18m$ input-dependent real products.*

Proof. Put $q = 2m + 1$, the number of coordinates in each input and output of B_k. For each $1 \leq i \leq m$, the prescribed elementary products in one real evaluation are

$$\delta s_i\,, \qquad \gamma t_i\,, \qquad \delta u_i\,, \qquad \gamma v_i\,, \qquad s_i v_i\,, \qquad t_i u_i\,.$$

The parameters and pairing weights in the standard tower are signs, so they do not require additional arithmetic operations. To apply (2.2), write the complex inputs as

$$(u, \gamma, s) = (u_0, \gamma_0, s_0) + \iota(u_1, \gamma_1, s_1)\,,$$
$$(v, \delta, t) = (v_0, \delta_0, t_0) + \iota(v_1, \delta_1, t_1)\,,$$

where all quantities on the right have real coordinates. The evaluations Δ and Ω use the real and imaginary parts directly. The evaluation W instead requires the following coordinate sums as its input factors:

$$u_{0,i} + u_{1,i}\,, \quad s_{0,i} + s_{1,i}\,, \quad \gamma_0 + \gamma_1\,,$$
$$v_{0,i} + v_{1,i}\,, \quad t_{0,i} + t_{1,i}\,, \quad \delta_0 + \delta_1\,.$$

Here i ranges from 1 to m in the vector sums, while each scalar sum is formed once. Every listed sum occurs in at least one prescribed product because $m \geq 1$. There are therefore q required sums for each operand, and

computing them directly uses $2q$ additions. Each real evaluation uses $6m$ multiplications, $2m$ subtractions for its vector outputs, and $2m-1$ additions and subtractions for its scalar output. Reconstructing each complex output from the three real evaluations uses 3 additions, so the specified evaluation has multiplication count $18m$ and addition count

$$2q + 3(4m-1) + 3q = 22m + 2\,.$$

For the lower bound, the prescribed products require each input program to produce all of its q listed sums, up to nonzero fixed scaling. For each operand, its real and imaginary input coordinates are independent variables. The required sums involve different pairs of these variables, so they are distinct up to nonzero scalar multiplication, and none is a scalar multiple of an input coordinate. Each required sum must therefore first appear at an addition or subtraction step. Such a step creates at most one new linear form up to scaling, while multiplication by a fixed scalar creates none. Thus each input program needs at least q additions. The two input programs together require at least $2q$ additions, independently of the output program.

For a real matrix J, let $a(J)$ denote the minimum number of additions and subtractions needed to evaluate its linear map when fixed scalar multiplications are free. Let W_B be the $2q \times 18m$ matrix mapping the independent elementary product values to the real and imaginary output coordinates. The products contributing to each vector coordinate of a real evaluation of B_k are its two scalar–coordinate products. The scalar coordinate instead receives the $2m$ products in the pairing difference. No product contributes to another coordinate of that real evaluation.

For the jth output coordinate, let r_j be the number of these products and let ε_j be the row vector of their signs. Thus $r_j = 2$ for each of the $2m$ vector coordinates and $r_j = 2m$ for the scalar coordinate, with every entry of ε_j equal to 1 or -1. In particular, $\sum_j r_j = 6m$. Ordering the product columns by the evaluations Δ, Ω, and W, the block of W_B for this real-imaginary output pair is

$$W_{B,j} = \begin{pmatrix} \varepsilon_j & -\varepsilon_j & 0 \\ -\varepsilon_j & -\varepsilon_j & \varepsilon_j \end{pmatrix}.$$

The zero block has r_j entries. This follows directly from the reconstruction $(\Delta - \Omega) + \iota(W - \Delta - \Omega)$, applied coordinate by coordinate. After reordering rows and columns, W_B consists of these independent blocks. Every column is nonzero, and both rows of every block are nonzero, so W_B has no zero rows or columns.

Let σ_j, χ_j be the two inputs to the transposed block. Its outputs are, up to their signs and repetitions,

$$\sigma_j - \chi_j\,, \qquad \sigma_j + \chi_j\,, \qquad \chi_j\,.$$

Across the q pairs, the $2q$ sums and differences are distinct up to nonzero scaling and are not scalar multiples of input coordinates. The same new-form argument used for the input programs therefore gives $a(W_B^{\mathsf{T}}) \geq 2q$.

