EXACT RADIX KERNELS FOR TRUNCATED NEUMANN SERIES: NO FOUR-PRODUCT RADIX-15 KERNEL EXISTS, AND $\mu_{15} = 5$

KOREA SUPERINTELLIGENCE LABS

ABSTRACT. Let $T_m(B) = I + B + \cdots + B^{m-1}$ be the radix-m kernel of the truncated Neumann series and let μ_m be the least number of matrix products with which a straight-line program (products of linear combinations, starting from I and B; additions and scalings free) evaluates T_m. Sao (arXiv:2602.11843) gives an exact three-product radix-9 kernel, an approximate four-product radix-15 kernel whose output leaks $0.185\,B^{15} + 0.458\,B^{16}$, and asks whether exact radix-15 kernels exist. We prove that they do not, over every field: a four-product program outputs a polynomial of degree exactly 16 or of degree at most 12; degrees 13, 14 and 15 are impossible. The proof is a degree-parity argument — both factors of the last product are forced to have the same degree, so $\deg P_4$ is even, and $\deg P_4 = 14$ would force $\deg P_3 = 7$, itself twice a degree. Consequently $\mu_{14}, \mu_{15}, \mu_{16} \geq 5$, the degree-16 spillover that Sao observes numerically is forced, and with the five-product construction $T_{15}(B) = T_3(B)\,T_5(B^3)$ of Dimitrov and Coelho one gets $\mu_{15} = 5$ exactly. The exact radix-15 update therefore costs $7/\log_2 15 \approx 1.79$ products per doubling of the series length, worse than radix 9's $5/\log_2 9 \approx 1.58$; more precisely, radix 9 strictly beats every exact radix-m kernel with $2 \leq m \leq 16$, and radix 17 is the first radix the argument leaves open. Two attributions in the source are corrected: the quinary kernel $\mu_5 = 2$ is not in the paper it is credited to, and the "degree 15 in 4 products" scheme it cites is a degree-16 polynomial. Every statement is machine-checked in Lean 4 with no computational reflection and no axiom beyond the three standard ones.

1. INTRODUCTION

The objects. For a square matrix A over a field, the truncated Neumann series is $S_k(A) = I + A + A^2 + \cdots + A^{k-1}$; when $\rho(A) < 1$ it approximates $(I - A)^{-1}$, and on dense problems its cost is measured in matrix–matrix products, additions and scalar multiples being free. Sao [1] organises the evaluation around the *radix-m kernel*

$$T_m(B) = I + B + B^2 + \cdots + B^{m-1},$$

through two identities: block concatenation $S_{mn}(A) = S_n(A)\,T_m(A^n)$ and telescoping $T_m(B)(I - B) = I - B^m$ ([1], equation (2) and Section 2.2). Writing μ_m for *the minimum number of products to evaluate* T_m ([1], Section 1 and the paragraph "Cost model" of Section 2.2), one radix-m update $(S_n, A^n) \mapsto (S_{mn}, A^{mn})$ costs $C(m) = \mu_m + 2$ products, and the series of length $k = m^t$ costs $(C(m)/\log_2 m)\log_2 k$ products. The game is therefore to make the *coefficient* $C(m)/\log_2 m$ small. Since a product at most doubles the degree and T_m has degree $m - 1$, every source states $\mu_m \geq \lceil \log_2(m-1) \rceil$; in particular $\mu_9 \geq 3$ and $\mu_{15} \geq 4$.

The source and its question. Sao's paper contributes an *exact* radix-9 kernel with three products and rational coefficients ([1], Theorem 3.1), giving coefficient $5/\log_2 9 \approx 1.58$, and for radix 15 a four-product kernel obtained by numerical optimisation (L-BFGS-B over a 28-parameter circuit) that matches T_{15} through degree 14 to within 9×10^{-7} but leaks *spillover* $0.185\,B^{15} + 0.458\,B^{16} + O(B^{17})$ ([1], Section 4.2). The spillover breaks telescoping; if the kernel were exact, radix 15 would reach coefficient $6/\log_2 15 \approx 1.54$ and beat radix 9. The paper reports (Section 4.2) "Whether an exact radix-15 kernel exists remains open", says in Section 5 that numerical search "shows that exact radix-15 kernels almost certainly do not exist", and closes (Section 8, first item of the list of directions) with the question, verbatim:

> *Exact vs. approximate boundary*: Can we prove that exact radix-15 kernels do not exist, or find one?

All section, theorem and equation numbers of [1] used in this paper were read off the arXiv PDF of version 1.

What is settled here. They do not exist, over any field, and the exact value is $\mu_{15} = 5$. The mechanism is a degree argument with no search in it. A four-product program has the shape $P_2 = L_2R_2$, $P_3 = L_3R_3$, $P_4 = L_4R_4$, output $q_0 + c_0P_2 + e_0P_3 + g_0P_4$, where each factor is a linear combination of the quantities available at that stage (Definition 2.1). The degrees available at each stage are $\{0, 1, 2\} \cup \{\deg P_2\} \cup \{\deg P_3\}$ and nothing in between (Lemma 3.1): a factor of degree above 4 must be carried by its P_3 term and then has degree exactly $\deg P_3$. So if the output has a nonzero coefficient in a degree ≥ 13, both factors of P_4 have degree ≥ 5, both equal $\deg P_3$, and $\deg P_4 = 2 \deg P_3$ is *even*; 14 would force $\deg P_3 = 7$, and the same argument one level down forces $7 = 2 \deg P_2$. Hence:

- **Degree dichotomy** (Theorem 3.2, Corollary 3.3). A four-product program outputs a polynomial of degree exactly 16 or of degree at most 12; if any coefficient in degrees 13–16 is nonzero, the degree-16 coefficient is nonzero.
- **No exact radix-m kernel with four products for** $m = 14, 15, 16$ (Theorem 3.4): $\mu_{14}, \mu_{15}, \mu_{16} \geq 5$ over every field.
- **Forced spillover** (Corollary 3.5). A four-product circuit whose degree-14 coefficient is 1 — as any exact prefix match has — has a nonzero coefficient at degree 16. Sao's $0.458\,B^{16}$ is not an artefact of the optimiser.
- $\boldsymbol{\mu_{15} = 5}$ (Theorem 4.2), the upper bound being the five-product circuit $T_{15}(B) = T_3(B)\,T_5(B^3)$ of Dimitrov and Coelho [2], verified as a polynomial identity over $\mathbb{Q}$.
- **Radix 9 is optimal below radix 17** (Theorem 5.1): for every $2 \leq m \leq 16$, $m \neq 9$, the coefficient of any exact radix-m kernel is strictly larger than $5/\log_2 9$. The radices that need the new bound are exactly 14, 15 and 16: with the bound $\lceil \log_2(m-1) \rceil$ alone they would have looked *better* than radix 9 (Proposition 5.2).

