

The Spectrum of the Fractal Kitaev Model:

An Exact Recursion, an $O(k)$ Eigenvalue Counter, and Exact Multiplicity Evidence for a Candidate Atomic Component

Fabrizio Vassallo

vassalloff@gmail.com

September 26, 2026

Abstract

The Fractal Kitaev Model K_k , constructed in the companion paper [1] by decorating the hierarchical graph G_k into a maximum-degree-3 tree, has an isotropic Majorana spectrum [2] that was previously accessed only by brute-force diagonalization, limited to $k \leq 6$ by memory. This note establishes four results. First, an exact recursive factorization of the characteristic polynomial of an isolated depth- d gadget subtree, $A_d(x) = A_{d-1}(x) \cdot R_d(x)$ with $\deg R_d = 10 \cdot 6^{d-1}$, proved in full generality by induction and independently verified by exact-arithmetic polynomial division, together with an exact assembly formula for the full tree. Second, an eigenvalue-counting algorithm requiring only $O(k)$ recursive arithmetic steps – exponentially fewer than the $|K_k| \sim 6^k$ dimension of the matrix itself – built from the same recursive structure via the standard pivot-counting form of Sylvester’s law of inertia, verified to exact agreement against brute-force spectra for $k = 2, \dots, 6$ and used here to reach $k = 15$. Third, extending that same pivot to complex argument and identifying it exactly as the reciprocal of a diagonal resolvent entry, we find its real part numerically at $-x/2$ to $\sim 10^{-7}$ precision on the real axis, and, at finite $k = 15$, a macroscopic counting-function jump at a value consistent with a root of R_1 , backed by the exact divisibilities $R_1 \mid R_3$ and $R_1 \mid R_4$, though without yet a general sharing rule between generations or a proof that this persists as $k \rightarrow \infty$. Fourth, resolving the evidential core of that open question: working entirely inside the exact quotient ring $\mathbb{Z}[x]/(R_1(x))$ – which avoids the combinatorially infeasible full polynomials ($\deg R_6 = 77760$, with coefficients hundreds of digits long already at $d = 4$) – we exactly verify $R_1 \mid R_d$ for every tested $d = 3, \dots, 40$, and that the exact power of R_1 dividing R_d grows sharply: $e(3) = 5$, $e(4) = 30$, $e(5) = 180$, all exact (not extrapolated), matching a conjectured scaling law $e(d) = 5 \cdot 6^{d-3}$ at three points but not proved beyond them. Since R_1 is irreducible over $\mathbb{Q}$, its ten roots are Galois conjugates and so share, provably, the same exact multiplicity wherever R_1 divides – turning the assembly formula, conditional on that scaling law and a second conjectured law for the full tree, into a quantitative prediction for the mass at the single root $x \approx 0.469$: 1.3889% at $k = 15$, numerically close to the $\approx 1.4\%$ found empirically but conditional on both conjectural exponent laws. Both scaling laws remain open beyond the exact points checked.

Contents

1	Setup	2
2	An exact recursive factorization	3
3	An $O(k)$ -step eigenvalue counter	6
4	Finite-size spectral measure and a candidate limit	7
5	Open Problem	9
6	An exact reduction that avoids the full polynomials	9
7	R_1 divides (almost) every generation	10
8	The multiplicity is a tower, not a single copy	11
9	Quantitative consequence for the point mass	12
10	Consequence for the specific heat	13
11	Outlook	13
A	Verification code	14

1 Setup

We work throughout with the isotropic ($J_x = J_y = J_z = 1$) Majorana Hamiltonian [2] on the Fractal Kitaev graph K_k constructed in [1]: each original vertex of the hierarchical graph G_k is replaced by a trivalent gadget (a leaf for degree 1, a 5-proxy path P_1 - P_2 - P_3 - P_4 - P_5 for degree 7, a 4-vertex star for the degree-6 root: a center R_0 joined to three secondary hubs R_1, R_2, R_3 , each carrying two of the six original external edges), and gadgets are wired together exactly as the original tree edges dictate. Since K_k is a tree, it contains no plaquette cycles and hence no nontrivial loop-flux invariants; the $\mathbb{Z}_2$ link variables can therefore be gauge-fixed globally, and the physics reduces to a single real antisymmetric matrix A , oriented outward from the root, with $H = \frac{i}{4} \sum A_{uv} c_u c_v$.

Proposition 1.1 (Reduction to an ordinary tree spectrum). *$\text{spec}(iA) = \text{spec}(B)$ as multisets, where B is the ordinary (unsigned, symmetric) adjacency matrix of the same tree.*

Proof. Let U, V be the two color classes of the tree's bipartition (unique up to swap, since K_k is connected and acyclic), ordered so that $A = \begin{pmatrix} 0 & C \\ -C^T & 0 \end{pmatrix}$: every edge joins U to V , so A has no U - U or V - V entries, and C 's entries are ± 1 according to which of the two orientations each edge happens to carry – a genuine mix of signs in general, not the all-+1 pattern of $B = \begin{pmatrix} 0 & |C| \\ |C|^T & 0 \end{pmatrix}$. Since K_k is a tree (no cycles, hence no obstruction to gauge-fixing any sign pattern on its edges), there exist diagonal ± 1 matrices D_U, D_V with $D_U C D_V = |C|$: root the tree, set $D = +1$ at the root, and propagate outward,

flipping the sign at a vertex exactly when the edge to its parent carries a -1 in C ; this is well-defined precisely because there are no cycles to create a consistency conflict. As D_U, D_V are orthogonal, $\sigma_j(C) = \sigma_j(D_U C D_V) = \sigma_j(|C|)$ for every singular value. Now $iA = \begin{pmatrix} 0 & iC \\ -iC^T & 0 \end{pmatrix}$ is Hermitian of the standard chiral block form, with eigenvalues $\pm\sigma_j(C)$ (σ_j unchanged by the unit-modulus factor i); B has the identical block form with $|C|$ in place of iC , hence eigenvalues $\pm\sigma_j(|C|)$. Combining, $\text{spec}(iA) = \{\pm\sigma_j(C)\} = \{\pm\sigma_j(|C|)\} = \text{spec}(B)$; the last equality needs no separate " $\pm$ " since a bipartite graph's adjacency spectrum is already symmetric under negation. $\square$

Proposition 1.1 was checked directly, on two counts: first, for $k = 4$, the eigenvalues of the ordinary adjacency matrix and the singular values obtained from the antisymmetric route agree to machine precision; second, on a genuinely mixed-sign orientation (not aligned to the U, V bipartition, so C provably has both signs), $\sigma_j(C) = \sigma_j(|C|)$ was confirmed to hold exactly, ruling out the possibility that the argument secretly relied on a special choice of orientation. Everything below is stated for the ordinary tree adjacency matrix: for a weighted tree the characteristic polynomial is the corresponding weighted matching polynomial, with each edge weight entering squared [3], already used in [1] for its exact zero-mode count; since every weight here has modulus 1, the weights enter only through $(\pm 1)^2 = 1$ and the characteristic polynomial coincides with the ordinary (unweighted) matching polynomial used throughout below.