Computing these sums and differences directly uses $2q$ additions, after which all other outputs are copies or sign changes. Hence

$$a(W_B^{\mathsf{T}}) = 2q\,.$$

The transposition principle of Kaminski, Kirkpatrick, and Bshouty (1988) states that, for a matrix J without zero rows or columns, with d_{out} rows and d_{in} columns, a linear program can be transposed with an addition-count change of $d_{\text{out}} - d_{\text{in}}$. Applied to optimal programs in both directions, this gives

$$a(J^{\mathsf{T}}) \le a(J) + d_{\text{out}} - d_{\text{in}}\,,$$
$$a(J) \le a(J^{\mathsf{T}}) + d_{\text{in}} - d_{\text{out}}\,,$$

and therefore $a(J^{\mathsf{T}}) = a(J) + d_{\text{out}} - d_{\text{in}}$. Substituting $d_{\text{out}} = 2q$ and $d_{\text{in}} = 18m$ yields

$$a(W_B) = a(W_B^{\mathsf{T}}) + 18m - 2q = 18m\,.$$

Together with the $2q$ additions required by the input programs, this gives the lower bound $18m + 2q = 22m + 2$. The evaluation counted at the beginning attains it, completing the proof. □

Proposition 4.2 (Exact arithmetic counts)**.** *The recursive schedule of Section 1, with the octonion base case of Proposition 3.1 and the one-call real-input root of Corollary 2.1, multiplies two elements of the standard real algebra A_n, $n \ge 4$, $N = 2^n$, using*

$$\begin{aligned} M_{\text{QL}} &= \frac{(36n-69)N}{8} + 10\,, \quad A_{\text{QL}} = \frac{(76n-196)N}{8} + 26\,, \\ E_{\text{QL}} &= \frac{(16n-53)N}{16} + 4\,, \quad T_{\text{QL}} = \frac{(240n-583)N}{16} + 40\,. \end{aligned} \tag{4.1}$$

For $n = 1, 2, 3$, the respective counts (M, A, E) are $(4, 2, 0)$, $(16, 12, 0)$, and $(58, 62, 0)$.

Proof. Let M_k, P_k, H_k denote the numbers of input-dependent real products, real additions and subtractions, and real halvings, respectively, in one evaluation of the complex kernel C_k, including its recursive calls. Proposition 3.1 gives the base counts

$$(M_3, P_3, H_3) = (72, 156, 15)\,.$$

Complex nodes. Fix $k \ge 4$ and put $m = 2^{k-1} - 1$, so that u, v, s, t have m complex coordinates each. By Proposition 4.1, the nonrecursive map uses $18m$ real multiplications and $22m + 2$ additions. Each coordinate of $u \pm \iota s$ requires 2 real additions for each sign, and the same holds for $v \pm \iota t$, giving $8m$ additions for the transformed inputs. Each recursive output has m complex coordinates, so forming $Z_+ + Z_-$ and $Z_+ - Z_-$ uses $4m$ real additions. Combining them with the two vector outputs of the nonrecursive map costs another $4m$ additions. The scalar output λ is inserted in the otherwise zero unit coordinate of the second component and requires no addition.

Halving the sum and difference of the recursive outputs costs $4m$ real halvings. Multiplication by ι only exchanges coordinates and changes signs, so it has no additional arithmetic cost. The work outside the recursive calls is therefore

	M	A	E
Nonrecursive map	$18m$	$22m+2$	0
Transformed inputs	0	$8m$	0
Recursive sums and differences	0	$4m$	0
Vector recombination	0	$4m$	0
Halving	0	0	$4m$
Total	$18m$	$38m+2$	$4m$

Including both recursive calls gives

$$\begin{aligned} M_k &= 2M_{k-1} + 18(2^{k-1}-1)\,, \\ P_k &= 2P_{k-1} + 38(2^{k-1}-1) + 2\,, \\ H_k &= 2H_{k-1} + 4(2^{k-1}-1)\,. \end{aligned}$$