Two attributions in [1] are corrected in Section 7. The quinary kernel $\mu_5 = 2$ is credited there to Gustafsson, Bertilsson, Klasson and Ingemarsson [4], a two-page paper that contains no kernel construction; the construction and the whole $C(m) = \mu_m + 2$ cost accounting are in Dimitrov and Coelho [2], which [1] does not cite. And the statement that Sastre and Ibáñez "achieved degree 15 in 4 products" cannot be read as an exact degree-15 evaluation: by Theorem 3.2 no four-product program outputs a polynomial of degree exactly 15, and the scheme in question is described by Jarlebring and Lorentzon [3] as a degree-16 polynomial.

What is known. The upper bound $\mu_{15} \leq 5$ is not new: Dimitrov and Coelho [2] prove $f(N, x) = f(K, x)\, f(J, x^K)$ for $N = KJ$ (their Theorem 1, with $f(N, x) = \sum_{n<N} x^n$), give $f(5, x) = 1+(1+y)(x+y)$ with $y = x^2$ in two multiplications (their Table 1), and their odd/even recursion (their equation (4), the general-size form of their prime-size (3)) applied twice gives the same count of five. They prove no lower bounds. Jarlebring and Lorentzon [3] study, in exactly the computational model used here, the set $\Pi^*_{2^m}$ of polynomials computable with m matrix–matrix multiplications, over $\mathbb{C}$; for $m > 2$ they prove $\dim \Pi^*_{2^m} \leq m^2$ (their Corollary 4.1) and then equality (their Theorem 4.2, which is a conjecture in versions 1 and 2), and for $m = 4$ that $\max\{d : \Pi_d \subset \Pi^*_{2^4}\} = 12$ "holds in both a real and a complex sense" (their equation (5.5)). That is a statement about *all* polynomials of a degree; it does not decide whether the particular polynomial T_{15} is computable, and the degree dichotomy of Corollary 3.3 is not stated there. The non-scalar cost model itself goes back to Paterson and Stockmeyer [5]. The founding search of this project found no statement of a lower bound for T_m beyond $\lceil \log_2(m-1) \rceil$ in the literature; at the time of writing [1] has no forward citations recorded by OpenAlex.

Organisation. Section 2 fixes the model and proves the degree-doubling bounds inside it. Section 3 proves the dichotomy and the non-existence theorem. Section 4 gives the five-product kernel and $\mu_{15} = 5$. Section 5 compares coefficients. Section 6 reproduces the two published kernels exactly and records the negative controls. Section 7 states the two errata. Section 8 reports a numerical probe for radix 17 that is *not* part of the formal development. Section 9 describes the verification, Section 10 the frontier, and Appendix A lists the formal statements verbatim.

2. The model and the degree-doubling bounds

Throughout, K is a field and $K[X]$ the polynomial ring. Everything a program builds from I and B by products and linear combinations is a polynomial in B, and Sao treats kernels as polynomials ("Since $\widetilde{T}_{15}(B) \neq T_{15}(B)$ as polynomials, this is an *approximate* kernel", [1], Section 4.2); so we work in $K[X]$ and write

$$T_m = \sum_{i=0}^{m-1} X^i \in K[X], \qquad [T_m]_j = \begin{cases} 1 & j < m \\ 0 & j \geq m, \end{cases}$$

where $[p]_j$ denotes the coefficient of X^j in p. (For matrices: a program that computes $T_m(B)$ for every square matrix B over K, in particular for a nilpotent Jordan block of size ≥ 17, computes T_m as a polynomial identity; this reduction is a remark, not part of the formal development.)

A *straight-line program with k non-scalar products* over K starts from 1 and X; its i-th product is $P_i = L_i R_i$ with L_i, R_i in the K-span of $\{1, X, P_1, \ldots, P_{i-1}\}$, and its output is an element of the span of $\{1, X, P_1, \ldots, P_k\}$. Write $\mu_m = \mu_m(K)$ for the least k for which some such program outputs T_m. This is the model of [1] (whose radix-15 ansatz, equations (4)–(5), fixes $P_1 = B^2$, takes L_i, R_i in the span of $\{I, B, P_1, \ldots, P_{i-1}\}$ and the output $I + \gamma_1 B + \gamma_2 P_1 + \gamma_3 P_2 + \gamma_4 P_3 + P_4$, counting $6+8+10+4 = 28$ free parameters), of [2], and of [3]. Since $P_1 = L_1 R_1$ with $L_1, R_1 \in \operatorname{span}\{1, X\}$ has degree at most 2, every element of $\operatorname{span}\{1, X, P_1\}$ has degree at most 2. The formal development therefore takes the following slightly *larger* class as its definition of a four-product program, so that its impossibility theorems are stronger than the question needs.

Definition 2.1 (four-product program)**.** A four-product program over K consists of polynomials $L_2, R_2, q_3, q_3', q_4, q_4', q_0 \in K[X]$, each of degree at most 2, and scalars $c_3, c_3', c_4, c_4', e_4, e_4', c_0, e_0, g_0 \in K$. It determines

$$\begin{aligned} P_2 &= L_2 R_2, \\ L_3 &= q_3 + c_3 P_2, & R_3 &= q_3' + c_3' P_2, & P_3 &= L_3 R_3, \\ L_4 &= q_4 + c_4 P_2 + e_4 P_3, & R_4 &= q_4' + c_4' P_2 + e_4' P_3, & P_4 &= L_4 R_4, \\ \mathrm{out} &= q_0 + c_0 P_2 + e_0 P_3 + g_0 P_4. \end{aligned}$$

Its *output* is out. Setting $g_0 = 0$ gives every three-product program, $e_0 = g_0 = 0$ every two-product program, and $c_0 = e_0 = g_0 = 0$ every one-product program, in the same sense.

Every straight-line program with at most four non-scalar products outputs the output of some four-product program in the sense of Definition 2.1 (take q's in the span of $\{1, X, P_1\}$); this identification is immediate from the description above and is made at the level of the paper, not inside Lean, where Definition 2.1 *is* the model. Sao's 28-parameter ansatz is the special case with q's in $\operatorname{span}\{I, B, B^2\}$, constant term 1 and $g_0 = 1$.

Proposition 2.2 (degree doubling inside the model)**.** *Let p be a four-product program over K. Then* $\deg \mathrm{out} \leq 16$. *If $g_0 = 0$ then* $\deg \mathrm{out} \leq 8$*; if $e_0 = g_0 = 0$ then* $\deg \mathrm{out} \leq 4$*; if $c_0 = e_0 = g_0 = 0$ then* $\deg \mathrm{out} \leq 2$. *(Degrees are natural-number degrees, with* $\deg 0 = 0$*.)*

Proof. $\deg P_2 \leq 4$, hence $\deg L_3, \deg R_3 \leq 4$ and $\deg P_3 \leq 8$, hence $\deg L_4, \deg R_4 \leq 8$ and $\deg P_4 \leq 16$; the output is a linear combination. These are the $k \leq 4$ cases of the bound $\mu_m \geq \lceil \log_2(m-1) \rceil$ stated in [1] (Section 1 and the paragraph "Lower bound" of Section 2.2): a k-product output has degree at most 2^k, so T_m needs $k \geq \log_2(m-1)$. They are routine and are recorded only because the cost comparison of Section 5 uses exactly them for $m \leq 13$. □

3. The degree dichotomy

Lemma 3.1 (the top of a factor)**.** *Let $L = r + eP$ in $K[X]$ with $e \in K$,* $\deg r \leq d$ *and* $\deg L > d$ *(degrees in $\{-\infty\} \cup \mathbb{N}$). Then $e \neq 0$ and* $\deg L = \deg P$.

