Counterexamples to log-concavity of local BPS invariants

Abstract

We disprove the conjecture of Su, Xie, and Yu (arXiv, 2026) that the local BPS invariants of every reduced plane curve singularity form a log-concave sequence. The ordinary 43-fold point $y^{43} - x^{43} = 0$ satisfies $n_{24}^2 < n_{23}n_{25}$, and the singularities $y^{36} - x^{36k} = 0$ satisfy $n_{21}^2 < n_{20}n_{22}$ for every integer $k \geq 10$. All BPS coefficients in these examples are strictly positive. The rainbow closures of the corresponding positive braids also give counterexamples to their conjecture for normalized ruling polynomials. Combining the Hilbert–HOMFLY and ruling identities with type-A Hecke character theory, we express the BPS polynomial of $y^r - x^{rk} = 0$ as the canonical trace of a full-twist power. For fixed r and h, the coefficient n_h is polynomial in $u = k(k+1)$, of degree at most h; its coefficient of u^h is the number of length-$2h$ transposition walks from the identity to itself in S_r, divided by $(2h)!$. These leading coefficients explain the eventual failure of log-concavity, while an exact polynomial sign certificate gives the explicit range $k \geq 10$. The finite integer and rational computations certifying both results are included in full.

Note. This paper was generated entirely by AI, including MiMo, using an automated research pipeline developed by Chenghua Liu and Hanyu Li.

2020 Mathematics Subject Classification. 14H20, 14N10, 20C08.

Keywords. Plane curve singularities; local BPS invariants; log-concavity; Hilbert schemes; Hecke algebras; full twists.

1 Introduction

Local BPS invariants organize the Euler characteristics of punctual Hilbert schemes into a finite genus expansion. This viewpoint is part of the broader relation between curve contributions and BPS counts: Pandharipande–Thomas [5] constructed such contributions in stable pair theory, while Shende [7, Definition 7 and Corollary 10] formulated the local expansion for plane curve singularities. The resulting integers connect local algebraic geometry to knot theory. Oblomkov–Shende [4] conjectured that the refined Hilbert-scheme series recovers the HOMFLY polynomial of the singularity link; Maulik [3] proved this correspondence. Rutherford's relation between HOMFLY coefficients and oriented rulings [6] then gives a combinatorial way to calculate the relevant specialization.

Su–Xie–Yu [8, Conjecture 1.1] conjectured that the local BPS sequence of every reduced plane curve singularity is log-concave and has no internal zeros. They prove the conjecture for irreducible weighted-homogeneous singularities and for ADE singularities. They also conjecture log-concavity for normalized ruling polynomials of rainbow closures of positive braids [8, Conjecture 2.3]. The restriction to this class of Legendrian links is important: Capovilla-Searle–Pan [1, Theorem 1.4] realize arbitrary polynomials in z^2 with nonnegative integer coefficients as graded ruling polynomials of Legendrian links, without imposing the positive-braid or algebraic-link hypotheses. This general realization result does not settle the restricted conjecture, as already noted in [8, Remark 2.5].

We construct counterexamples within the algebraic-link class. An ordinary multiple point already violates log-concavity, although every BPS coefficient is positive. Increasing the common contact order of the branches gives an infinite family. The uniform argument rests on a polynomial dependence on the contact order, not on extrapolation from finitely many examples.

Let $(C,0)$ be a reduced complex plane curve singularity, with δ-invariant δ and b branches. Let $\mathrm{Hilb}^\ell(C,0)$ denote the scheme of length-ℓ subschemes supported at 0, and let χ denote its topological Euler characteristic. The local expansion of [7, Definition 7 and Corollary 10], in the indexing convention of [8], defines the integers $n_h(C,0)$ by

$$q^{-\delta}(1-q)^b \sum_{\ell\geq 0} \chi\big(\mathrm{Hilb}^\ell(C,0)\big) q^\ell = \sum_{h=0}^{\delta} n_h(C,0) w^h, \qquad w = (q^{1/2}-q^{-1/2})^2. \tag{1}$$

In this normalization, log-concavity means $n_h^2 \geq n_{h-1}n_{h+1}$ for $1 \leq h \leq \delta-1$.

We study the reduced weighted-homogeneous singularities

$$C_{r,k} = \{y^r - x^{rk} = 0\}, \qquad r,k \in \mathbf{Z}, \quad r \geq 2, \quad k \geq 1. \tag{2}$$

Their r smooth branches $y = \zeta x^k$, $\zeta^r = 1$, have pairwise intersection multiplicity k. Thus the additivity formula for δ and the plane-curve Milnor formula give, with $N = \binom{r}{2}$ and $d = kN$,

$$b = r, \qquad \delta = d, \qquad \mu = 2d - r + 1. \tag{3}$$

Write $B_{r,k}(w) = \sum_{h=0}^{d} n_h(C_{r,k}) w^h$.