At level j of a kernel evaluation beginning at level k, there are 2^{k-j} nodes, and the recursion ends with 2^{k-3} octonion leaves. Unrolling the recurrences therefore gives

$$\begin{aligned} M_k &= 72\cdot 2^{k-3} + 18\sum_{j=4}^{k} 2^{k-j}(2^{j-1}-1)\,, \\ P_k &= 156\cdot 2^{k-3} + 38\sum_{j=4}^{k} 2^{k-j}(2^{j-1}-1) + 2\sum_{j=4}^{k} 2^{k-j}\,, \\ H_k &= 15\cdot 2^{k-3} + 4\sum_{j=4}^{k} 2^{k-j}(2^{j-1}-1)\,. \end{aligned}$$

For $k \geq 3$, with empty sums interpreted as zero,

$$\begin{aligned} \sum_{j=4}^{k} 2^{k-j} &= 2^{k-3} - 1\,, \\ \sum_{j=4}^{k} 2^{k-j}(2^{j-1}-1) &= (k-3)2^{k-1} - 2^{k-3} + 1\,. \end{aligned}$$

Substitution and collection of terms now yield

$$\begin{aligned} M_k &= (36k-81)2^{k-2} + 18\,, \\ P_k &= (76k-168)2^{k-2} + 36\,, \\ H_k &= (16k-37)2^{k-3} + 4\,. \end{aligned}$$

Real root. For $n \geq 4$, R_n forms $u + \iota s$ and $v + \iota t$ by interleaving real coordinates, at no arithmetic cost, and makes one call to C_{n-1}. With $m = 2^{n-1} - 1$, the real nonrecursive map uses $6m$ multiplications and $4m - 1$

additions. Combining its two vector outputs with the real and imaginary parts of the recursive output costs another $2m$ additions. The work at the root therefore adds $6m$ multiplications and $6m - 1 = 3N - 7$ additions, with no halvings.

Reconstruction. In (1.3), the scalar product $\alpha\beta$ requires 1 multiplication, the pairing requires $N - 1$, and each scalar–vector product requires $N - 1$. Their total is

$$1 + (N - 1) + 2(N - 1) = 3N - 2$$

input-dependent real multiplications. Summing the terms of the pairing costs $N - 2$ additions, and subtracting it from $\alpha\beta$ costs 1 more. Forming $\alpha Y + \beta X$ costs $N - 1$ additions, and adding $X \times_n Y$ costs another $N - 1$, giving

$$(N - 2) + 1 + (N - 1) + (N - 1) = 3N - 3$$

additions. The standard pairing weights are all 1, so reconstruction requires no additional fixed-scalar multiplications.

All charged fixed-scalar operations are consequently halvings in the complex kernel. Summing the kernel call, real root, and reconstruction costs gives

$$\begin{aligned} M_{\mathrm{QL}} &= M_{n-1} + 6(2^{n-1} - 1) + 3N - 2 = M_{n-1} + 6N - 8\,, \\ A_{\mathrm{QL}} &= P_{n-1} + (3N - 7) + (3N - 3) = P_{n-1} + 6N - 10\,, \\ E_{\mathrm{QL}} &= H_{n-1}\,. \end{aligned}$$

Substituting $k = n - 1$ into the kernel formulas gives the first three counts in (4.1). Adding them gives

$$T_{\mathrm{QL}} = M_{\mathrm{QL}} + A_{\mathrm{QL}} + E_{\mathrm{QL}} = \frac{(240n - 583)N}{16} + 40\,.$$

Small dimensions. At $n = 1$, the alternating product vanishes, and the full product is

$$(\alpha + x_1 e_1)(\beta + y_1 e_1) = (\alpha\beta - x_1 y_1) + (\alpha y_1 + \beta x_1) e_1\,.$$

Evaluating the four displayed scalar products uses 4 multiplications, and the two output coordinates use 2 additions, giving $(4, 2, 0)$. At $n = 2$, the real alternating product is the ordinary three-dimensional cross product, with 6 multiplications and 3 subtractions. Full reconstruction adds 10 multiplications and 9 additions, so the counts are $(16, 12, 0)$.