Proof. $eP = L - r$ has degree $\deg L$ because $\deg r < \deg L$; so $eP \neq 0$, whence $e \neq 0$ and $\deg(eP) = \deg P$. □

Applied with $d = 2$ to L_3, R_3 and with $d = 4$ to L_4, R_4 (whose parts other than the P_3 term have degree ≤ 4), the lemma says: a factor of the third product of degree > 2 has degree exactly $\deg P_2$, and a factor of the fourth product of degree > 4 has degree exactly $\deg P_3$.

Theorem 3.2. *Let p be a four-product program over a field K and let $13 \leq j \leq 16$. If $[\text{out}]_j \neq 0$, then $[\text{out}]_{16} \neq 0$.*

Proof. Write $\text{out} = \text{out}_3 + g_0 P_4$ with $\text{out}_3 = q_0 + c_0 P_2 + e_0 P_3$, of degree ≤ 8. A nonzero coefficient of out in degree $j \geq 13$ must come from $g_0 P_4$: so $g_0 \neq 0$, $P_4 \neq 0$, and $\deg P_4 \geq j \geq 13$. Both factors of P_4 are nonzero and $\deg P_4 = \deg L_4 + \deg R_4$ with $\deg L_4, \deg R_4 \leq 8$, so both are $\geq 5 > 4$, and by Lemma 3.1 both equal $\deg P_3$. Hence $\deg P_4 = 2 \deg P_3 \in \{14, 16\}$. Suppose $\deg P_4 = 14$, so $\deg P_3 = 7$ and $P_3 \neq 0$. Then $\deg L_3 + \deg R_3 = 7$ with both ≤ 4, so both are $\geq 3 > 2$, and by Lemma 3.1 both equal $\deg P_2$: $7 = 2 \deg P_2$, impossible. So $\deg P_4 = 16$, $[P_4]_{16} = \text{lc}(P_4) \neq 0$, and $[\text{out}]_{16} = g_0 \, [P_4]_{16} \neq 0$. □

Corollary 3.3 (degree dichotomy)**.** *The output of a four-product program over any field has degree exactly 16 or degree at most 12. Degrees 13, 14, 15 are impossible.*

Proof. By Proposition 2.2 the degree is ≤ 16; if $[\text{out}]_{16} = 0$, Theorem 3.2 gives $[\text{out}]_j = 0$ for $13 \leq j \leq 15$. □

Theorem 3.4 (no exact radix-14, 15, 16 kernel with four products)**.** *Let K be any field and $14 \leq m \leq 16$. No four-product program over K outputs T_m. Consequently $\mu_{14}(K), \mu_{15}(K), \mu_{16}(K) \geq 5$.*

Proof. $[T_m]_{m-1} = 1 \neq 0$ with $13 \leq m - 1 \leq 15$, while $[T_m]_{16} = 0$; this contradicts Theorem 3.2. Fewer than four products are covered by Proposition 2.2: three products give degree $\leq 8 < m - 1$. □

This answers the question of [1] quoted in the introduction, for $m = 15$ and also for $m = 14, 16$, against the bound $\lceil \log_2(m-1) \rceil = 4$ that all three satisfy.

Corollary 3.5 (forced spillover)**.** *If a four-product program over K has $[\text{out}]_{14} = 1$, then $[\text{out}]_{16} \neq 0$. More generally $[\text{out}]_{14} \neq 0$ already forces $[\text{out}]_{16} \neq 0$.*

Proof. Theorem 3.2 with $j = 14$. □

Sao's optimised circuit ([1], Section 4.2 and Appendix 2) has $[\text{out}]_{14} \approx 1 \neq 0$, so its nonzero degree-16 term (0.458) is forced by the shape of the circuit, whatever the optimiser does. The theorem says nothing about the degree-15 term: writing out the convolution, $[P_4]_{15} = 2\, e_4 e_4' \, [P_3]_7 [P_3]_8$, which vanishes when $[P_3]_7 = 0$; the explicit program of Proposition 6.3 below has $[\text{out}]_{15} = 0$ and $[\text{out}]_{14} = 4$. (The convolution formula is a paper-level remark; only the witness is formalised.)

Remark 3.6. The argument is uniform in the field and in the coefficients: it never solves an equation. Jarlebring and Lorentzon [3] enumerate "reduction patterns" of their tabular parameterisation and plot the degrees they reach (their Figure 4.1), which would exhibit the same gap at 13–15 for four multiplications; Lemma 3.1 is what makes the gap a theorem about *every* program, including those in which a leading coefficient cancels. Their observation that "X^7 requires four multiplications" (version 3, Section 2, in the unnumbered subsection "Examples of instances of $\Pi^*_{2^m}$") is the same mechanism one level down. Two points of accuracy about that figure. Its caption says only that the Hessenberg matrices of the triplet are reduced and that "the parameter $\epsilon = 0.1$ is selected for visualization purposes"; it does not restrict the field, though their triplets (A, B, c) are taken over $\mathbb{C}$ throughout. And the gap is genuinely visible there: the plot data shipped with the version-3 e-print lists, at four multiplications, the degrees $16, 12, 10, 9, 8, 7, 6, 5, 4, 3, 2$ — so 13, 14, 15 (and 11) are absent. A plot of the degrees that a finite list of reduction patterns happens to reach is of course not a proof that no other program reaches them.

4. Five products suffice: $\mu_{15} = 5$

Proposition 4.1 (a five-product radix-15 kernel; Dimitrov–Coelho)**.** *In $\mathbb{Q}[X]$ let*

$$r_1 = X \cdot X, \quad r_2 = X \cdot r_1, \quad r_3 = r_2 \cdot r_2, \quad r_4 = r_3 \cdot (r_2 + r_3), \quad r_5 = (1 + X + r_1) \cdot (1 + r_2 + r_3 + r_4).$$

Then $r_5 = T_{15}$. Each factor is a 0/1-combination of 1, X and earlier products, so this is a straight-line program with five non-scalar products, and $\mu_{15}(\mathbb{Q}) \le 5$.