Theorem 1.1. *For the ordinary 43-fold point $C_{43,1}$, the coefficients*

$$\begin{aligned} n_{23} &= 467438304194650004605257498534262968215115139192 67, \\ n_{24} &= 447870695400119839679788896911479468288995454466029 5, \\ n_{25} &= 429147501803199891064835526788080765663004079176388506 \end{aligned}$$

satisfy

$$20000\, n_{24}^2 < 19999\, n_{23} n_{25}. \tag{4}$$

In particular, $n_{24}^2 < n_{23}n_{25}$. All 904 coefficients $n_0, \ldots, n_{903}$ are strictly positive.

Theorem 1.2. *For every integer $k \geq 10$, the singularity $C_{36,k}$ satisfies*

$$n_{21}(C_{36,k})^2 < n_{20}(C_{36,k})\, n_{22}(C_{36,k}).$$

All its BPS coefficients are strictly positive.

Theorem 1.1 disproves Conjecture 1.1 of [8], and Theorem 1.2 gives counterexamples of infinitely many embedded topological types, since $\delta(C_{36,k}) = 630k$. The link of $C_{r,k}$ is the r-component torus link $T(r, rk)$; thus the coprimality hypothesis in the torus-knot result of [8, Theorem 5.1] is essential. Strict positivity throughout these examples leaves the absence-of-internal-zeros question intact. The remaining irreducible cases and the tame character-variety conjecture of [8] are also outside the scope of these counterexamples.

The same construction applies to normalized ruling polynomials. Equation (8) identifies the BPS polynomial of $C_{43,1}$ with the normalized ruling polynomial of the rainbow closure of $(\sigma_1 \cdots \sigma_{42})^{43}$. Its coefficient sequence therefore also disproves Conjecture 2.3 of [8].

The established Hilbert–HOMFLY correspondence and braid-walk formula identify $B_{r,k}$ with $\tau(\Omega^{k+1})$, where Ω is the full twist and τ is the canonical Hecke trace. The shift from k to $k+1$ comes from the trace dual basis. The standard character expansion then gives a hook-length formula for the Hilbert series. Pairing conjugate partitions shows that each fixed BPS coefficient is polynomial in $u = k(k+1)$; the leading coefficients are normalized return counts for transposition walks in S_r.

These formulas serve two different computations. Triangular integer extraction yields the entire BPS sequence for the ordinary point. For the infinite family, an expansion over $\mathbf{Q}[u]$ gives the log-concavity difference as a polynomial of degree at most 42. Translating its argument by 110 makes all its coefficients negative, proving the inequality for every $k \geq 10$. A separate braid-walk argument proves positivity throughout both sequences. Appendix A contains a complete verifier, including all inputs and the checks on the displayed certificate.

2 Full twists and the canonical Hecke trace

Let $v = q^{1/2}$, $z = v - v^{-1}$, and $w = z^2$. The generic Hecke algebra $\mathcal{H}_r$ over $\mathbf{Q}(v)$ has generators $T_1, \dots, T_{r-1}$, the braid relations, and

$$T_i^2 = zT_i + 1. \tag{5}$$

Its standard basis is $(T_\sigma)_{\sigma \in S_r}$. If s_i is the ith simple transposition and $\ell(\sigma)$ is the inversion number of σ, then

$$T_i T_\sigma = \begin{cases} T_{s_i\sigma}, & \ell(s_i\sigma) > \ell(\sigma), \\ T_{s_i\sigma} + zT_\sigma, & \ell(s_i\sigma) < \ell(\sigma). \end{cases} \tag{6}$$

Thus a multiplication step always permits a move $\sigma \mapsto s_i\sigma$, and permits a stay, of weight z, precisely at a descent.

Let w_0 be the longest permutation. For a positive braid word $\beta = \sigma_{i_1} \cdots \sigma_{i_m}$ read from left to right, put

$$W_\beta(z) = [T_{w_0}](T_{i_m} \cdots T_{i_1} T_{w_0}).$$

Here $[T_\sigma]$ denotes coefficient extraction in the standard basis. Thus W_β counts the w_0-to-w_0 walks of [8, Definition 2.1] by their number of stays. Let Λ_β be the Legendrian rainbow closure of β, and let $R_\beta(z)$ be its 2-graded ruling polynomial, with a ruling of j switches weighted by z^{j-r}. The walk–ruling correspondence [8, (2.1)] gives

$$R_\beta(z) = z^{-r} W_\beta(z).$$

If the closure has b components, its normalized ruling polynomial is $\widetilde{R}_\beta(z) = z^b R_\beta(z)$. These are the conventions used in [8, Conjecture 2.3].