At $n = 3$, the complex call is C_2, whose nonrecursive map has $m = 1$ and $q = 3$. Its three real evaluations use 18 multiplications and 9 additions. The input sums use 6 additions, and the complex output reconstruction uses 9, giving kernel counts $(18, 24, 0)$. The real root has $m = 3$ and adds 18 multiplications and 17 additions, while full reconstruction adds 22 multiplications and 21 additions. Thus the full counts are

$$(18 + 18 + 22,\ 24 + 17 + 21,\ 0) = (58, 62, 0)\,.$$

□

Corollary 4.3 (Real bilinear bound)**.** *For any fixed nonzero real doubling parameters, use the real-input schedule of Section 2, with the complex octonion base case of Proposition 3.1 transported by fixed linear coordinate changes. Let C_n denote its number of input-dependent real products. Then*

$$C_n \le (36n - 93)2^{n-3} + 12 \qquad (n \ge 4)\,. \tag{4.2}$$

Consequently,

$$\mathrm{R}_{\mathbb{R}}(T_{A_n}) \le (36n - 69)2^{n-3} + 10 \qquad (n \ge 4)\,. \tag{4.3}$$

For $n = 1, 2, 3$, this schedule gives the nonoptimal upper bounds 4, 16, and 58, respectively.

Proof. First transport the complex octonion formula to the given parameters. For $1 \le k \le 3$, choose $\tau_k \in \mathbb{C}^\times$ with $\tau_k^2 = -\mu_k$, define D_0 to be the identity on $\mathbb{C}$, and put

$$D_k(a, b) := (D_{k-1}a, \tau_k D_{k-1}b)\,.$$

Inductively, D_k is an isomorphism from the complexification of the parameterized level-k algebra to the complexification of the standard one. To check this, suppose that D_{k-1} is such an isomorphism and commutes with the involution, and write $\widetilde{z} = D_{k-1}z$ for $z = a, b, c, d$. The involution is $\mathbb{C}$-linear, so it fixes τ_k. Multiplication of the images in the standard algebra therefore gives

$$\begin{aligned}
&(\widetilde{a}, \tau_k\widetilde{b})(\widetilde{c}, \tau_k\widetilde{d}) \\
&\quad = (\widetilde{a}\widetilde{c} - \tau_k^2\widetilde{\bar{d}}\widetilde{b},\ \tau_k(\widetilde{d}\widetilde{a} + \widetilde{b}\widetilde{\bar{c}})) \\
&\quad = D_k(ac + \mu_k\bar{d}b,\ da + b\bar{c})\,,
\end{aligned}$$

where the last line uses $\tau_k^2 = -\mu_k$ and the induction hypothesis. This is the image of the parameterized product in (1.1). Moreover,

$$\overline{D_k(a, b)} = (D_{k-1}\bar{a}, -\tau_k D_{k-1}b) = D_k(\bar{a}, -b)\,.$$

Since $\tau_k \ne 0$ and D_{k-1} is invertible, D_k is invertible as well. Starting from D_0, this proves the asserted isomorphisms and their compatibility with the involution. The maps are diagonal in the recursive standard coordinates and preserve the imaginary subspaces.

Write $\mathsf{C}_3^{(\mu)}$ and $\mathsf{C}_3^{\mathrm{std}}$ for the parameterized and standard alternating products. Preservation of multiplication and of the involution gives

$$\mathsf{C}_3^{(\mu)}(X, Y) = D_3^{-1}\mathsf{C}_3^{\mathrm{std}}(D_3X, D_3Y)\,.$$

Upon writing each complex coordinate as two real coordinates, D_3 and D_3^{-1} are fixed real-linear maps. They change the input linear forms and the output recombination but introduce no input-dependent products. Proposition 3.1 therefore supplies a complex alternating-product evaluation with 72 input-dependent real products for any nonzero real parameters. In that evaluation, the input maps L_j and the sums used to evaluate the complex cross products act separately on the two operands. Each input-dependent real multiplication

consequently multiplies a real-linear form in the first operand by a real-linear form in the second, and the output reconstruction is real-linear in these products. Composing the input maps with D_3 and the output map with D_3^{-1} preserves this property. The fixed scalings in the coordinate changes can affect the total arithmetic count, so the exact total in Proposition 4.2 remains a statement about the standard tower.