Definition 1.2. For $d \geq 0$, let $A_d(x)$ be the matching polynomial of an isolated depth- d gadget-7 subtree (a leaf, for $d = 0$), and let $A'_d(x)$ be the matching polynomial of the same subtree with its own root vertex P_1 deleted.

Equivalently, this depth- d subtree has $N_d = \deg A_d = 2 \cdot 6^d - 1$ vertices ($1, 11, 71, 431, \dots$ for $d = 0, 1, 2, 3, \dots$): each of the six pendant ports of the gadget-7 (Section 1) not used to attach to a parent instead attaches a depth- $(d - 1)$ copy of the same subtree, recursively, down to the single-vertex leaf at $d = 0$.

2 An exact recursive factorization

Write $N_d := \deg A_d$. Attaching one more pendant depth- $(d - 1)$ subtree to a chain continuing as (X, X') is governed, by the standard rooted-tree matching-polynomial recursion, by

$$\text{next}(X, X') := (xA_{d-1}X - A'_{d-1}X - A_{d-1}X', A_{d-1}X).$$

The gadget's own five proxies, eliminated from P_5 back to P_1 , are built from this operation in sequence: P_5 carries two pendant depth- $(d - 1)$ subtrees, giving the seed pair

$$(E_d, E'_d) := (xA_{d-1}^2 - 2A_{d-1}A'_{d-1}, A_{d-1}^2),$$

and then $(D_d, D'_d) = \text{next}(E_d, E'_d)$, $(C_d, C'_d) = \text{next}(D_d, D'_d)$, $(B_d, B'_d) = \text{next}(C_d, C'_d)$, $(A_d, A'_d) = \text{next}(B_d, B'_d)$.

Theorem 2.1 (Exact generational factorization). For every $d \geq 1$,

$$A_d(x) = A_{d-1}(x) \cdot R_d(x), \quad \deg R_d = 10 \cdot 6^{d-1},$$

where R_d is obtained by applying $\text{next}(\cdot, \cdot)$ four times to the seed $(xA_{d-1} - 2A'_{d-1}, A_{d-1})$.

Proof. Preservation lemma. Suppose $(X, X') = (A_{d-1}\tilde{x}, A_{d-1}\tilde{x}')$ for some polynomials $\tilde{x}, \tilde{x}'$. Then the first component of $\text{next}(X, X')$ is

$$xA_{d-1}(A_{d-1}\tilde{x}) - A'_{d-1}(A_{d-1}\tilde{x}) - A_{d-1}(A_{d-1}\tilde{x}') = A_{d-1} \cdot (xA_{d-1}\tilde{x} - A'_{d-1}\tilde{x} - A_{d-1}\tilde{x}'),$$

and the second is $A_{d-1}(A_{d-1}\tilde{x}) = A_{d-1} \cdot (A_{d-1}\tilde{x})$; so $\text{next}(X, X') = A_{d-1} \cdot (\tilde{y}, \tilde{y}')$, where

$$(\tilde{y}, \tilde{y}') := (xA_{d-1}\tilde{x} - A'_{d-1}\tilde{x} - A_{d-1}\tilde{x}', A_{d-1}\tilde{x}) = \text{next}(\tilde{x}, \tilde{x}')$$

– the identical operation, applied to the reduced pair. So the property “both components carry a factor of A_{d-1} ” is preserved by next , and the reduced pair evolves under the same recursion.

The seed pair already has this form: $E_d = A_{d-1} \cdot (xA_{d-1} - 2A'_{d-1})$ and $E'_d = A_{d-1} \cdot A_{d-1}$, i.e. $(\tilde{x}_0, \tilde{x}'_0) = (xA_{d-1} - 2A'_{d-1}, A_{d-1})$. Applying the lemma four times (through D, C, B, A) gives $A_d = A_{d-1} \cdot a_4$, where a_4 is the result of applying next four times to $(\tilde{x}_0, \tilde{x}'_0)$ – this is exactly R_d as defined above.

Degree. An invariant makes this airtight: whenever $\deg X' = \deg X - 1$, the leading term of the numerator in $\text{next}(X, X')$ is unambiguously $xA_{d-1}X$, since $A'_{d-1}X$ has degree $\leq \deg A_{d-1} + \deg X - 1$ and $A_{d-1}X'$ has degree $\deg A_{d-1} + \deg X - 1$, both strictly below $\deg(xA_{d-1}X) = 1 + \deg A_{d-1} + \deg X$; and the new pair again satisfies $\deg(A_{d-1}X) = \deg X - 1 + \deg A_{d-1} = (1 + \deg A_{d-1} + \deg X) - 1$, so no cancellation of the leading term is possible at any step. The seed $(\tilde{x}_0, \tilde{x}'_0) = (xA_{d-1} - 2A'_{d-1}, A_{d-1})$ starts the invariant ($\deg \tilde{x}'_0 = \deg A_{d-1} = \deg \tilde{x}_0 - 1$, since $\deg A'_{d-1} < \deg A_{d-1}$), so each application of next raises degree by exactly $1 + \deg A_{d-1}$. By the induction hypothesis $N_{d-1} = 2 \cdot 6^{d-1} - 1$ (the base case $N_0 = 1$ is immediate, and the inductive step is completed at the end of this proof), the seed itself has $\deg(\tilde{x}_0) = 1 + \deg A_{d-1} = N_{d-1} + 1$; four further applications each add that same amount $N_{d-1} + 1$, giving

$$\deg R_d = 5(N_{d-1} + 1) = 5 \cdot 2 \cdot 6^{d-1} = 10 \cdot 6^{d-1},$$

and correspondingly $N_d = N_{d-1} + 10 \cdot 6^{d-1} = 2 \cdot 6^d - 1$, consistent with $N_0 = 1$. $\square$

Corollary 2.2 (Assembly for the full tree). *Writing $d = k - 2$ for the depth of the subtrees hanging from the root’s gadget, and*

$$Y_d(x) := xA_d(x)^2 - 2A_d(x)A'_d(x), \quad Y'_d(x) := A_d(x)^2$$

for the matching-polynomial pair rooted at one secondary hub R_i together with its two attached depth- d subtrees (the same seed formula as (E_d, E'_d) above, since R_i carries two pendants exactly as P_5 does),

$$\text{Total}_{K_k}(x) = Y_d(x)^2 \cdot [xY_d(x) - 3Y'_d(x)].$$

Proof. R_0 joins three identical copies of the branch rooted at R_i ; the same “attach three identical branches to a new root” recursion used implicitly throughout (a single application of the $m = 3$ case of the underlying matching-polynomial identity) gives the stated formula directly from (Y_d, Y'_d) , exactly as (A_d, A'_d) is built from (A_{d-1}, A'_{d-1}) one gadget-proxy at a time above. $\square$

Example 2.3 (Verification). *Theorem 2.1 was checked with exact (arbitrary-precision rational) arithmetic, not floating point: for $d = 1, 2, 3$, dividing A_d by A_{d-1} leaves zero remainder, and re-multiplying the quotient by A_{d-1} reconstructs A_d coefficient-for-coefficient*