We next fix the normalization relating this polynomial to the local BPS invariants. Use the HOMFLY skein convention $a^{-1}P_+ - aP_- = zP_0$ and $P_{\text{unknot}} = 1$. For the link L of $(C, 0)$, put $H_C(q) = \sum_\ell \chi(\mathrm{Hilb}^\ell(C, 0))q^\ell$ and $\mu = 2\delta + 1 - b$. The Hilbert–HOMFLY correspondence [4, 3], specialized to the lowest a-degree, reads

$$[a^\mu]P_L(a, z) = (-1)^{b-1} q^{-\mu/2}(1 - q)H_C(q). \tag{7}$$

The sign is due to the choice $z = q^{1/2} - q^{-1/2}$: the skein convention in [4] has the opposite sign of z, and $P_L(a, -z) = (-1)^{b-1}P_L(a, z)$. Indeed, smoothing changes the component number by one, so $(-1)^{b-1}P_L(a, -z)$ satisfies the same skein relation and unknot normalization as $P_L(a, z)$. In particular,

$$z^{b-1}(-1)^{b-1}q^{-\mu/2}(1 - q) = q^{-\delta}(1 - q)^b.$$

For a positive r-strand braid representative of an algebraic link, its fiber surface gives $\mu = m - r + 1$. The rainbow closure has Thurston–Bennequin number $\mathrm{tb}(\Lambda_\beta) = m - r$. Rutherford's identity [6], in the convention above, is therefore $[a^\mu]P_L = zR_\beta$. Combining these identities gives

$$\sum_h n_h(C, 0)z^{2h} = \widetilde{R}_\beta(z) = z^{b-r} W_\beta(z). \tag{8}$$

This recovers the normalized formulas [8, (4.2), (4.3)].

Remark 2.1. The factor $(-1)^{b-1}$ in (7) is needed with the stated skein convention. Formula (4.1) of version 2 of [8] omits this sign, whereas its normalized formula (4.2) agrees with (8). For a check, the node has $H_C(q) = (1-q+q^2)/(1-q)^2$ and its Hopf link has $[a]P_L = z + z^{-1}$. Both give the BPS polynomial $w+1$. The Hilbert–HOMFLY correspondence used here is independent of the Severi-multiplicity identity discussed in [8, Remark 1.4].

Let Δ be the positive half twist. The link of $C_{r,k}$ is the closure of

$$\beta = \Delta^{2k} = (\sigma_1 \cdots \sigma_{r-1})^{rk}.$$

It is pure and has length $2d$. Hence $b = r$, the prefactor in (8) is 1, and exactly $2h$ stays contribute to $n_h(C_{r,k})$. A closed walk has an even number of stays, since each move changes permutation parity and the total length is even.

Lemma 2.2. *For all $r \geq 2$, $k \geq 1$, and $0 \leq h \leq d = k\binom{r}{2}$,*

$$1 \leq n_h(C_{r,k}) \leq \binom{2d}{2h}. \tag{9}$$

In particular, $n_0 = n_d = 1$.

Proof. Choose a word Q for Δ^k of length d. Reversing a word for Δ gives another word for Δ, because it reverses a reduced expression of the involution w_0. Hence Q^{rev} represents Δ^k and QQ^{rev} represents Δ^{2k}.

For a fixed h, stay during the first h and last h letters of this palindromic word, and move at every other letter. The initial stays are allowed at w_0, since every simple reflection is a descent there. The middle word has the form RR^{rev}; moving at all its letters returns to w_0, since the corresponding simple transpositions cancel in reverse order. The final stays are therefore allowed as well. This is an admissible closed walk with $2h$ stays.

Conversely, specifying the set of stays determines the entire walk: every other step must be a move. There are at most $\binom{2d}{2h}$ such subsets. For $h = 0$ and $h = d$ there is only one subset. □

The walk polynomial can now be evaluated through the character theory of the Hecke algebra. The canonical symmetrizing trace is $\tau(T_\sigma) = \delta_{\sigma,e}$, where e is the identity permutation. The standard dual-basis identity is

$$\tau(T_\sigma T_{\rho^{-1}}) = \delta_{\sigma,\rho}. \tag{10}$$

To verify this identity, equip $\mathcal{H}_r$ with the symmetric bilinear form for which the standard basis is orthonormal. Right multiplication by T_i is self-adjoint: on a pair $(T_\sigma, T_{\sigma s_i})$ with $\ell(\sigma s_i) = \ell(\sigma) + 1$, its matrix is $\left(\begin{smallmatrix} 0 & 1 \\ 1 & z \end{smallmatrix}\right)$. Moving the factors of a reduced expression for T_ρ across this form gives

$$\delta_{\sigma,\rho} = \langle T_\sigma, T_\rho \rangle = \langle T_\sigma T_{\rho^{-1}}, 1 \rangle = \tau(T_\sigma T_{\rho^{-1}}).$$

This also implies $\tau(ab) = \tau(ba)$ by checking basis elements. Put $\Omega = T_{w_0}^2$, the full twist.