For the complex kernel, the level-two formula uses $M_2 = 18$ products and the transported level-three formula uses $M_3 = 72$. At higher levels, the parameters and pairing weights contribute only fixed linear operations, while the nonrecursive map uses $18(2^{k-1} - 1)$ input-dependent products. Consequently, the same product recurrence as in Proposition 4.2 applies, giving

$$M_k = (36k - 81)2^{k-2} + 18 \qquad (k \geq 3)\,.$$

The same bilinear property holds throughout the complex recursion. More precisely, in real coordinates, each input-dependent multiplication is the product of a fixed real-linear form in the first kernel input and a fixed real-linear form in the second, and every output coordinate is a fixed real-linear combination of the resulting products. At level 2, this follows from the elementary products of B_2 and the fusion formula (2.2). The transported octonion evaluation establishes the assertion at level 3. For $k \geq 4$, assume it at level $k - 1$. Multiplication by the fixed coefficient ρ_k and formation of the sums and differences in the recursive inputs are real-linear operations on each operand separately. Substituting these transformed inputs into the linear forms from the induction hypothesis therefore gives real-linear forms in the original kernel inputs. The nonrecursive map has the same property: each real evaluation of B_k uses elementary products with one factor from each operand, and the fusion formula forms its inputs separately and recombines its outputs linearly. Finally, (1.6) uses only fixed real-linear combinations of the outputs of the nonrecursive map and the recursive calls. This proves the assertion by induction.

For real inputs, the nonrecursive map at level n uses $6(2^{n-1} - 1)$ products. If $\mu_n < 0$, one complex call at the preceding level suffices by Corollary 2.1. If $\mu_n > 0$, the transformed inputs are real and two real-input calls suffice instead. The real base formulas give $C_1 = 0$ and $C_2 = 6$. At level 3, the nonrecursive map uses $6 \cdot 3 = 18$ products, so the negative-parameter case gives $C_3 \leq M_2 + 18 = 36$, while the positive-parameter case gives $C_3 \leq 2C_2 + 18 = 30$. Hence $C_3 \leq 36$ for either sign.

The real-input schedule also has the bilinear property just proved. Its level-one and level-two formulas have this property directly. At a higher level with $\mu_n < 0$, it follows from the property of the complex kernel, since the transformed inputs are real-linear in the separate operands and both $\mathrm{Re_c}$ and $\mathrm{Im_c}$ are real-linear maps. When $\mu_n > 0$, the transformed inputs are real, so induction applies to the two real-input calls. In both cases, the

real nonrecursive map pairs factors from the separate operands, and the remaining output operations are real-linear.

Put $U_n = (36n - 93)2^{n-3} + 12$ for $n \geq 4$. At level 4, the nonrecursive map uses $6 \cdot 7 = 42$ products. The negative-parameter case gives $C_4 \leq M_3 + 42 = 114$, and the positive-parameter case gives $C_4 \leq 2C_3 + 42 \leq 114$. Since $U_4 = 114$, the bound holds at level 4.

Now let $n \geq 5$ and suppose that $C_{n-1} \leq U_{n-1}$. If $\mu_n < 0$, substitution of the complex-kernel formula yields

$$\begin{aligned} C_n &\leq M_{n-1} + 6(2^{n-1} - 1) \\ &= (36n - 117)2^{n-3} + 18 + 24 \cdot 2^{n-3} - 6 \\ &= (36n - 93)2^{n-3} + 12 = U_n \,. \end{aligned}$$

If $\mu_n > 0$, the two real-input calls and the induction hypothesis give

$$\begin{aligned} C_n &\leq 2U_{n-1} + 6(2^{n-1} - 1) \\ &= (36n - 105)2^{n-3} + 18 \\ &= U_n - (12 \cdot 2^{n-3} - 6) \leq U_n \,, \end{aligned}$$

since $12 \cdot 2^{n-3} - 6 \geq 0$ for $n \geq 5$. This proves (4.2) for every choice of signs of the parameters.

Full reconstruction in (1.3) adds $3N - 2$ input-dependent products. Each of these also multiplies a real-linear form in x by a real-linear form in y: they are the scalar product $\alpha\beta$, the weighted coordinate products in the pairing, and the coordinate products in αY and βX. Since extracting the scalar and imaginary coordinates is real-linear, the products in the alternating-product schedule likewise have this form in the full inputs x and y. Collecting the fixed output coefficients of all counted products therefore gives a decomposition

$$xy = \sum_{j=1}^{r} \alpha_j(x)\beta_j(y)v_j \,, \qquad r \leq C_n + 3N - 2 \,,$$

where α_j, β_j are real-linear forms and each $v_j \in A_n$ is fixed. By the definition of real tensor rank, this proves $\mathrm{R}_{\mathbb{R}}(T_{A_n}) \leq C_n + 3N - 2$. For $n \geq 4$,

$$\begin{aligned} C_n + 3N - 2 &\leq (36n - 93)2^{n-3} + 12 + 24 \cdot 2^{n-3} - 2 \\ &= (36n - 69)2^{n-3} + 10 \,, \end{aligned}$$

which gives (4.3). At $n = 1, 2, 3$, the corresponding counts are $C_1 + 4 = 4$, $C_2 + 10 = 16$, and $C_3 + 22 \leq 58$, respectively. □