(Listing 1). Independently, the general formula for R_d derived in the proof above – built from the reduced seed, not from A_{d-1} directly – was checked to equal the true quotient exactly for $d = 1, 2, 3$. Corollary 2.2 was checked by explicitly constructing K_k vertex-by-vertex from the gadget rules for $k = 2, 3$ and comparing its true characteristic polynomial (via the adjacency matrix, independent of the matching-polynomial machinery) against the formula: exact agreement, coefficient-for-coefficient, for both $k = 2$ (10 vertices) and $k = 3$ (70 vertices). An earlier version of this corollary, missing the Y_d step (i.e. using $A_d^2[xA_d - 3A_d']$ directly), gives the wrong degree (4 and 34 against the true 10 and 70) – caught by this same direct-construction check, which a floating-point eigenvalue comparison at $k = 2$ had previously failed to catch.

```

1 from fractions import Fraction
2
3 def pmul(a,b):
4     r=[0]*(len(a)+len(b)-1)
5     for i,ai in enumerate(a):
6         if ai:
7             for j,bj in enumerate(b): r[i+j]+=ai*bj
8     return r
9 def psub(a,b):
10    n=max(len(a),len(b)); a=[0]*(n-len(a))+list(a); b=[0]*(n-len(b))+list(b)
11    return [p-q for p,q in zip(a,b)]
12 def pscale(a,s): return [s*c for c in a]
13 def xmul(a): return list(a)+[0]
14
15 def nxt(Ad1,Apd1,X,Xp):
16     Y = psub(psub(xmul(pmul(Ad1,X)), pmul(Apd1,X)), pmul(Ad1,Xp))
17     Yp = pmul(Ad1,X)
18     return Y, Yp
19
20 def exact_divide(num,den):
21     num=[Fraction(c) for c in num]; den=[Fraction(c) for c in den]
22     q=[Fraction(0)]*(len(num)-len(den)+1); rem=num[:]
23     for i in range(len(q)):
24         c=rem[i]/den[0]; q[i]=c
25         for j in range(len(den)): rem[i+j]-=c*den[j]
26     return q, rem
27
28 A={0:[1,0]}; Ap={0:[1]}
29 def full_step(d):
30     Ad1,Apd1=A[d-1],Ap[d-1]
31     E,Ep = nxt(Ad1,Apd1,Ad1,Apd1)
32     D,Dp = nxt(Ad1,Apd1,E,Ep)
33     C,Cp = nxt(Ad1,Apd1,D,Dp)
34     B,Bp = nxt(Ad1,Apd1,C,Cp)
35     return nxt(Ad1,Apd1,B,Bp)
36
37 def claimed_Rd(d):
38     Ad1,Apd1=A[d-1],Ap[d-1]
39     seed = (psub(xmul(Ad1), pscale(Apd1,2)), Ad1[:])
40     for _ in range(4):
41         seed = nxt(Ad1, Apd1, *seed)
42     return seed[0]
43
44 for d in (1,2,3):
45     A[d],Ap[d] = full_step(d)

```

```

46     q_true, rem = exact_divide(A[d], A[d-1])
47     assert all(c==0 for c in rem)
48     R_claim = claimed_Rd(d)
49     assert [Fraction(c) for c in R_claim] == q_true
50     print("Theorem 2.1 verified exactly for d=1,2,3.")

```

Listing 1: Exact-arithmetic verification of Theorem 2.1 (the general R_d formula against the true quotient, for $d = 1, 2, 3$).

3 An $O(k)$ -step eigenvalue counter

Theorem 2.1 makes the polynomial’s exact structure explicit, but extracting individual roots from an explicit degree- $(2 \cdot 6^d - 1)$ polynomial by generic numerical methods fails well before $d = 3$: coefficients span too many orders of magnitude for companion-matrix eigensolvers, which is a coefficient-conditioning problem, not a flaw in the factorization (confirmed by re-multiplication in exact arithmetic, Example 2.3). The same recursive structure, used differently, avoids this entirely, via the standard pivot/Sturm-sequence approach to eigenvalue counting on trees [4].

Proposition 3.1 (Pivot recursion). *For $x \in \mathbb{R}$ not an eigenvalue of any A_j encountered in the recursion (so that no pivot below is exactly zero), define $\pi_0(x) = -x$, and recursively, writing $g = \pi_{d-1}(x)$,*

$$\begin{aligned}
\pi_{P_5} &= -x - \frac{2}{g}, & \pi_{P_4} &= -x - \frac{1}{g} - \frac{1}{\pi_{P_5}}, & \pi_{P_3} &= -x - \frac{1}{g} - \frac{1}{\pi_{P_4}}, \\
\pi_{P_2} &= -x - \frac{1}{g} - \frac{1}{\pi_{P_3}}, & \pi_d(x) &= -x - \frac{1}{g} - \frac{1}{\pi_{P_2}}.
\end{aligned}$$

Then, by Sylvester’s law of inertia applied to Gaussian elimination of $A_d - xI$ from the leaves inward, the number of eigenvalues of A_d strictly below x equals the count of negative values among $\{\pi_{P_5}, \pi_{P_4}, \pi_{P_3}, \pi_{P_2}, \pi_d\}$ at every recursion level, accumulated as

$$\nu_d(x) = 6 \nu_{d-1}(x) + \#\{\text{negative pivots among the five, at level } d\},$$

with base case $\nu_0(x) = \mathbb{K}[\pi_0(x) < 0]$. The same elimination, assembled through the root’s gadget-6 (three secondary hubs, each with two depth- $(k-2)$ pendants, per Corollary 2.2), gives $\nu(x; k)$, the count for the full tree K_k , in $O(k)$ recursive arithmetic steps – independent of, and exponentially fewer than, $|K_k| \sim 6^k$ itself, since $k \sim \log |K_k|$; Corollary 3.2 spells this final assembly out explicitly.

Corollary 3.2 (Root assembly, explicit). *Writing $g := \pi_{k-2}(x)$ for the pivot of one depth- $(k-2)$ subtree (Proposition 3.1), the pivot at each secondary hub is $\pi_{R_i} = -x - 2/g$ ($i = 1, 2, 3$, identical by symmetry), and the pivot at the center is $\pi_{R_0} = -x - 3/\pi_{R_1}$. Accumulating negative-pivot counts through all four eliminated sites,*

$$\nu(x; k) = 6 \nu_{k-2}(x) + \#\{i : \pi_{R_i}(x) < 0\} + \mathbb{K}[\pi_{R_0}(x) < 0],$$

which is the $O(k)$ -recursive-step count of Proposition 3.1 claimed above, made fully explicit rather than left as "assembled through the root’s gadget".

[DERIVED] Checked directly against true Gaussian elimination on the full 70-vertex K_3 ($k = 3$, so $d = k - 2 = 1$): $\pi_{R_0}(x)$ from the formula above agrees with the true root

pivot (computed by direct leaves-inward elimination on the actual adjacency matrix) to 8 significant digits, at three tested x .