Proposition 2.3. *For $r \geq 2$ and $k \geq 1$,*

$$B_{r,k}(w) = \tau(\Omega^{k+1}). \tag{11}$$

Proof. Reversing a word for Δ^{2k} again represents Δ^{2k}, as in the proof of Lemma 2.2. Thus the reversed braid word in W_β has Hecke element Ω^k. Equations (8) and (10) give

$$B_{r,k}(w) = [T_{w_0}](\Omega^k T_{w_0}) = \tau(\Omega^k T_{w_0} T_{w_0}) = \tau(\Omega^{k+1}).$$ □

For a partition $\lambda \vdash r$, let λ' be its conjugate and S^λ the corresponding generic Specht module. Write

$$\nu(\lambda) = \sum_i (i-1)\lambda_i, \qquad c_\lambda = \sum_{(i,j)\in\lambda} (j-i) = \nu(\lambda') - \nu(\lambda).$$

For a box $\square = (i,j) \in \lambda$, its hook length is $h_\square = \lambda_i - j + \lambda'_j - i + 1$. The dimension of the Specht module is

$$f^\lambda = \frac{r!}{\prod_{\square\in\lambda} h_\square}.$$

We use the standard Schur-element and Murphy-element formulas for the generic type-A Hecke algebra [2]. To compare normalizations, $G_i = vT_i$ satisfies $(G_i - q)(G_i + 1) = 0$, and the canonical trace is unchanged by this rescaling of the standard basis. Its character expansion is

$$\tau(a) = \sum_{\lambda\vdash r} \frac{\mathrm{Tr}_{S^\lambda}(a)}{s_\lambda(q)}, \qquad s_\lambda(q) = q^{-\nu(\lambda)} \prod_{\square\in\lambda} \frac{1-q^{h_\square}}{1-q}. \tag{12}$$

In our normalization, the Murphy elements are

$$L_1 = 1, \qquad L_j = T_{j-1}\cdots T_2 T_1^2 T_2 \cdots T_{j-1} \quad (2 \le j \le r), \qquad \Omega = \prod_{j=1}^{r} L_j.$$

On a seminormal tableau vector, L_j acts by $q^{\mathrm{col}(j)-\mathrm{row}(j)}$, where the row and column refer to the box containing j. Consequently Ω acts on S^λ as q^{c_λ}. At $q = 1$, (12) gives $s_\lambda(1) = r!/f^\lambda$, as required for the normalized regular trace of $\mathbf{Q}[S_r]$.

Proposition 2.4. *The punctual Hilbert series of $C_{r,k}$ is*

$$H_{r,k}(q) := \sum_{\ell\ge 0} \chi(\mathrm{Hilb}^\ell(C_{r,k}, 0)) q^\ell = \sum_{\lambda\vdash r} \frac{f^\lambda q^{E_{r,k}(\lambda)}}{\prod_{\square\in\lambda}(1-q^{h_\square})}, \tag{13}$$

where

$$\begin{aligned} E_{r,k}(\lambda) &= d + (k+1)c_\lambda + \nu(\lambda) \\ &= k(N - \nu(\lambda)) + (k+1)\nu(\lambda') \ge 0. \end{aligned} \tag{14}$$

Proof. Equations (11) and (12) give

$$B_{r,k}(w) = \sum_{\lambda\vdash r} \frac{f^\lambda q^{(k+1)c_\lambda + \nu(\lambda)}(1-q)^r}{\prod_{\square\in\lambda}(1-q^{h_\square})}. \tag{15}$$

Multiply by $q^d/(1-q)^r$ and apply (1). The last expression in (14) is nonnegative because $\nu(\lambda) \le \binom{r}{2}$. □

3 Polynomial dependence on the contact order

To study a fixed BPS coefficient as the contact order varies, we expand (15) at $w = 0$. Conjugate partitions pair the positive and negative powers of v, exposing the dependence on $k(k+1)$. Put $s = v + v^{-1}$, so $s^2 = w + 4$, and set

$$[a]_v = \frac{v^a - v^{-a}}{v - v^{-1}} \qquad (a \in \mathbf{Z}_{\ge 1}).$$

Define polynomials $D_a(w)$ by

$$[a]_v = \begin{cases} D_a(w), & a \text{ odd}, \\ sD_a(w), & a \text{ even}. \end{cases} \tag{16}$$

Explicitly,

$$\begin{gathered} D_1 = D_2 = 1, \qquad D_3 = w+3, \qquad D_4 = w+2, \\ D_a = (w+2)D_{a-2} - D_{a-4} \quad (a \geq 5). \end{gathered} \tag{17}$$

In particular, $D_a(0) = a$ for odd a, and $D_a(0) = a/2$ for even a.