Proof. $r_1 = X^2$, $r_2 = X^3$, $r_3 = X^6$, $r_4 = X^9 + X^{12}$, and $(1 + X + X^2)(1 + X^3 + X^6 + X^9 + X^{12}) = T_3(X)\,T_5(X^3) = T_{15}(X)$. The identity is checked in Lean by normalisation in $\mathbb{Q}[X]$. The construction is Dimitrov and Coelho's [2]: their Theorem 1 with $15 = 3 \cdot 5$ and a two-multiplication $f(5,\cdot)$, of which Table 1 gives one (the form used above is the one [1] prints; see Proposition 6.1(a)), applied to x^3; their recursion (their equation (4)) applied twice, $f(15,x) = 1 + (x + x^2)f(7,x^2)$, $f(7,x^2) = 1 + (x^2 + x^4)f(3,x^4)$, gives the same count. Only the formal identity is ours. □

Theorem 4.2 ($\mu_{15} = 5$)**.** *No four-product program over $\mathbb{Q}$ outputs T_{15}, and $r_5 = T_{15}$. Hence $\mu_{15}(\mathbb{Q}) = 5$. The lower bound holds over every field (Theorem 3.4), and the identity $r_5 = T_{15}$ has coefficients in $\{0,1\}$ and holds in $\mathbb{Z}[X]$, so $\mu_{15}(K) = 5$ for every field K; the transfer of the upper bound from $\mathbb{Q}$ to K is a remark, the formal statement being over $\mathbb{Q}$.*

Proof. Theorem 3.4 with $m = 15$ and Proposition 4.1. □

5. Cost comparison: radix 9 is optimal below radix 17

For radices m_1, m_2 and costs C_1, C_2, the comparison of coefficients is an integer comparison:

$$\frac{C_1}{\log_2 m_1} < \frac{C_2}{\log_2 m_2} \iff C_1 \log_2 m_2 < C_2 \log_2 m_1 \iff m_2^{C_1} < m_1^{C_2}.$$

Define the lower bound on $C(m) = \mu_m + 2$ that the results above prove, for $2 \le m \le 16$:

$$c_{\mathrm{low}}(m) = \begin{cases} 2 & m \le 2, \\ 3 & m = 3, \\ 4 & 4 \le m \le 5, \\ 5 & 6 \le m \le 9, \\ 6 & 10 \le m \le 13, \\ 7 & 14 \le m \le 16. \end{cases}$$

For $m \le 13$ this is $\lceil \log_2(m-1) \rceil + 2$, i.e. Proposition 2.2; for $m = 14, 15, 16$ it is one more, by Theorem 3.4. In every case $C(m) \ge c_{\mathrm{low}}(m)$ for every exact radix-m kernel over any field, the identification of programs with Definition 2.1 being the paper-level step noted in Section 2.

Theorem 5.1 (radix 9 beats every other exact radix below 17)**.** *For every integer $2 \le m \le 16$ with $m \ne 9$,*

$$m^5 < 9^{c_{\mathrm{low}}(m)}, \qquad \textit{equivalently} \qquad \frac{5}{\log_2 9} < \frac{c_{\mathrm{low}}(m)}{\log_2 m} \le \frac{C(m)}{\log_2 m}.$$

So the radix-9 coefficient $5/\log_2 9 \approx 1.577$ is strictly smaller than the coefficient of any exact radix-m kernel with $2 \le m \le 16$, $m \ne 9$.

Proof. Fifteen integer inequalities, decided in Lean by kernel evaluation (`decide`); the largest is $16^5 = 1\,048\,576 < 9^7 = 4\,782\,969$ and the tightest is $13^5 = 371\,293 < 9^6 = 531\,441$. Table 1 lists them. □

Proposition 5.2 (the old bound would not have sufficed)**.** $9^6 < 14^5$, $9^6 < 15^5$ *and* $9^6 < 16^5$. *That is, with $C(m) \ge \lceil \log_2(m-1) \rceil + 2 = 6$ alone, radices* 14, 15 *and* 16 *would each have had a conceivable coefficient $6/\log_2 m$ strictly* below $5/\log_2 9$ *(namely* $\approx$ 1.576, 1.536, 1.500*); the improvement from $\mu_m \ge 4$ to $\mu_m \ge 5$ is exactly what settles them.*

Proof. $531\,441 < 537\,824 < 759\,375 < 1\,048\,576$, by `decide`. □

Proposition 5.3 (real form)**.** $\dfrac{5}{\log_2 9} < \dfrac{7}{\log_2 15}$*: an exact radix-15 kernel costs $7/\log_2 15 \approx 1.792$ products per doubling, against $5/\log_2 9 \approx 1.577$ for radix 9.*

TABLE 1. Lower bounds on the update cost $C(m) = \mu_m + 2$ and on the coefficient $C(m)/\log_2 m$ for exact radix-m kernels, $2 \le m \le 16$. "old" is the coefficient that $\lceil \log_2(m-1) \rceil + 2$ alone would permit; boldface marks the three radices where Theorem 3.4 changes the comparison with radix 9. The values at $m = 2, 3, 5, 9$ are attained (Section 6); at $m = 15$ by Theorem 4.2.

m	$c_{\text{low}}(m)$	m^5	$9^{c_{\text{low}}(m)}$	$c_{\text{low}}(m)/\log_2 m$	old
2	2	32	81	2.000	2.000
3	3	243	729	1.893	1.893
4	4	1 024	6 561	2.000	2.000
5	4	3 125	6 561	1.723	1.723
6	5	7 776	59 049	1.934	1.934
7	5	16 807	59 049	1.781	1.781
8	5	32 768	59 049	1.667	1.667
9	5	59 049	59 049	1.577	1.577
10	6	100 000	531 441	1.806	1.806
11	6	161 051	531 441	1.734	1.734
12	6	248 832	531 441	1.674	1.674
13	6	371 293	531 441	1.621	1.621
14	7	537 824	4 782 969	1.839	**1.576**
15	7	759 375	4 782 969	1.792	**1.536**
16	7	1 048 576	4 782 969	1.750	**1.500**

Proof. $15^5 < 9^7$, so $5 \log 15 < 7 \log 9$; divide by $\log 9 \cdot \log 15 > 0$ and by $\log 2$. Formalised over $\mathbb{R}$ with `Real.logb`. □

Radix 17 is the first radix the argument leaves open: degree 16 *is* reachable with four products, $\lceil \log_2 16 \rceil = 4$, and $\mu_{17} = 4$ would give $6/\log_2 17 \approx 1.468$, better than anything published (Section 8).

6. The published kernels, reproduced exactly; controls

The two exact kernels that [1] states were reproduced from its printed coefficients in exact rational arithmetic before anything else was attempted, and are certified as identities in $\mathbb{Q}[X]$.