For $x > 0$ the recursion is ill-conditioned near the massive zero-eigenvalue degeneracy; this is sidestepped completely using the symmetry of Proposition 1.1: for $x \notin \text{spec}(A_k)$,

$$\nu(x; k) = N_k^{\text{spins}} - \nu(-x; k).$$

Example 3.3 (Verification and reach). *Proposition 3.1 was checked against every brute-force spectrum available ($k = 2, \dots, 6$, 25 test points each): zero discrepancy, after the symmetrization above (which alone fixes a systematic error, itself a useful cautionary note: naive pole handling exactly at the massive $x = 0$ degeneracy silently overcounts by exactly a factor tied to that degeneracy if not corrected this way). Being $O(k)$ per query, it reaches $k = 15$ ($N_{15}^{\text{spins}} = 12 \cdot 6^{13} - 2 \approx 1.567 \times 10^{11}$, matching Corollary 2.2’s corrected degree count exactly) in seconds, twelve k -values past the previous brute-force ceiling.*

4 Finite-size spectral measure and a candidate limit

Let $F(x; k) := \nu(x; k)/N_k^{\text{spins}}$.

Numerical Result 4.1 (Numerical stabilization of the integrated density). *Writing $F_k(x) := F(x; k)$, $F_k(x) \approx F_{15}(x)$ to six decimal digits for $k = 9, 12, 15$, at every tested $x \in (0, 2.75)$.*

[REPORTED] This is a direct numerical finding (Proposition 3.1 evaluated at $k = 6, 9, 12, 15$; stabilization checked at finitely many tested x , not proved for all x). It is consistent with – but does not by itself prove – convergence of F_k to a fixed limiting distribution $F_\infty(x)$, the standard notion of a limiting spectral measure for a recursively-defined infinite tree operator. F_∞ is used below as a candidate object motivated by this stabilization, not one this note constructs; every claim that follows is really a claim about F_{15} (or the specific R_d ’s), with F_∞ named only where the claim is explicitly conditional on that candidate’s existence.

To probe F_∞ ’s shape directly rather than as a $k \rightarrow \infty$ extrapolation, extend Proposition 3.1’s pivot to complex argument, $\pi_d(z)$ for $z = x + i\eta$, by the identical recursion – the same self-consistent, cavity-method idea used for tree-like operators in [5]. This is a Schur-complement pivot, not itself a Green’s function – an earlier version of this note conflated the two, with an algebraically incorrect claim about which trace π_d equals. The correct relationship is:

Lemma 4.2 (Pivot–resolvent identity). $\pi_d(z) = 1/[(A_d - zI)^{-1}]_{\text{root}, \text{root}}$ for every z off the real spectrum of A_d : π_d is the reciprocal of the diagonal resolvent entry at the recursion’s root site, not the entry itself.

Proof. Standard Schur-complement fact: for a symmetric matrix M partitioned with the root as a single block, the root’s Schur complement after eliminating every other row and column equals $1/[M^{-1}]_{\text{root}, \text{root}}$; Proposition 3.1’s π_d is exactly this Schur complement for $M = A_d - zI$, built up one gadget-proxy at a time. $\square$

[DERIVED] (the general Schur-complement fact is standard; the identification with Proposition 3.1’s specific recursion was checked directly for $d = 1$, comparing $\pi_1(x)$ against $1/[(A_1 - xI)^{-1}]_{11}$ from explicit 11×11 matrix inversion, at four values of x : agreement to

8 significant digits.) We fix the convention $g_d(z) := 1/\pi_d(z) = [(A_d - zI)^{-1}]_{\text{root},\text{root}}$ for the local Green's function itself; this differs by an overall minus sign from the perhaps more familiar $G_{\text{std}}(z) := [(zI - A_d)^{-1}]_{\text{root},\text{root}} = -g_d(z)$. The eigenvalue-counting content of Proposition 3.1 (Sylvester's law) uses only π_d directly and needs no resolvent interpretation at all; what follows is additionally about π_d 's limiting behavior, not a restatement of the counting result.

Remark 4.3 (Local root measure versus the global density). *[DERIVED] π_d (and hence $g_d = 1/\pi_d$) is built entirely from the gadget's own root vertex – a single, structurally distinguished site, not a generic one, on a tree that is not vertex-transitive (leaves alone already outnumber the five internal proxies at every generation, so a “typical” vertex is overwhelmingly leaf-like, unlike the root). The object that actually controls F_∞ 's density is instead the trace-averaged resolvent $G_d(z) := \frac{1}{N_d} \sum_v [(A_d - zI)^{-1}]_{vv}$, and $g_d \neq G_d$ in general: checked directly here, at $z = 0.5 + 0.05i$ for $d = 1, 2, 3$, the two already disagree in real part, not merely by a small correction. Sylvester's law of inertia (Section 3) needs none of this – it counts eigenvalues of the whole matrix regardless of which vertex is eliminated last, so $\nu(x; k)$, and hence F_k itself, are already the correct global quantities throughout this note. The gap below is specifically about using π_d/g_d for anything beyond the counting result, taken up in Open Problem 4.5.*

Numerical Result 4.4 (Real part of the limiting pivot). $\Re \pi_\infty(x + i0^+) \approx -x/2$ to $\sim 10^{-7}$, across nine tested values of x between 0.1 and 2.75, where $\pi_\infty(z) := \lim_{d \rightarrow \infty} \pi_d(z)$ when that limit exists – observed numerically to stabilize to machine precision within ~ 20 recursion levels at every tested z , not established as an existing limit in general (limited by the regulator η on the real axis, not by the iteration itself – the same recursion converges cleanly when η is fixed at a genuinely nonzero value instead, e.g. at a Matsubara-type point $z = i\omega$, $\omega > 0$).

[REPORTED] Exactness of $-x/2$ is a numerical observation about π_∞ , not (yet) a theorem, and not by itself a statement about $g_\infty = 1/\pi_\infty$ or about F_∞ 's density: since $g = 1/\pi$ is a nonlinear function of π , $\Re(\pi) = -x/2$ does not imply any correspondingly simple form for $\Re(g)$ without also knowing $\Im(\pi)$. An earlier version of this note compared $\Im(\pi)$ against a semicircle law of radius $2\sqrt{2}$ and reported a non-constant ratio to the numerically-obtained F_{15} as evidence against a purely continuous, semicircular measure. That comparison's translation into a statement about the true density (which is governed by $\Im(g)$, via Lemma 4.2) has not been carried out and is retracted here rather than repeated uncritically; Open Problem 4.5 records it as open. Numerical Result 4.6 below does not depend on it: it is an independent, purely combinatorial argument about polynomial root multiplicity, not about π or g at all.