Let e_λ be the number of even hook lengths of λ. The identity

$$\sum_{\square \in \lambda} h_\square = r + \nu(\lambda) + \nu(\lambda') = r + 2\nu(\lambda) + c_\lambda$$

implies $e_\lambda \equiv c_\lambda \pmod 2$. Define the power series

$$R_\lambda(w) = (1 + w/4)^{-\lfloor e_\lambda/2 \rfloor} \prod_{\square \in \lambda} \frac{D_{h_\square}(0)}{D_{h_\square}(w)}. \tag{18}$$

It has constant coefficient 1.

Put $u = k(k+1)$. For each partition define $A_{\lambda,0}(u) = 1$, and for $j \geq 1$ define

$$A_{\lambda,j}(u) = \frac{A_{\lambda,j-1}(u)}{4(2j)(2j-1)} \begin{cases} c_\lambda^2(4u+1) - 4(j-1)^2, & c_\lambda \text{ even}, \\ c_\lambda^2(4u+1) - (2j-1)^2, & c_\lambda \text{ odd}. \end{cases} \tag{19}$$

These are polynomials of degree at most j.

Proposition 3.1. *For integers $r \geq 2$ and $h \geq 0$ there is a polynomial $Q_{r,h}(u) \in \mathbf{Q}[u]$ of degree at most h such that, whenever $k \geq 1$ and $h \leq k\binom{r}{2}$,*

$$n_h(C_{r,k}) = Q_{r,h}(k(k+1)) = \frac{1}{r!} \sum_{\lambda \vdash r} (f^\lambda)^2 \sum_{j=0}^{h} A_{\lambda,j}(k(k+1))[w^{h-j}]R_\lambda(w). \tag{20}$$

The polynomial extension at $k = 0$ satisfies $Q_{r,0}(0) = 1$ and $Q_{r,h}(0) = 0$ for $h > 0$.

Proof. Rewriting (15) in the variable v gives

$$B_{r,k}(w) = \sum_{\lambda \vdash r} \frac{f^\lambda v^{(2k+1)c_\lambda}}{\prod_{\square \in \lambda} [h_\square]_v}. \tag{21}$$

Transposition preserves the hooks and f^λ, and changes the sign of c_λ. Averaging the summands for λ and λ' therefore replaces the numerator by $\frac{1}{2} f^\lambda (v^a + v^{-a})$, where $a = (2k+1)|c_\lambda|$. This averaging also applies to self-transpose partitions.

Write $\epsilon \in \{0, 1\}$ for the parity of a and normalize the numerator as

$$\Phi_a(w) = \begin{cases} (v^a + v^{-a})/2, & \epsilon = 0, \\ (v^a + v^{-a})/s, & \epsilon = 1. \end{cases}$$

The identities

$$\Phi_0 = \Phi_1 = 1, \qquad \Phi_2 = 1 + w/2, \qquad \Phi_3 = 1 + w, \qquad \Phi_{a+2} = (w+2)\Phi_a - \Phi_{a-2} \quad (a \geq 2)$$

show that $\Phi_a \in \mathbf{Q}[w]$ and $\Phi_a(0) = 1$. To compute its coefficients, set $v = e^t$, so $w = 4\sinh^2 t$ and $\Phi_a = \cosh(at)/(\cosh t)^\epsilon$. Differentiation gives, with primes denoting w-derivatives,

$$w(w+4)\Phi_a'' + ((1+\epsilon)w + 2)\Phi_a' - \frac{a^2-\epsilon}{4}\Phi_a = 0.$$

If $\Phi_a(w) = \sum_{j\ge 0} \alpha_j w^j$, comparison of the coefficient of w^{j-1} yields

$$\alpha_0 = 1, \qquad \alpha_j = \frac{a^2 - \epsilon - 4(j-1)(j-1+\epsilon)}{4(2j)(2j-1)}\alpha_{j-1} \quad (j \ge 1).$$

Since $a^2 = c_\lambda^2(4u+1)$, this is exactly (19).

Since $e_\lambda \equiv a \pmod 2$, all remaining powers of s in the denominator combine to $(w+4)^{\lfloor e_\lambda/2 \rfloor}$. The constant term of the resulting summand is $f^\lambda / \prod h_\square = (f^\lambda)^2/r!$. Normalizing its denominator yields precisely (18), proving (20). Its degree assertion follows from (19). The right side of (21) also makes sense at $k = 0$. The character formula (12) identifies it with $\tau(\Omega) = 1$, by (10). Its w^h-coefficient is $Q_{r,h}(0)$, proving the final assertion without introducing a curve $C_{r,0}$. □

The leading coefficients have a symmetric-group interpretation. Let $T_h(r)$ be the number of ordered products of $2h$ transpositions in S_r whose product is the identity. By the ordinary Murphy-element eigenvalue formula [2], the central element $J = \sum_{i<j}(ij)$ acts on the irreducible S_r-module of shape λ by c_λ. The coefficient of the identity in J^{2h} is $T_h(r)$; taking the normalized regular character therefore gives

$$T_h(r) = \frac{1}{r!}\sum_{\lambda \vdash r} (f^\lambda)^2 c_\lambda^{2h}. \tag{22}$$