Proposition 6.1 (the quinary and radix-9 kernels)**.**

(a) *($\mu_5 \le 2$.) With $U = X \cdot X$ and $V = U \cdot (X + U)$, $1 + X + U + V = T_5$ in $\mathbb{Q}[X]$.*

(b) *($\mu_9 \le 3$.) Let $u = X \cdot X$, $v = u \cdot (X + 2u)$, $\hat{p} = 6X + 80u + 40v$, $\hat{q} = 11X - 5u + 10v$ in $\mathbb{Q}[X]$. Then $v = X^3 + 2X^4$ and*

$$1600\, T_9 = 1600 + 1600X + 1534\, u + 750\, v + \hat{p}\, \hat{q}.$$

Moreover $1534/1600 = 767/800$, $750/1600 = 15/32$, and $\hat{p}/40$, $\hat{q}/40$ are exactly $P = \frac{3}{20}X + 2u + v$ and $Q = \frac{11}{40}X - \frac{1}{8}u + \frac{1}{4}v$: this is the circuit of [1]*, Theorem 3.1 (steps (MM1)–(MM3), $T_9 = W + I + B + \frac{767}{800}U + \frac{15}{32}V$ with $W = PQ$), with denominators cleared.*

Proof. Normalisation in $\mathbb{Q}[X]$ and `norm_num` for the eight rational conversions. Neither construction is new: (a) is the two-product evaluation of $f(5, x)$ in [2], Table 1 (there written $1 + (1 + y)(x + y)$, $y = x^2$, a different factorisation with the same count; the form above is the one [1] prints), and (b) is Sao's own theorem. Every printed fraction of [1] is exactly right. □

Proposition 6.2 (negative controls)**.** *With the notation of Proposition 6.1:*

(a) $1600 + 1600X + 1535\, u + 750\, v + \hat{p}\, \hat{q} \neq 1600\, T_9$*: moving the printed* $767/800$ *to* $307/320$ *(*$1534 \to 1535$ *after clearing denominators) breaks the identity.*

(b) $1 + X + U \neq T_5$*: the quinary kernel cannot be truncated.*

Proof. (a) Subtracting the true identity leaves $u = 0$, i.e. $X^2 = 0$, false. (b) Subtracting gives $X^3+X^4 = 0$, whose X^4-coefficient is 1. Both are refuted by the same checker that certifies the kernels, which is the point of recording them. □

Proposition 6.3 (non-vacuity of the model and of Theorem 3.2)**.**

(a) *The four-product program with* $L_2 = X \cdot X$, $R_2 = X + X \cdot X$, $q_0 = 1 + X + X \cdot X$, $c_0 = 1$ *and every other datum* 0 *outputs* T_5*: the model contains the published constructions, using two of its four products.*

(b) *The four-product program with* $L_2 = R_2 = q_3 = q'_3 = X \cdot X$, $c_3 = c'_3 = e_4 = e'_4 = g_0 = 1$ *and every other datum* 0 *— so* $P_2 = X^4$, $P_3 = (X^2 + X^4)^2$, $P_4 = P_3^2$ *— outputs* $X^8 + 4X^{10} + 6X^{12} + 4X^{14} + X^{16}$*; its coefficients at degrees* 14 *and* 16 *are* $4 \neq 0$ *and* $1 \neq 0$.

So the hypothesis of Theorem 3.2 is satisfiable: what four products cannot do is the combination *"nonzero at degree* 14 *and zero at degree* 16*".*

Proof. Unfolding the definitions and normalising in $\mathbb{Q}[X]$. □

7. Two errata in the source

7.1. **The quinary kernel is not in Gustafsson et al.** Sao writes ([1], Section 2.3, and again in Sections 1, 7 and 8): "Gustafsson et al. [5] achieved $\mu_5 = 2$ with the following construction: (i) $U := B^2$; (ii) $V := U(B + U) = B^3 + B^4$; (iii) $T_5(B) := I + B + U + V$." Reference [5] there is the two-page invited paper [4]. This is a literature fact with no formal counterpart, stated here as an erratum on the following evidence:

- The full self-archived postprint of [4] (Linköping University DiVA record `diva2:1121300`, three PDF pages: a cover page and the whole two-page paper) was read on 7 September 2026, and contains *no radix-5 kernel and no kernel construction of any kind* — indeed no polynomial evaluation and no count of matrix products anywhere. It is a multiply–add and latency comparison, for the inversion of an $N \times N$ symmetric positive definite complex matrix in massive-MIMO detection, between an LDL^{T} factorisation with equation solving and K-term Neumann series with $K = 2, 3$ and three choices of the initial guess (diagonal, tri-diagonal, diagonal-plus-column). Its Section 2 says, verbatim: "We have not considered $K \geq 4$ since the arithmetic complexity is significantly higher compared to exact matrix inversion." Its bibliographic data — title, four authors, ARITH 2017, pp. 62–63, DOI `10.1109/ARITH.2017.11` — agrees with Crossref (7 September 2026), whose title record drops the parenthetical "(Invited Paper)".
- The construction *is* published, by Dimitrov and Coelho [2], which [1] does not cite: Table 1 there gives, for $P = 5$, $y = x^2$ and $f(5, x) = 1 + (1 + y) \cdot (x + y)$ — two multiplications — with complexity $1.72\ldots \log_2(N) - 2$ for series of length a power of five (the source's $\mu_5 = 2$ and its quinary coefficient 1.72); the same table gives $P = 7$ in three and $P = 11$ in four multiplications. Its cost accounting ([2], Section 3.2 and the proofs of Theorems 2 and 3: the kernel, one multiplication for the next power via $x^P = 1 + f(P, x)(x - 1)$, one to multiply the running product) is exactly $C(m) = \mu_m + 2$; indeed the five coefficients printed in their Table 1, namely 2, $1.89\ldots$, $1.72\ldots$, $1.78\ldots$ and $1.73\ldots$ at $P = 2, 3, 5, 7, 11$, are $(\mu_P + 2)/\log_2 P$ to the digits shown. These locations were read off the arXiv PDF of [2].
- A further, minor defect in the same reference: the DOI printed for [5] in [1] ends in `ARITH.2017.14` and resolves (Crossref, 7 September 2026) to a different ARITH 2017 paper ("Accelerating Matrix Processing with GPUs", pp. 139–141); the DOI of [4] ends in `ARITH.2017.11`.

The abstract of [1] says "Beyond the known radix-5 kernel, explicit higher-radix constructions were not available"; [2] in fact gives explicit exact kernels for radices 7 and 11 as well, and a mixed-basis scheme with coefficient $1.7932\ldots$ (their Table 2), which Sao's radix-9 kernel beats.

7.2. **"Degree 15 in 4 products".** Section 7 of [1] says: "Sastre and Ibáñez [12] achieved degree 15 in 4 products, exceeding the PS bound of degree 9 achievable with 4 products."