Open Problem 4.5 (Translate the pivot results into resolvent/density language). $g_\infty = 1/\pi_\infty$ is the root's own local Green's function (Remark 4.3), not automatically F_∞ 's Stieltjes transform: the needed first step is to establish, or refute, that the root spectral measure and the trace-averaged measure G_∞ coincide in this limit – true for free on a vertex-transitive graph, not shown here for K_k – before $\Re(g_\infty), \Im(g_\infty)$ can be read as F_∞ 's density off the atoms. Absent that bridge, the more direct route is to test the old semicircle-law comparison against F_{15} 's own derivative from the counting function of Proposition 3.1, which needs neither g nor π , hence no such bridge. Neither is attempted here.

Numerical Result 4.6 (A candidate point mass). F_{15} has a jump of ≈ 0.014 within an interval of width 10^{-3} around $x \approx 0.46942$.

[REPORTED] This interval contains the smallest positive root of R_1 , $r_1 = 0.469459409462421 \dots$ (Theorem 2.1, $d = 1$), to within the stated numerical resolution; we say the jump is *consistent with* r_1 , not that its location is exactly r_1 , since the jump itself was only ever resolved to that 10^{-3} window. Theorem 2.1 guarantees r_1 , once a root of A_{d-1} , is inherited unchanged into $A_d, A_{d+1}, \dots$; a macroscopic (not merely $O(1)$) point mass at r_1 requires more than reappearing in infinitely many later $R_{d'}$'s – reappearing with multiplicity 1 at each of $d' = 1, \dots, d$ still only gives $O(d)$ total multiplicity against $N_d \sim 6^d$, i.e. a vanishing fraction; what is actually needed is positive *asymptotic multiplicity density*, $T(d)/N_d \not\rightarrow 0$. Direct computation confirms the mechanism is active at least twice more: $\gcd(R_1, R_3) = \deg R_1$ and $\gcd(R_1, R_4) = \deg R_1$, i.e. R_1 divides both completely. Two further data points cannot establish a nonvanishing density; Sections 7 and 8 below establish exactly how far this mechanism extends, by a route independent of Lemma 4.2 and Open Problem 4.5: direct polynomial-root multiplicity, not resolvent asymptotics.

5 Open Problem

Open Problem 5.1 (The sharing rule between generations). *Table 1 lists $\deg \gcd(R_d, R_{d'})$ for all pairs with $d, d' \leq 4$.*

pair	$\deg \gcd$	$\deg(\text{smaller } R)$	sharing
R_1, R_2	0	10	none
R_1, R_3	10	10	complete
R_1, R_4	10	10	complete
R_2, R_3	6	60	partial
R_2, R_4	60	60	complete
R_3, R_4	80	360	partial

Table 1: Exact-arithmetic GCDs among the first four generational factors.

A period-2 pattern (matching the characteristic roots $6, -1$ of the matching-number recursion $\mu(k) = 5\mu(k-1) + 6\mu(k-2) + 22$ already used in [1]) might suggest, as one possible heuristic, complete sharing at even index-differences and none at odd ones. $R_1 \mid R_4$ (difference 3, odd, complete) already rules out that specific heuristic, without yet supplying a replacement. No general rule is yet known for $\deg \gcd(R_d, R_{d'})$, and hence neither is the total point-mass fraction of F_∞ , nor whether the point-mass locations are dense in the support. Resolving this is the natural next step; Sections 6 to 9 below take it up directly, for the specific root at $x \approx 0.469$.

6 An exact reduction that avoids the full polynomials

Table 1's two data points ($d = 3, 4$) alone cannot distinguish a persistent, growing degeneracy from a coincidence: proving a macroscopic atom requires the multiplicity of the relevant root inside A_d to grow proportionally to $N_d = \deg A_d = 2 \cdot 6^d - 1$, and two data points cannot show a growth rate. What is needed is that multiplicity for d well beyond 4, without paying the combinatorial cost of the full polynomials – direct computation is hopeless past $d = 4$: $\deg R_5 = 12960$, $\deg R_6 = 77760$, and the coefficients are already

hundreds of digits long at $d = 4$. The following observation removes the obstruction entirely for the specific question of divisibility by R_1 .

Lemma 6.1 (Validity of exact reduction modulo R_1). *Let $M \in \mathbb{Z}[x]$ be monic. Polynomial division by M never requires inverting a leading coefficient, so the remainder map $\rho_M : \mathbb{Z}[x] \rightarrow \mathbb{Z}[x]/(M)$ stays inside $\mathbb{Z}[x]$ and is a ring homomorphism. The operations defining R_d in Theorem 2.1 – the map $\text{next}(A_{d-1}, A'_{d-1}; X, X')$ and its four-fold composition on the seed $(xA_{d-1} - 2A'_{d-1}, A_{d-1})$ – use only addition, subtraction, and multiplication; no division appears anywhere in this chain. Consequently, for any monic M ,*

$$\rho_M(R_d) = \Phi_M\left(\rho_M(A_0), \rho_M(A'_0), \dots, \rho_M(A_{d-1}), \rho_M(A'_{d-1})\right),$$

where Φ_M denotes the same recursive construction carried out entirely inside $\mathbb{Z}[x]/(M)$, reducing after every multiplication. In particular, taking $M = R_1$ (monic, degree 10; Equation (1)), every intermediate polynomial in the construction of $R_d \bmod R_1$ can be kept at degree ≤ 9 for every d , with no reference to the true, exponentially large A_d . The same reduction applies verbatim to Corollary 2.2's Y_d, Y'_d and to Total_{K_k} , since these too are built from A_d, A'_d by $+, -, \times$ alone.

Proof. Immediate from $\rho_M(p \pm q) = \rho_M(p) \pm \rho_M(q)$ and $\rho_M(pq) = \rho_M(p)\rho_M(q)$, applied inductively to each of the finitely many $+, -, \times$ operations in the construction, together with the fact (checked directly on the coefficient list) that R_1 is monic. $\square$

$$R_1(x) = x^{10} - 10x^8 + 32x^6 - 38x^4 + 16x^2 - 2. \quad (1)$$

[DERIVED] (Equation (1) itself: exact integer arithmetic on the recursion of Theorem 2.1, Listing 2). R_1 is squarefree and irreducible over $\mathbb{Q}$ – exact factorization over $\mathbb{Q}$, using SymPy 1.14.0 (`sympy.Poly(R1, x).is_irreducible`, Listing 2), returns `True`; this is a decision procedure, not a numerical approximation. As an independent, software-light certificate: $R_1 \bmod 5 = x^{10} + 2x^6 + 2x^4 + x^2 - 2$ is irreducible over $\mathbb{F}_5$ with the same leading coefficient (1, so degree is preserved under reduction), and irreducibility mod a prime with unchanged degree implies irreducibility over $\mathbb{Q}$ (Listing 2). R_1 has 10 distinct real roots, 5 positive: 0.469459..., 0.680380..., 1.137453..., 1.744120..., 2.231801... (found by standard numerical root-finding on the now-exactly-known degree-10 polynomial) – the first matches the location of Numerical Result 4.6 to within its stated 10^{-3} resolution window.

Listing 2 implements Φ_{R_1} and, for any target power p , $\Phi_{R_1^p}$ in the same way – the lemma places no restriction on M beyond monicity, and R_1^p is monic whenever R_1 is.