Proposition 3.2. *The coefficient of u^h in $Q_{r,h}(u)$ is*

$$[u^h]Q_{r,h}(u) = \frac{T_h(r)}{(2h)!}.$$

Proof. The coefficient of u^j in (19) is $c_\lambda^{2j}/(2j)!$. In (20), only $j = h$ can contribute to u^h, and $R_\lambda(0) = 1$. Equation (22) proves the assertion. □

Thus, for $h \ge 1$, a necessary condition for $Q_{r,h}(u)^2 \ge Q_{r,h-1}(u)Q_{r,h+1}(u)$ for all sufficiently large u is

$$(2h+2)(2h+1)T_h(r)^2 \ge (2h)(2h-1)T_{h-1}(r)T_{h+1}(r). \tag{23}$$

4 Exact certification of the ordinary point

Formula (13) computes the coefficients $H_\ell = [q^\ell]H_{r,k}$ by a finite partition sum. To compute a summand through degree d, start with the array $(1, 0, \ldots, 0)$ and, for each hook length a, replace its entries successively by

$$t_j \longleftarrow t_j + t_{j-a} \qquad (j = a, a+1, \ldots, d-E).$$

Multiply the resulting array by f^λ, shift it by E, and add it to (H_ℓ). Thus only ordinary nonnegative integer additions and the hook-length dimension formula are required.

Put $F_\ell = [q^\ell](1-q)^r H_{r,k}(q)$. Equation (1) is equivalent to

$$(1-q)^r H_{r,k}(q) = \sum_{h=0}^{d} n_h q^{d-h}(1-q)^{2h}. \tag{24}$$

It gives the triangular recurrence

$$n_h = F_{d-h} - \sum_{j=h+1}^{d} (-1)^{j-h} \binom{2j}{j-h} n_j, \qquad h = d, d-1, \ldots, 0. \tag{25}$$

The function `bps_coefficients` in Appendix A implements these operations with integer arrays. Applied to $(r,k) = (43,1)$, it produces all 904 coefficients and checks the three integers in Theorem 1.1. No truncation error is involved: terms of degree greater than d cannot affect (25), and each step determines one coefficient from a term with leading coefficient 1.

Proof of Theorem 1.1. The finite partition sum (13) and the triangular recurrence (25), evaluated by the complete integer algorithm in Appendix A, give the three displayed values. Their exact products satisfy (4). The parameters follow from (3); strict positivity of every coefficient follows from Lemma 2.2, and is also checked for all 904 entries by the algorithm. □

5 An infinite family with an explicit threshold

Fix $r = 36$ and abbreviate $Q_h = Q_{36,h}$. The leading-coefficient test (23) fails at $(r,h) = (36,21)$. Exact integer evaluation of (22) gives

$$\frac{99993}{100000} < \frac{44 \cdot 43\, T_{21}(36)^2}{42 \cdot 41\, T_{20}(36) T_{22}(36)} < \frac{99994}{100000}. \tag{26}$$

Consequently the polynomial

$$G(u) = Q_{21}(u)^2 - Q_{20}(u) Q_{22}(u), \qquad \deg G \le 42, \tag{27}$$

has a negative leading coefficient. This proves eventual failure of log-concavity. To establish the explicit range $k \ge 10$, we need the full polynomial rather than its leading term alone.

We compute Q_{20}, Q_{21}, Q_{22} directly from (20), retaining terms through degree 22 in w. Every D_a has nonzero constant term, so the required truncated inverses exist over $\mathbf{Q}$.

The series R_λ in (18) can be expanded without manipulating square roots. Write $R_\lambda = \sum_{m \ge 0} r_m w^m$, with $r_0 = 1$, and

$$\frac{R'_\lambda}{R_\lambda} = -\frac{\lfloor e_\lambda/2 \rfloor}{w+4} - \sum_{\square \in \lambda} \frac{D'_{h_\square}}{D_{h_\square}} = \sum_{i \ge 1} \gamma_i w^{i-1}. \tag{28}$$

Then

$$r_m = \frac{1}{m} \sum_{i=1}^{m} \gamma_i r_{m-i}. \tag{29}$$

Only $\gamma_1, \ldots, \gamma_{22}$ and $r_0, \ldots, r_{22}$ are needed. For each partition, (19) constructs the coefficient arrays of $A_{\lambda,j}(u)$ exactly; convolution with these truncated series gives (20). The function `contact_polynomials` in Appendix A carries out this calculation. It groups the weighted series R_λ by $|c_\lambda|$, since the polynomials $A_{\lambda,j}$ depend only on that value. This reduces repeated polynomial arithmetic without changing the partition sum.