For direct multiplication, each Cayley–Dickson step evaluates four products of dimension $N/2$ and combines their coordinates using N additions and subtractions. If $M_{\mathrm{direct}}(n)$ and $A_{\mathrm{direct}}(n)$ denote its operation counts at level n, then

$$\begin{aligned} M_{\mathrm{direct}}(n) &= 4M_{\mathrm{direct}}(n-1) \,, \\ A_{\mathrm{direct}}(n) &= 4A_{\mathrm{direct}}(n-1) + 2^n \,, \end{aligned}$$

with $M_{\text{direct}}(0) = 1$ and $A_{\text{direct}}(0) = 0$. Consequently,

$$M_{\text{direct}} = 4^n = N^2\,, \qquad A_{\text{direct}} = \sum_{j=1}^{n} 4^{n-j}2^j = N^2 - N\,.$$

The only fixed coefficients in the standard product are signs, so $E_{\text{direct}} = 0$ and

$$T_{\text{direct}} = 2N^2 - N\,.$$

For $N \geq 2$, the uniform Cariow–Cariowa (CC) method uses a Hadamard factorization with a sparse correction of the basis-product signs. The counts in Cariow and Cariowa (2015, p. 38) can be recovered as follows, with the power-of-two scalings included under the present convention.

In the standard basis, induction in (1.1) gives

$$e_i e_j = \sigma_n(i,j) e_{i\oplus j}\,, \qquad \sigma_n(i,j) \in \{1,-1\}\,,$$

where $\oplus$ is bitwise exclusive-or on the binary indices. The basis unit gives $\sigma_n(0,j) = \sigma_n(j,0) = 1$, and the imaginary basis elements satisfy

$$\sigma_n(i,i) = -1 \quad (i>0)\,, \qquad \sigma_n(i,j) = -\sigma_n(j,i) \quad (i,j>0,\ i\neq j)\,.$$

Indeed, these last identities follow from $e_ie_j + e_je_i = -2\langle e_i, e_j\rangle_n$ and the standard pairing. Put $\epsilon_0 = -1$ and $\epsilon_k = 1$ for $k > 0$, and define

$$\mathcal{I}_k := \{i : \epsilon_k \sigma_n(i, i\oplus k) = -1\}\,.$$

The unsigned convolution coordinates $c_k := \sum_i x_i y_{i\oplus k}$ then recover the product coordinates by

$$(xy)_k = \epsilon_k \left(c_k - 2 \sum_{i\in\mathcal{I}_k} x_i y_{i\oplus k} \right)\,.$$

Thus the correction replaces exactly the terms whose signs differ from the unsigned convolution, followed by the free sign change ϵ_k.

For $k = 0$, the correction set is $\mathcal{I}_0 = \{0\}$. For $k > 0$, the indices form $N/2$ disjoint pairs $\{i, i\oplus k\}$. The pair $\{0,k\}$ has positive basis-product signs and contributes no correction. In every other pair, both indices are nonzero and their basis products have opposite signs. Exactly one index in that pair therefore belongs to $\mathcal{I}_k$, giving

$$|\mathcal{I}_0| = 1\,, \qquad |\mathcal{I}_k| = N/2 - 1 \quad (k>0)\,.$$

The total number of correction products is consequently

$$\sum_{k=0}^{N-1} |\mathcal{I}_k| = 1 + (N-1)(N/2-1) = \frac{N(N-3)+4}{2}\,.$$

To evaluate the convolution, let $\mathcal{H}_N$ be the Walsh–Hadamard matrix with entries $(\mathcal{H}_N)_{a,b} = (-1)^{a\cdot b}$, where $a\cdot b$ is the binary digit dot product modulo