7 R_1 divides (almost) every generation

Proposition 7.1 (Persistence of the base divisibility). *$R_1 \mid R_d$ for every $d \in \{3, 4, 5, \dots, 40\}$ tested. The sole exception below $d = 40$ is $d = 2$, matching $\deg \gcd(R_1, R_2) = 0$ already reported in Table 1.*

[DERIVED] Verified by exact computation in $\mathbb{Z}[x]/(R_1)$ (Listing 2); the check for each d costs a handful of degree- ≤ 18 polynomial multiplications, regardless of d . This already extends Table 1's two data points ($d = 3, 4$) to thirty-eight – a finite, if long, list, not a proof for all d .

8 The multiplicity is a tower, not a single copy

Persistence of $R_1 \mid R_d$ alone is not enough: if each generation contributed exactly one extra copy of R_1 's ten roots, the cumulative multiplicity in A_d would grow only linearly in d , i.e. vanishingly relative to $N_d \sim 6^d$. What actually matters is the *exponent*, not the yes/no fact. We therefore computed, for each d , the largest p such that $R_1^p \mid R_d$, using $\Phi_{R_1^p}$ from Lemma 6.1.

Proposition 8.1 (Exact exponents, $d = 3, 4, 5$). *Writing $e(d)$ for the largest p with $R_1^p \mid R_d$:*

$$e(3) = 5, \quad e(4) = 30, \quad e(5) = 180.$$

[DERIVED] Each value certified by a matching divides/does-not-divide pair, leaving no room for an "extrapolated from a scan" reading:

$$R_1^5 \mid R_3, \quad R_1^6 \nmid R_3; \quad R_1^{30} \mid R_4, \quad R_1^{31} \nmid R_4; \quad R_1^{180} \mid R_5, \quad R_1^{181} \nmid R_5,$$

the first two by exhaustive scan and the third by two direct checks (Listing 2).

Lemma 8.2 (Galois-conjugate roots share multiplicity). *Let $P \in \mathbb{Q}[x]$ be nonzero and let $M \in \mathbb{Q}[x]$ be irreducible over $\mathbb{Q}$. Then every root of M has the same multiplicity in P .*

Proof. M irreducible means M is the minimal polynomial over $\mathbb{Q}$ of each of its roots, so any automorphism of $\mathbb{Q}$ fixing $\mathbb{Q}$ pointwise permutes the roots of M , and this action is transitive (standard fact for irreducible polynomials). Such an automorphism σ fixes P coefficientwise (as $P \in \mathbb{Q}[x]$), so for a root r of M , $\text{mult}_r(P) = \text{mult}_{\sigma(r)}(\sigma(P)) = \text{mult}_{\sigma(r)}(P)$. Transitivity gives the claim. $\square$

Since R_1 is irreducible over $\mathbb{Q}$ (Section 6), Lemma 8.2 upgrades Proposition 8.1 from a statement about the block R_1 to a statement about the specific root $x \approx 0.469$ itself: its multiplicity in R_d is exactly $e(d)$, not merely "some share of" the ten roots' combined $10e(d)$.

d	$\deg R_d = 10 \cdot 6^{d-1}$	$e(d)$	$10e(d)/\deg R_d$
3	360	5 (exact)	$5/36 = 0.13889$
4	2160	30 (exact)	$5/36 = 0.13889$
5	12960	180 (exact)	$5/36 = 0.13889$

Table 2: Exact exponent of R_1 in each fresh generational factor. The last column is the fraction of $\deg R_d$ accounted for by all ten roots of R_1 together ($10e(d)$ roots out of $\deg R_d$); the fraction for the single root $x \approx 0.469$ alone is $e(d)/\deg R_d$, ten times smaller. The combined-roots ratio is *identical* at $d = 3, 4, 5$, not merely similar.

Conjecture 8.3 (Exponential exponent law).

$$e(d) = 5 \cdot 6^{d-3} \quad (d \geq 3).$$

[CONJECTURE] This reproduces $e(3) = 5$, $e(4) = 30$, and $e(5) = 180$ exactly – three confirmed points, not two. The base-6 growth is the natural guess: it is the same branching factor that already governs N_d , $\deg R_d$, and the graph G_k 's own 6-ary hierarchy in [1], so no new scale enters the problem under this conjecture. We have not found a proof for $d \geq 6$; Open Problem 9.5 records this as open.

9 Quantitative consequence for the point mass

Write $E(d) := \sum_{d'=1}^d e(d')$ for the cumulative exponent of R_1 in $A_d = A_0 \cdot R_1 \cdots R_d$ (a sum of exponents, since R_1 irreducible is prime in $\mathbb{Q}[x]$, so exponents add across the product); by Lemma 8.2 this is again the exact multiplicity of $x \approx 0.469$ itself, not just of the ten-root block. We need the corresponding exponent $T(d)$ for $x \approx 0.469$ inside $\text{Total}_{K_k} = Y_d^2[xY_d - 3Y_d']$ (Corollary 2.2); this does *not* reduce to a one-line combination of $E(d)$, because A'_d turns out to also carry R_1 once $d \geq 2$ (checked directly: $R_1 \nmid A'_1$ but $R_1 \mid A'_d$ for $d = 2, \dots, 10$), so we compute $T(d)$ directly rather than through an intermediate coprimality assumption.

Proposition 9.1 (Exact exponents in the assembled tree). *Writing $T(d)$ for the multiplicity of $x \approx 0.469$ in $\text{Total}_{K_k}(x)$ ($d = k - 2$):*

$$T(1) = 3, \quad T(2) = 6, \quad T(3) = 36, \quad T(4) = 216.$$

[DERIVED] Each computed directly in $\mathbb{Z}[x]/(R_1^p)$ (Lemma 6.1 applied to $Y_d, Y'_d, \text{Total}_{K_k}$, extending Listing 2): $T(1)$ by exhaustive scan on the true (unreduced) 10-vertex and 70-vertex assembled polynomials of Example 2.3; $T(2), T(3)$ by exhaustive scan in the quotient ring; $T(4)$ by directly confirming $R_1^{216} \mid \text{Total}$ and $R_1^{217} \nmid \text{Total}$.

Conjecture 9.2 (Exponent law for the assembled tree). *$T(d) = 6 E(d)$ for $d \geq 2$ (with the transitional value $T(1) = 3 E(1)$, reflecting that A'_1 does not yet carry R_1).*

[CONJECTURE] Exactly reproduces $T(2) = 6, T(3) = 36, T(4) = 216$ against $E(2) = 1, E(3) = 6, E(4) = 36$; not independently confirmed beyond $d = 4$.

Proposition 9.3 (Mass at $x \approx 0.469$, assuming Conjectures 8.3 and 9.2). *By Lemma 8.2, the mass at $x \approx 0.469$ in the normalized spectrum at $k = 15$ ($d = 13$) is exactly $T(13)/\deg(\text{Total}_{K_{15}})$. With $E(13) = 362,797,056$, $T(13) = 6 E(13) = 2,176,782,336$, and $\deg(\text{Total}_{K_{15}}) = 12 \cdot 6^{13} - 2 = 156,728,328,190$ (Corollary 2.2), this gives*

$$\frac{T(13)}{\deg(\text{Total}_{K_{15}})} = 1.3889\%,$$

against the $\approx 1.4\%$ of Numerical Result 4.6: numerically close, but conditional on both conjectural exponent laws, not a statistical accuracy claim.