Exact rational expansion and multiplication give

$$G(110 + x) = -\frac{1}{\mathcal{D}} \sum_{j=0}^{42} b_j x^j, \qquad b_j \in \mathbf{Z}_{>0}, \tag{30}$$

where the positive denominator is

$$\mathcal{D} = 2^{63}3^{32}5^{16}7^{10}11^{7}13^{6}17^{4}19^{4}(23 \cdot 29 \cdot 31 \cdot 37 \cdot 41)^{2}\,43. \tag{31}$$

Table 1 displays the signs compactly: its entry (d_j, e_j) means the exact bound

$$d_j 10^{e_j} \le b_j < (d_j + 1)10^{e_j}. \tag{32}$$

All d_j are positive. The full integers are determined, without rounding, by $b_j = -\mathcal{D}[x^j]G(110 + x)$ and the finite formulas (17)–(20). The appendix generates these numbers and verifies their signs directly; the displayed decimal bounds are a compact record of the output.

Table 1: Exact positive bounds for all coefficients in (30), in the notation of (32).

j	d_j	e_j	j	d_j	e_j	j	d_j	e_j
0	300722	242	15	319860	224	30	170198	193
1	450959	242	16	522779	222	31	618795	190
2	146709	242	17	771483	220	32	199581	188
3	253819	241	18	103046	219	33	566917	185
4	292356	240	19	124805	217	34	140542	183
5	248562	239	20	137250	215	35	300596	180
6	165463	238	21	137166	213	36	546495	177
7	895750	236	22	124629	211	37	827970	174
8	404867	235	23	102950	209	38	101710	172
9	155743	234	24	772822	206	39	973272	168
10	517344	232	25	526728	204	40	680591	165
11	150076	231	26	325506	202	41	309354	162
12	383600	229	27	182048	200	42	685924	158
13	870127	227	28	919200	197			
14	176170	226	29	417723	195			

Explicitly, if $G(u) = \sum_{i=0}^{42} g_i u^i$, the coefficient used in the sign test is

$$[x^j]G(110 + x) = \sum_{i=j}^{42} \binom{i}{j} 110^{i-j} g_i.$$

The appendix evaluates these sums over $\mathbf{Q}$, checks that all 43 are strictly negative, computes the least common multiple of their denominators, and checks every bound in Table 1. It also verifies (26) and compares the three moments with the leading coefficients of Q_{20}, Q_{21}, Q_{22}. Thus the displayed bounds record exact integer inequalities, not rounded numerical evidence.

Proof of Theorem 1.2. For an integer $k \ge 10$, put $u = k(k + 1)$. Then $u - 110 \ge 0$. Equation (30), with $\mathcal{D} > 0$ and every $b_j > 0$, gives $G(u) < 0$. By (20) this is exactly the required strict log-concavity violation. Lemma 2.2 proves strict positivity of all the BPS coefficients. □

References

[1] O. Capovilla-Searle and Y. Pan, On ruling polynomials of Legendrian links, arXiv:2608.18255v1, 18 August 2026.

[2] A. Mathas, *Iwahori–Hecke Algebras and Schur Algebras of the Symmetric Group*, University Lecture Series, vol. 15, American Mathematical Society, Providence, RI, 1999.

[3] D. Maulik, Stable pairs and the HOMFLY polynomial, *Invent. Math.* **204** (2016), 787–831. https://arxiv.org/abs/1210.6323.

[4] A. Oblomkov and V. Shende, The Hilbert scheme of a plane curve singularity and the HOMFLY polynomial of its link, *Duke Math. J.* **161** (2012), 1277–1303. https://arxiv.org/abs/1003.1568.

[5] R. Pandharipande and R. P. Thomas, Stable pairs and BPS invariants, *J. Amer. Math. Soc.* **23** (2010), 267–297. https://arxiv.org/abs/0711.3899.

[6] D. Rutherford, The Thurston–Bennequin number, Kauffman polynomial, and ruling invariants of a Legendrian link: The Fuchs conjecture and beyond, *Int. Math. Res. Not.* **2006** (2006), Art. ID 78591. https://arxiv.org/abs/math/0511097.

[7] V. Shende, Hilbert schemes of points on a locally planar curve and the Severi strata of its versal deformation, *Compos. Math.* **148** (2012), 531–547. https://arxiv.org/abs/1009.0914.

[8] T. Su, B. Xie, and C. Yu, *Log-concavity from enumerative geometry of planar curve singularities*, arXiv:2603.27888v2, 30 April 2026.

A Complete exact-arithmetic verifier

The following program uses Python 3.10 or later and only its standard library. It has no external inputs: the main routine specifies the parameters, and all partitions, hooks, dimensions, Hilbert coefficients, and contact-order polynomials are generated from the formulas above. Integers have arbitrary precision, and every nonintegral division uses `Fraction`. No floating-point arithmetic is used.