Corollary 7.1. *No straight-line program with four non-scalar products over any field outputs a polynomial of degree exactly* 15 *(nor of degree exactly* 13 *or* 14*).*

Proof. Theorem 3.2 with $j = 15$ (resp. 13, 14). □

So the sentence cannot mean an exact evaluation of a degree-15 polynomial. Independently, Jarlebring and Lorentzon describe the same scheme — they cite it as their reference [25], which is [6] — as follows ([3], Section 1, verbatim from the version-3 PDF): "the approximation known as 15+ corresponds to a polynomial of degree 16 that matches all Taylor coefficients except for the last, i.e., the coefficient for X^{16}." (The wording of this sentence changed between versions — version 2 has "which is the coefficient for X^{16}" and version 1 is phrased differently again — so the version pin matters; the content does not change.) Corollary 7.1 shows this reading is the only one possible. (A bibliographic aside: reference [12] of [1] prints the title "Boosting the Computation of the Matrix Exponential" with DOI `10.3390/math9141600`; that DOI resolves, per Crossref on 7 September 2026, to [6], whose title is the one [3] gives.)

8. Outside the formal development: radix 17

Nothing in this section is a theorem of this paper; it is reported so that the first open radix is priced. Radix 17 is the first radix that the degree argument does not settle: degree 16 is reachable with four products (Proposition 6.3(b)), and an exact four-product radix-17 kernel would give coefficient $6/\log_2 17 \approx 1.468$. Sao's own Table 5 (Appendix 1.4 of [1]) lists approximate four-product kernels for radices 16 and 17 with kernel residuals 1×10^{-2} and 1×10^{-1}; by Theorem 3.4 the radix-16 one cannot be made exact, while for radix 17 the question is open.

Remark 8.1 (a numerical probe, not a refutation). The founding record reports a Levenberg–Marquardt sweep, in pure Python, over the full 30-parameter four-product circuit for radix 17: 200 random restarts of 120 iterations each, 17 residuals; best sum-of-squares residual 1.03×10^{-1}; 11 CPU-minutes. The same sweep on the radix-15 four-product system — which Theorem 3.4 proves has no solution — returns 1.87×10^{-2}. A failed sampling probe is evidence about the probe, not about the claim; the observation that the radix-15 landscape, provably without a solution, looks *closer* to one than the radix-17 landscape only underlines that.

Remark 8.2 (what an exact decision would take). The degree analysis normalises the problem sharply. Suppose a four-product program over K outputs T_{17}. Its degree is 16, so by the proof of Theorem 3.2 $\deg P_4 = 16$, $\deg L_4 = \deg R_4 = \deg P_3 = 8$, $\deg L_3 = \deg R_3 = \deg P_2 = 4$, and $c_3, c_3', e_4, e_4', g_0 \neq 0$. Rescaling L_2, R_2, and compensating in $c_3, c_3', c_4, c_4', c_0$, makes P_2 monic; writing $P_2 = M + s$ with $M = X^4 + \alpha X^3$ and $\deg s \leq 2$, every occurrence of P_2 can be rewritten with M in its place, absorbing s into the degree-≤ 2 terms $q_3, q_3', q_4, q_4', q_0$ — and $M = X^2(X^2 + \alpha X)$ is itself a legal P_2, so nothing is lost. Rescaling L_3, R_3 (which only rescales P_3, absorbed by e_4, e_4', e_0) gives $L_3 = M + u$, $R_3 = M + v$ with $u, v \in \operatorname{span}\{1, X, X^2\}$; rescaling L_4, R_4 (absorbed by g_0) gives $L_4 = P_3 + A$, $R_4 = P_3 + B$ with $A, B \in \operatorname{span}\{1, X, X^2, M\}$. Matching out $= q_0 + c_0 M + e_0 P_3 + g_0 P_4$ against T_{17} coefficientwise then makes an exact radix-17 kernel a solution of 17 polynomial equations in the 21 unknowns α, u (3), v (3), A (4), B (4), q_0 (3), c_0, e_0, g_0, of total degree at most 5. The equations from degrees 16 and 15 are $g_0 = 1$ and $4\alpha g_0 = 1$, so over a field of characteristic $\neq 2$ they fix $g_0 = 1$ and $\alpha = 1/4$, leaving 15 equations in 19 unknowns. A Gröbner-basis or elimination computation, or first a consistency test modulo several primes, would decide it; the founding record of this project priced that above its per-statement budget and parked it. Heuristically, [3] prove $\dim \Pi^*_{2^4} \leq 16$ while the polynomials of degree at most 16 form a 17-dimensional space, so a *specific* such polynomial is generically outside $\Pi^*_{2^4}$; that is a heuristic, not an argument. Nothing in this remark is formalised.

9. Formal verification

Every theorem, proposition and corollary of Sections 2–6 and Corollary 7.1 is formalised and machine-checked in Lean 4 [7] against *Mathlib* [8]; Appendix A reproduces the statements verbatim. The development is three modules, imported in order — `Model.lean` (the model of Definition 2.1 as a

structure `Prog4 K` over an arbitrary field, the degree lemmas, Lemma 3.1 as `degree_eq_of_top`, Proposition 2.2, Theorem 3.2, Theorem 3.4, Corollary 3.5); `Kernels.lean` (Propositions 4.1, 6.1, 6.2, 6.3 over $\mathbb{Q}$); `Cost.lean` (Theorem 4.2, c_{low} as `costLower`, Theorem 5.1, Propositions 5.2 and 5.3) — totalling 727 lines, 21 theorems, 22 lemmas and 23 definitions, importing only the Mathlib files on polynomial degrees and coefficients and on `Real.logb`, not `Mathlib` as a whole. Corollary 3.3 is Theorem 3.2 together with `natDegree_out` and carries no separate declaration. There is no `sorry`, no added axiom, and no computational reflection (`native_decide`); the only decision procedure used is kernel `decide` on the small natural-number inequalities of Theorem 5.1 and Proposition 5.2, all polynomial identities being closed by `ring` or `linear_combination`. The three modules were re-elaborated from source for this version, and `#print axioms` re-run on every declaration named in Appendix A: each of `kernelPoly_coeff`, `coeff_16_ne_zero`, `out_ne_kernelPoly`, `spillover_forced`, the four `natDegree_out` theorems, `quinary_kernel`, `v9_eq`, `radix9_kernel_scaled`, `radix9_printed_coefficients`, `radix15_kernel_five`, `radix9_perturbed_fails`, `quinary_truncated_fails`, `quinaryProg_out`, `bigProg_out_eq`, `bigProg_coeff`, `mu15_eq_five`, `radix9_optimal_below_17` and `radix15_coefficient_gt` depends on exactly `propext`, `Classical.choice` and `Quot.sound`; `old_bound_insufficient` on `propext` alone. Each of the three module checks and the axiom audit takes under a minute of CPU; the exact-rational reproduction of the printed kernels, in a hand-rolled fraction arithmetic, under a second. Not formalised: the identification of arbitrary straight-line programs with Definition 2.1, the matrix-to-polynomial reduction, the transfer of $\mu_{15} \le 5$ from $\mathbb{Q}$ to other fields, the convolution formula for $[P_4]_{15}$, the statements attributed to [1], [2] and [3] (quoted, not re-proved), and everything in Sections 7.1 and 8. The repository is private: repository reference to be added at submission.