[REPORTED] This combines two conjectures (Conjectures 8.3 and 9.2, each unproven beyond $d = 4$) but needs no equal-split assumption: Lemma 8.2 makes the per-root mass a proven consequence of R_1 's irreducibility, not an average over its roots. It should still be read as an order-of-magnitude-and-better consistency check, not a derivation of the 1.4% figure.

Remark 9.4. *This changes the character of the open question of Open Problem 5.1. That two data points ($d = 3, 4$) cannot distinguish a growing fraction from a coincidence was correct as a statement about Numerical Result 4.6 taken alone. Propositions 7.1, 8.1 and 9.1 show the underlying divisibility is far more persistent, and the exponent in the assembled tree far larger, than a single-copy-per-generation picture would predict – consistent with, though not proof of, genuinely exponential growth. Whether it is exactly strong enough to produce the observed 1.4%, rather than merely the right order of magnitude, is exactly what Conjectures 8.3 and 9.2 would need to be proved to settle.*

Open Problem 9.5 (Prove the exponent laws). *The most tractable-looking route to Conjecture 8.3 is the single-step statement $e(d+1) = 6e(d)$: if proved for all $d \geq 3$, it would give $e(d) = 5 \cdot 6^{d-3}$ immediately by induction from $e(3) = 5$, and lives entirely inside $\text{next}(A_{d-1}, A'_{d-1}; \cdot, \cdot)$ acting on $\mathbb{Z}[x]/(R_1^p)$ for growing p – Lemma 6.1 makes each fixed- p instance computable, but a closed-form argument, not a scan, is needed to settle the R_1 -adic valuation shift for all d . Conjecture 9.2 is a harder second target, since it additionally requires tracking the R_1 -adic valuation of A'_d itself (found here to be 0 at $d = 1$ and positive for $d = 2, \dots, 10$, but not derived in closed form); we would attack it only after the first. Both are left open here, in the same spirit as Open Problem 5.1: reported as open rather than pushed to a guessed closed form.*

10 Consequence for the specific heat

The motivating question was the finite-temperature specific heat of the Schottky-pair spectrum,

$$C_v(T) = \sum_{a: \varepsilon_a > 0} \left(\frac{\varepsilon_a}{T} \right)^2 \frac{e^{\varepsilon_a/T}}{(1 + e^{\varepsilon_a/T})^2},$$

summed once over each of the $N_k^{\text{spins}}/2$ positive independent quasiparticle energies – one per particle-hole pair of Proposition 1.1, i.e. one per physical complex fermion mode, not once per signed eigenvalue x : the summand above is itself even in ε_a , so summing over the full $\pm$ -symmetric spectrum would silently double-count every mode. Here ε_a denotes the physical quasiparticle energy; whether it coincides with the positive adjacency eigenvalue x used throughout, or differs by a convention-dependent overall factor c beyond the $J = 1$ choice already absorbed into A in Section 1, has not been checked and is left open (Open Problem 11.2). C_v as written is extensive; the per-mode specific heat $c_v := C_v/(N_k^{\text{spins}}/2)$ is the quantity with a finite $k \rightarrow \infty$ limit, and is what should be compared across system sizes. Numerical Results 4.1 and 4.6 together suggest a precise, testable prediction that pure numerics on finite k could not: if the continuous part of F_∞ contributes a smooth background to $C_v(T)$, each point mass would contribute its own genuine Schottky peak at a *fixed* temperature that does not broaden or wash out with system size, in contrast to a peak built from finitely many discrete levels. Writing $y := \varepsilon_a/(2T)$, the single-mode term is $y^2 \text{sech}^2 y$, maximized at the root of $y \tanh y = 1$, $y^* \approx 1.19968$, i.e. at $T_{\max} = \varepsilon_a/(2y^*) \approx 0.4167 \varepsilon_a$ – for the mass identified in Numerical Result 4.6 ($x \approx 0.469$), $T_{\max} \approx 0.4167 c x \approx 0.196 c$, not $T \sim x$ itself. A finite- k calculation cannot distinguish the atom-vs-continuum mechanisms by shape alone; Sections 7 to 9 above are what make the point-mass reading the better-supported one, without yet proving it.

11 Outlook

Two extensions were suggested externally during this work and are recorded here, deliberately undeveloped, rather than folded into the results above.

Open Problem 11.1 (Matsubara reformulation). $\pi_\infty(x+i0^+)$ (Section 4) and a fermionic Matsubara pivot $\pi_\infty(i\omega_n)$, $\omega_n = (2n+1)\pi T$, would be the same function evaluated at different points, related to the actual Green's function $g_\infty = 1/\pi_\infty$ (Lemma 4.2) and, only once the root/average bridge of Open Problem 4.5 is resolved, to the Stieltjes transform of F_∞ ; the present note develops neither the Matsubara evaluation nor that bridge. Writing the

Matsubara sum for $U(T)$ and $C_v(T)$ directly from the pivot recursion of Proposition 3.1 evaluated at $z = i\omega_n$ (numerically well-behaved, since $i\omega_n$ is bounded away from the real support for every n) would give a second, independent computational route to Section 10 – but, on its own, no new evidence for or against Numerical Result 4.6: any such calculation must still assume, via the Lehmann representation, the atomic weight it would then reproduce in $C_v(T)$.

Open Problem 11.2 (Normalization and a possible superconducting extension). *Two conventions have not been pinned down and would need to be, before quantitative comparison to any physical realization: the overall prefactor relating the adjacency eigenvalues x used throughout to the quasiparticle energies ε_a of Section 10, and the sign convention for π_∞ and g_∞ (Section 4), which does not affect any claim above since only π_d itself is used, but would need fixing to compare against a specific many-body convention. Separately, whether a graphene-based or other artificial-lattice realization of K_k , or an added pairing interaction on top of the free Majorana Hamiltonian here, could exhibit a Cooper instability tied to the point mass of Numerical Result 4.6, is an interesting question raised externally during this work; it requires genuinely new physics (a pairing term absent from Section 1’s Hamiltonian) and new literature review, not attempted here.*

Acknowledgements

Developed with assistance from large-language-model tools (Claude, Anthropic; ChatGPT, OpenAI; and Gemini, Google, for the earlier construction of G_k in [1] that this note builds on) throughout, including the multiplicity test of Sections 6 to 9. Every mathematical and numerical claim reported here was checked either by exact arithmetic or by reproducible computation against the actual object being claimed about, not a proxy for it, except where explicitly marked otherwise: Open Problems 4.5, 5.1, 9.5, 11.1 and 11.2 remain open, and Conjectures 8.3 and 9.2 and proposition 9.3 remain conjectural or estimated.