2. It is symmetric and satisfies $\mathcal{H}_N^2 = NI_N$, where I_N is the $N \times N$ identity matrix. Consequently,

$$c = \frac{1}{N}\mathcal{H}_N\big((\mathcal{H}_N x) \odot (\mathcal{H}_N y)\big)\,,$$

where $\odot$ denotes coordinatewise multiplication. In fact, expansion of the right-hand side gives coefficient $N^{-1}\sum_a (-1)^{a\cdot(i\oplus j\oplus k)}$ on $x_i y_j$ in coordinate k, which is 1 when $i \oplus j = k$ and 0 otherwise. Each transform has n stages of $N/2$ butterflies, each using 2 additions or subtractions, and therefore costs nN additions. The three transforms use $3nN$ additions, the coordinatewise products use N multiplications, and normalization uses N fixed-scalar multiplications.

For a nonempty correction set of size r, summing its products costs $r-1$ additions and subtracting the doubled sum costs 1 more. An empty set requires neither operation. Hence the corrections require $\sum_k |\mathcal{I}_k|$ additions as well as that many multiplications. It follows that

$$M_{\mathrm{CC}} = \frac{N(N-1)}{2} + 2\,, \qquad A_{\mathrm{CC}} = 3nN + \frac{N(N-3)+4}{2}\,.$$

The normalizations and correction doublings are omitted in the published count because they are power-of-two scalings. When $N \geq 4$, every correction set is nonempty, so the schedule charges N normalizations and N doublings, giving $E_{\mathrm{CC}} = 2N$. At $N = 2$, only $\mathcal{I}_0 = \{0\}$ is nonempty. The outputs are then

$$(xy)_0 = -(c_0 - 2x_0y_0)\,, \qquad (xy)_1 = c_1\,,$$

so normalization and the only nonzero correction coordinate give $E_{\mathrm{CC}} = 3$. In particular, the full counts at $N = 2$ are $(3, 7, 3)$.

For completeness, the comparison thresholds can be checked directly from these formulas. For $n \geq 4$ and $N = 2^n$, the normalized differences are

$$\begin{aligned}
\frac{M_{\mathrm{direct}} - M_{\mathrm{QL}}}{N} &= N - \frac{36n-69}{8} - \frac{10}{N}\,,\\
\frac{M_{\mathrm{CC}} - M_{\mathrm{QL}}}{N} &= \frac{N}{2} - \frac{36n-65}{8} - \frac{8}{N}\,,\\
\frac{T_{\mathrm{direct}} - T_{\mathrm{QL}}}{N} &= 2N - \frac{240n-567}{16} - \frac{40}{N}\,,\\
\frac{T_{\mathrm{CC}} - T_{\mathrm{QL}}}{N} &= N - \frac{192n-583}{16} - \frac{36}{N}\,,\\
\frac{M_{\mathrm{CC}} + A_{\mathrm{CC}} - T_{\mathrm{QL}}}{N} &= N - \frac{192n-551}{16} - \frac{36}{N}\,.
\end{aligned}$$

All these expressions increase with n for $n \geq 4$. Their increments from n to $n+1$ are bounded below, respectively, by $N - 9/2$, $N/2 - 9/2$, $2N - 15$, and $N - 12$ for each of the last two; these are positive for $N \geq 16$. At $n = 4$, the five differences are

$$6\,, \quad -\frac{19}{8}\,, \quad \frac{79}{16}\,, \quad \frac{35}{16}\,, \quad \frac{3}{16}\,.$$

The second becomes $11/8 > 0$ at $n = 5$. Together with the small-dimensional counts, this shows that the quasilinear schedule uses fewer input-dependent multiplications than direct multiplication from $N = 8$ onward and than CC from $N = 32$ onward. Its total equals direct multiplication at $N = 2, 4, 8$ and is smaller from $N = 16$ onward, while it is smaller than CC for every $N \geq 2$. At $N = 16$, the quasilinear total is 417, compared with 496 for direct multiplication and 452 for CC. The last normalized difference also proves the advantage over CC for $N \geq 16$ when fixed-scalar costs are omitted, including $M_{\mathrm{CC}} + A_{\mathrm{CC}} = 420 > 417$ at $N = 16$.

References

[Lem26] Harrison Lemley. *Quasilinear multiplication in the Cayley–Dickson tower*. arXiv:2609.11588, 2026.