10. The frontier

Question 10.1. Does an exact four-product radix-17 kernel exist over $\mathbb{Q}$ (or over any field)?

It would give $6/\log_2 17 \approx 1.468$; the evidence and the price are in Section 8. Three cheaper items follow the same pattern as this paper. First, the lower bound $\mu_m \ge 5$ is proved here for $m = 14$ and 16 as well as 15 (and $\mu_{12} \ge 4$ by Proposition 2.2); a matching five-product (resp. four-product) construction for each — Dimitrov and Coelho's recursion is the first place to look — would pin three more exact values. Second, Jarlebring and Lorentzon report computing, by homotopy continuation, five-multiplication evaluation schemes "for a large number of given polynomials $p \in \Pi_{20}$, including the truncated Taylor expansion of the exponential as well as the function $1/(1-x)$" ([3], Section 5.2, version 3; version 1 has a comma before "as well as"). Since Π_{20} is their set of polynomials of degree exactly 20, the degree-20 truncation of $1/(1-x)$ is T_{21}, so that report reads $\mu_{21} \le 5$ and coefficient $7/\log_2 21 \approx 1.594$ — still worse than radix 9. It is a floating-point computation and they print numbers only for the exponential, so certifying a radix-21 kernel exactly would start from their code repository. Third, the mechanism of Lemma 3.1 is a statement about the *set of degrees* reachable after each product, and a single theorem about that set would replace one theorem per radix; a search over such a coarser degree-set relaxation, recorded in the founding journal of this project and neither formalised nor claimed here, suggests that it improves $\lceil \log_2(m-1) \rceil$ first at $m = 8$ (the X^7 observation of [3]) and then at $m = 12, 14, 15, 16, 20, 22, 23, 24, 26, \ldots$, agreeing with the shortest-addition-chain length of $m-1$ (OEIS A003313) on the whole range $m \le 33$ searched except at $m - 1 = 29$ and 31.

Appendix A. The formal statements

The Lean statements corresponding to the results above, verbatim (statements only; proofs, docstrings and attributes omitted), grouped by module. Throughout, `K` is a field, `K[X]` the polynomial ring, `p : Prog4 K` a four-product program, `C` the constant-polynomial embedding, `.natDegree` the natural-number degree and `.degree` the degree in `WithBot ℕ`; `kernelPoly K m` is T_m.

Correspondence with the text:

result	declaration(s)
Proposition 2.2	the four `natDegree_out` theorems
Theorem 3.2, Corollary 7.1	`coeff_16_ne_zero`
Theorem 3.4	`out_ne_kernelPoly`
Corollary 3.5	`spillover_forced`
Proposition 4.1	`radix15_kernel_five`
Theorem 4.2	`mu15_eq_five`
Theorem 5.1	`radix9_optimal_below_17`
Proposition 5.2	`old_bound_insufficient`
Proposition 5.3	`radix15_coefficient_gt`
Proposition 6.1	`quinary_kernel`, `v9_eq`, `radix9_kernel_scaled`, `radix9_printed_coefficients`
Proposition 6.2	`radix9_perturbed_fails`, `quinary_truncated_fails`
Proposition 6.3	`quinaryProg_out`, `bigProg_coeff` (`bigProg_out_eq` supplies the polynomial)

`Model.lean`

```
noncomputable def kernelPoly (K : Type*) [Field K] (m : ℕ) : K[X] :=
  ∑ i ∈ Finset.range m, X ^ i
lemma kernelPoly_coeff (m j : ℕ) :
    (kernelPoly K m).coeff j = if j < m then 1 else 0
structure Prog4 (K : Type*) [Field K] where
  L₂ : K[X]
  R₂ : K[X]
  hL₂ : L₂.degree ≤ 2
  hR₂ : R₂.degree ≤ 2
  q₃ : K[X]
  q₃' : K[X]
  hq₃ : q₃.degree ≤ 2
  hq₃' : q₃'.degree ≤ 2
  c₃ : K
  c₃' : K
  q₄ : K[X]
  q₄' : K[X]
  hq₄ : q₄.degree ≤ 2
  hq₄' : q₄'.degree ≤ 2
  c₄ : K
  c₄' : K
  e₄ : K
  e₄' : K
  q₀ : K[X]
  hq₀ : q₀.degree ≤ 2
  c₀ : K
  e₀ : K
  g₀ : K
def P₂ : K[X] := p.L₂ * p.R₂
def L₃ : K[X] := p.q₃ + C p.c₃ * p.P₂
def R₃ : K[X] := p.q₃' + C p.c₃' * p.P₂
def P₃ : K[X] := p.L₃ * p.R₃
def L₄ : K[X] := p.q₄ + C p.c₄ * p.P₂ + C p.e₄ * p.P₃
def R₄ : K[X] := p.q₄' + C p.c₄' * p.P₂ + C p.e₄' * p.P₃
def P₄ : K[X] := p.L₄ * p.R₄
def out : K[X] := p.q₀ + C p.c₀ * p.P₂ + C p.e₀ * p.P₃ + C p.g₀ * p.P₄
private lemma degree_eq_of_top {L r P : K[X]} {e : K} {d : ℕ}
    (hL : L = r + C e * P) (hr : r.degree ≤ (d : WithBot ℕ))
    (hd : (d : WithBot ℕ) < L.degree) : L.degree = P.degree
def out₃ : K[X] := p.q₀ + C p.c₀ * p.P₂ + C p.e₀ * p.P₃
theorem natDegree_out : p.out.natDegree ≤ 16
theorem natDegree_out_of_g₀_eq_zero (h : p.g₀ = 0) : p.out.natDegree ≤ 8
theorem natDegree_out_of_two_products (h : p.e₀ = 0) (h' : p.g₀ = 0) :
    p.out.natDegree ≤ 4
theorem natDegree_out_of_one_product (h : p.c₀ = 0) (h' : p.e₀ = 0) (h'' : p.g₀ = 0) :
    p.out.natDegree ≤ 2
theorem coeff_16_ne_zero {j : ℕ} (hj : 13 ≤ j) (hj' : j ≤ 16) (h : p.out.coeff j ≠ 0) :
    p.out.coeff 16 ≠ 0
theorem out_ne_kernelPoly {m : ℕ} (hm : 14 ≤ m) (hm' : m ≤ 16) :
    p.out ≠ kernelPoly K m
theorem spillover_forced (h : p.out.coeff 14 = 1) : p.out.coeff 16 ≠ 0
```