References

- [1] F. Vassallo, *Zeta Functions on a Self-Similar Set: What $-1/5$ Is, What It Isn’t, and What Remains Open*, September 2026.
- [2] A. Kitaev, *Anyons in an exactly solved model and beyond*, Ann. Phys. **321**, 2–111 (2006).
- [3] C. D. Godsil, I. Gutman, *On the theory of the matching polynomial*, J. Graph Theory **5**, 137–144 (1981).
- [4] C. D. Godsil, *Algebraic Combinatorics*, Chapman and Hall (1993).
- [5] R. Abou-Chacra, D. J. Thouless, P. W. Anderson, *A selfconsistent theory of localization*, J. Phys. C: Solid State Phys. **6**, 1734 (1973).

A Verification code

```

1  # Polynomials are coefficient lists, highest degree first (numpy-style).
2  def pmul(a,b):
3      r=[0]*(len(a)+len(b)-1)
4      for i,ai in enumerate(a):
5          if ai:
6              for j,bj in enumerate(b): r[i+j]+=ai*bj
7      return r
8  def psub(a,b):
9      n=max(len(a),len(b)); a=[0]*(n-len(a))+list(a); b=[0]*(n-len(b))+list(b)
10     return [p-q for p,q in zip(a,b)]
11  def pscale(a,s): return [s*c for c in a]
12  def xmul(a): return list(a)+[0]
13  def strip(a):
14      i=0
15      while i < len(a)-1 and a[i]==0: i+=1
16      return a[i:]
17  def is_zero(p): return all(c==0 for c in p)
18
19  def rem_mod_monic(a, m):          # m must be monic; stays in Z, no division ever needed
20      a = strip(list(a)); dm = len(m)-1
21      while len(a)-1 >= dm and not is_zero(a):
22          coeff = a[0]; deg_diff = (len(a)-1) - dm
23          sub = [coeff*c for c in m] + [0]*deg_diff # append zeros: multiply by x^
24              deg_diff
25          a = strip(psub(a, sub))
26      return a
27
28  def ppow(base,n):
29      r=[1]; b=base[:]
30      while n>0:
31          if n&1: r=pmul(r,b)
32          b=pmul(b,b); n>>=1
33      return r
34
35  R1 = [1,0,-10,0,32,0,-38,0,16,0,-2]
36
37  def build_AAp_mod(MOD, dmax):    # A_d, A_d' reduced mod MOD, d=0..dmax
38      def nxt(Ad1,Apd1,X,Xp):
39          Y = rem_mod_monic(strip(psub(psub(xmul(pmul(Ad1,X)),pmul(Apd1,X)),pmul(Ad1,Xp)
40              )), MOD)
41          Yp = rem_mod_monic(strip(pmul(Ad1,X)), MOD)
42          return Y, Yp
43      A={0: strip([1,0])}; Ap={0: strip([1])}
44      for d in range(1,dmax+1):
45          Ad1,Apd1=A[d-1],Ap[d-1]
46          E,Ep=nxt(Ad1,Apd1,Ad1,Apd1); D,Dp=nxt(Ad1,Apd1,E,Ep)
47          C,Cp=nxt(Ad1,Apd1,D,Dp); B,Bp=nxt(Ad1,Apd1,C,Cp)
48          A[d],Ap[d]=nxt(Ad1,Apd1,B,Bp)
49      return A,Ap
50
51  def Rd_mod(MOD, d):              # R_d reduced mod MOD, via the seed shortcut (Sec.2)
52      A,Ap = build_AAp_mod(MOD, d-1)
53      Ad1,Apd1 = A[d-1],Ap[d-1]
54      def nxt(X,Xp):
55          Y = rem_mod_monic(strip(psub(psub(xmul(pmul(Ad1,X)),pmul(Apd1,X)),pmul(Ad1,Xp)
56              )), MOD)
57          Yp = rem_mod_monic(strip(pmul(Ad1,X)), MOD)

```

```

55     return Y, Yp
56     seed = (rem_mod_monic(strip(psub(xmul(Ad1),pscale(Apd1,2))),MOD), Ad1[:])
57     for _ in range(4): seed = nxt(*seed)
58     return seed[0]
59
60 print({d: is_zero(Rd_mod(R1,d)) for d in range(1,41)}) # Sec.7: True for d=3..40, d!=2
61
62 def exact_power(d, pmax):          # Sec.8: full scan reproducing e(3),e(4) exactly, e(5)
    bounds
63     e = 0
64     for p in range(1, pmax+1):
65         if is_zero(Rd_mod(ppow(R1,p), d)): e = p
66         else: break
67     return e
68 print("e(3) =", exact_power(3, 10))          # -> 5
69 print("e(4) =", exact_power(4, 40))          # -> 30
70 print("e(5) >=", exact_power(5, 50))          # -> 50 (still rising at the scan
    cap)
71 print("R1^180 | R_5:", is_zero(Rd_mod(ppow(R1,180), 5))) # -> True
72 print("R1^181 | R_5:", is_zero(Rd_mod(ppow(R1,181), 5))) # -> False => e(5) = 180
    exactly
73
74 def total_mod(MOD, d):              # corrected Total_{K-k} (Cor. 1.2), reduced mod MOD
75     A,Ap = build_AAp_mod(MOD, d)
76     Ad,Apd = A[d],Ap[d]
77     Ad2 = rem_mod_monic(strip(pmul(Ad,Ad)), MOD)
78     AAp = rem_mod_monic(strip(pmul(Ad,Apd)), MOD)
79     Y = rem_mod_monic(strip(psub(xmul(Ad2), pscale(AAp,2))), MOD)
80     Y2 = rem_mod_monic(strip(pmul(Y,Y)), MOD)
81     bracket = rem_mod_monic(strip(psub(xmul(Y), pscale(Ad2,3))), MOD)
82     return rem_mod_monic(strip(pmul(Y2,bracket)), MOD)
83
84 for d,p_ok,p_fail in [(1,3,4), (2,6,7), (3,36,37)]: # Sec.9: T(d) exact values
85     print(d, is_zero(total_mod(ppow(R1,p_ok),d)), not is_zero(total_mod(ppow(R1,p_fail)
        ,d)))
86
87 import sympy as sp                  # Sec.6: R1 irreducibility (sympy
    1.14.0)
88 xs = sp.symbols('x')
89 R1_sym = sum(c*xs**(10-i) for i,c in enumerate(R1))
90 print("R1 irreducible over Q:", sp.Poly(R1_sym, xs).is_irreducible)
91 print("R1 mod 5 irreducible, same degree:",
92     sp.Poly(R1_sym, xs, modulus=5).is_irreducible, sp.Poly(R1_sym, xs, modulus=5).LC
    ())

```

Listing 2: Exact reduction modulo powers of R_1 , staying in $\mathbb{Z}$ throughout: after each reduction, every stored polynomial has degree less than $10p$, regardless of d (intermediate pre-reduction products can be larger before `rem_mod_monic` brings them back down). Extended from previous versions to also build the (corrected) Total polynomial, its R_1 -exponent $T(d)$, and an irreducibility check for R_1 .