`Kernels.lean` (over $\mathbb{Q}$)

```
theorem quinary_kernel :
    (1 : ℚ[X]) + X + (X * X) + ((X * X) * (X + X * X)) = kernelPoly ℚ 5
def u9 : ℚ[X] := X * X
```

```
def v9 : ℚ[X] := u9 * (X + 2 * u9)
def p9 : ℚ[X] := 6 * X + 80 * u9 + 40 * v9
def q9 : ℚ[X] := 11 * X - 5 * u9 + 10 * v9
theorem v9_eq : v9 = X ^ 3 + 2 * X ^ 4
theorem radix9_kernel_scaled :
    (1600 : ℚ[X]) * kernelPoly ℚ 9
      = 1600 + 1600 * X + 1534 * u9 + 750 * v9 + p9 * q9
theorem radix9_printed_coefficients :
    (1534 : ℚ) / 1600 = 767 / 800 ∧ (750 : ℚ) / 1600 = 15 / 32 ∧
    (6 : ℚ) / 40 = 3 / 20 ∧ (80 : ℚ) / 40 = 2 ∧ (40 : ℚ) / 40 = 1 ∧
    (11 : ℚ) / 40 = 11 / 40 ∧ (-5 : ℚ) / 40 = -(1 / 8) ∧ (10 : ℚ) / 40 = 1 / 4
def r₁ : ℚ[X] := X * X
def r₂ : ℚ[X] := X * r₁
def r₃ : ℚ[X] := r₂ * r₂
def r₄ : ℚ[X] := r₃ * (r₂ + r₃)
def r₅ : ℚ[X] := (1 + X + r₁) * (1 + r₂ + r₃ + r₄)
theorem radix15_kernel_five : r₅ = kernelPoly ℚ 15
theorem radix9_perturbed_fails :
    (1600 : ℚ[X]) + 1600 * X + 1535 * u9 + 750 * v9 + p9 * q9 ≠ 1600 * kernelPoly ℚ 9
theorem quinary_truncated_fails : (1 : ℚ[X]) + X + (X * X) ≠ kernelPoly ℚ 5
def quinaryProg : Prog4 ℚ where   -- data fields only; the degree proofs are omitted
  L₂ := X * X
  R₂ := X + X * X
  q₃ := 0
  q₃' := 0
  c₃ := 0
  c₃' := 0
  q₄ := 0
  q₄' := 0
  c₄ := 0
  c₄' := 0
  e₄ := 0
  e₄' := 0
  q₀ := 1 + X + X * X
  c₀ := 1
  e₀ := 0
  g₀ := 0
theorem quinaryProg_out : quinaryProg.out = kernelPoly ℚ 5
def bigProg : Prog4 ℚ where   -- data fields only
  L₂ := X * X
  R₂ := X * X
  q₃ := X * X
  q₃' := X * X
  c₃ := 1
  c₃' := 1
  q₄ := 0
  q₄' := 0
  c₄ := 0
  c₄' := 0
  e₄ := 1
  e₄' := 1
  q₀ := 0
  c₀ := 0
  e₀ := 0
  g₀ := 1
theorem bigProg_out_eq :
    bigProg.out = X ^ 8 + C 4 * X ^ 10 + C 6 * X ^ 12 + C 4 * X ^ 14 + X ^ 16
theorem bigProg_coeff : bigProg.out.coeff 14 ≠ 0 ∧ bigProg.out.coeff 16 ≠ 0
```

Cost.lean

```
theorem mu15_eq_five :
    (∀ p : Prog4 ℚ, p.out ≠ kernelPoly ℚ 15) ∧ r₅ = kernelPoly ℚ 15
def costLower (m : ℕ) : ℕ :=
  if m ≤ 2 then 2 else if m ≤ 3 then 3 else if m ≤ 5 then 4
  else if m ≤ 9 then 5 else if m ≤ 13 then 6 else 7
theorem radix9_optimal_below_17 :
    ∀ m ∈ Finset.Icc 2 16, m ≠ 9 → m ^ 5 < 9 ^ costLower m
theorem old_bound_insufficient :
    (9 : ℕ) ^ 6 < 14 ^ 5 ∧ (9 : ℕ) ^ 6 < 15 ^ 5 ∧ (9 : ℕ) ^ 6 < 16 ^ 5
theorem radix15_coefficient_gt :
    (5 : ℝ) / Real.logb 2 9 < 7 / Real.logb 2 15
```

References

[1] P. Sao, *Fast Evaluation of Truncated Neumann Series by Low-Product Radix Kernels*, arXiv:2602.11843v1 [math.NA] (cross-list cs.MS), 12 February 2026, Oak Ridge National Laboratory. All numbering used above (Sections 1–8, Theorem 3.1, equations (2), (4), (5), Table 5, Appendices 1.4 and 2) is that of the arXiv PDF of version 1, read on 7 September 2026; the arXiv record showed no later version on that date.

[2] V. S. Dimitrov and D. F. G. Coelho, *On the Computation of Neumann Series*, arXiv:1707.05846v1, 18 July 2017 (the PDF is stamped `[cs.NA]`; the arXiv listing gives primary math.NA, cross-list cs.CC). Theorems 1–3, Table 1, Table 2 and equations (3)–(4) are numbered as in the arXiv PDF of version 1, read on 7 September 2026.

[3] E. Jarlebring and G. Lorentzon, *The polynomial set associated with a fixed number of matrix-matrix multiplications*, arXiv:2504.01500 [math.NA]; v1 2 April 2025, v3 13 August 2025. Corollary 4.1, Theorem 4.2, Figure 4.1, Sections 2, 5.1 and 5.2, equation (5.5) and the Section 1 quotation are those of version 3, read on 7 September 2026 (PDF, and its LaTeX e-print recompiled to resolve the label numbers).

[4] O. Gustafsson, E. Bertilsson, J. Klasson and C. Ingemarsson, *Approximate Neumann Series or Exact Matrix Inversion for Massive MIMO? (Invited Paper)*, in: 2017 IEEE 24th Symposium on Computer Arithmetic (ARITH), pp. 62–63; DOI `10.1109/ARITH.2017.11` (Crossref, 7 September 2026); self-archived postprint at Linköping University DiVA, `diva2:1121300`, read in full on 7 September 2026.

[5] M. S. Paterson and L. J. Stockmeyer, *On the number of nonscalar multiplications necessary to evaluate polynomials*, SIAM Journal on Computing 2 (1973), no. 1, 60–66; DOI `10.1137/0202007`. Cited here as the origin of the cost model, via [1] and [3]; not read first-hand.

[6] J. Sastre and J. Ibáñez, *Efficient evaluation of matrix polynomials beyond the Paterson–Stockmeyer method*, Mathematics 9 (2021), no. 14, article 1600; DOI `10.3390/math9141600` (title and authors from Crossref, 7 September 2026). Reference [12] of [1] and reference [25] of [3]; not read first-hand, cited only as the paper those two sources refer to.

[7] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE 28, Lecture Notes in Computer Science, Springer, 2021, 625–635; DOI `10.1007/978-3-030-79876-5_37`.

[8] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381; DOI `10.1145/3372885.3373824`.