The partition recursion chooses each possible first part and then a partition of the remainder with no larger part. Induction on the remainder shows that it visits every partition exactly once. The function `shape` uses zero-based row and column indices, so its hook-length expression is $\lambda_{i+1} - j + \lambda'_{j+1} - i - 1$. The hook-length formula supplies f^λ. The function `bps_coefficients` implements (13) and (25) through degree d. All shifts E are nonnegative; discarding terms above d therefore cannot change any coefficient used by the triangular extraction.

In `contact_polynomials`, `g[a]` stores $-D'_a/D_a$ through degree $H-1$, `log` stores the γ_i of (28), and `R` is computed by (29). Grouping the weighted series by $|c_\lambda|$ implements (20), since (19) depends only on that absolute content. Terms above degree H in w cannot contribute to any $Q_{r,h}$ with $h \le H$.

The main routine checks the ordinary 43-fold point, the degree-42 sign certificate, every entry of Table 1, and the three transposition moments. The table bounds are checked against independently generated coefficients; they are not used to compute them. Save the listing as a Python file and execute it normally, without optimization flags. Successful completion prints `ALL CHECKS PASSED`; any failed assertion invalidates the computational certificate. The program rejects execution with assertions disabled. The sign test certifies all $k \ge 10$ because Proposition 3.1 supplies the degree bound and (30) holds as a polynomial identity.

```
from fractions import Fraction as F
from math import comb, factorial, prod, lcm
from collections import Counter


def partitions(n, cap=None):
    if n == 0:
        yield ()
    else:
        for a in range(min(n, n if cap is None else cap), 0, -1):
            for tail in partitions(n-a, a):
                yield (a,) + tail


def shape(lam):
    col = [sum(a > j for a in lam) for j in range(lam[0])]
    hooks = [a-j+col[j]-i-1
             for i, a in enumerate(lam) for j in range(a)]
    nu = sum(i*a for i, a in enumerate(lam))
    c = sum(a*(a-1)//2-i*a for i, a in enumerate(lam))
    f, rem = divmod(factorial(sum(lam)), prod(hooks))
    assert rem == 0
    return hooks, nu, c, f


def bps_coefficients(r, k=1):
    assert r >= 2 and k >= 1
    d = k*r*(r-1)//2
    H = [0]*(d+1)
    for lam in partitions(r):
        hooks, nu, c, f = shape(lam)
```

```
            E = d+(k+1)*c+nu
            assert E >= 0
            if E > d:
                continue
            t = [1]+[0]*(d-E)
            for a in hooks:
                for j in range(a, len(t)):
                    t[j] += t[j-a]
            for j, value in enumerate(t):
                H[E+j] += f*value
        for _ in range(r):
            H = [H[0]]+[H[j]-H[j-1] for j in range(1,d+1)]
        n = [0]*(d+1)
        for ell in range(d+1):
            h = d-ell
            n[h] = H[ell]
            for j in range(min(2*h,d-ell)+1):
                H[ell+j] -= n[h]*(-1)**j*comb(2*h,j)
        assert not any(H)
        return n


    def mul(a,b):
        out = [F(0)]*(len(a)+len(b)-1)
        for i,x in enumerate(a):
            for j,y in enumerate(b):
                out[i+j] += x*y
        return out


    def contact_polynomials(r, H):
        assert r >= 2 and H >= 0
        # D[a] is the normalized quantum integer polynomial.
        D = [[F(0)]*(H+1) for _ in range(r+1)]
        D[1][0] = D[2][0] = F(1)
        for a, coeffs in ((3, (3, 1)), (4, (2, 1))):
            if a <= r:
                for j, value in enumerate(coeffs[:H+1]):
                    D[a][j] = F(value)
        for a in range(5,r+1):
            for j in range(H+1):
                D[a][j] = 2*D[a-2][j]-D[a-4][j]
                if j:
                    D[a][j] += D[a-2][j-1]
        # g[a] = -D[a]'/D[a].
        g = [[F(0)]*H for _ in range(r+1)]
        for a in range(1,r+1):
            for j in range(H):
                g[a][j] = (-(j+1)*D[a][j+1]-sum(
                    D[a][t]*g[a][j-t] for t in range(1,j+1)))/D[a][0]
        by_content = {}
        order = factorial(r)
        dimension_sum = 0
        for lam in partitions(r):
            hooks,nu,c,f = shape(lam)
            dimension_sum += f*f
            even = sum(a%2 == 0 for a in hooks)
            assert (even-c)%2 == 0
            count = Counter(hooks)
            log = [F(0)]+[sum(m*g[a][j-1] for a,m in count.items())
                + F((even//2)*(-1)**j,4**j) for j in range(1,H+1)]
            R = [F(1)]+[F(0)]*H
            for j in range(1,H+1):
```

```
            R[j] = sum(log[t]*R[j-t] for t in range(1,j+1))/j
        total = by_content.setdefault(abs(c),[F(0)]*(H+1))
        weight = F(f*f,order)
        for j in range(H+1):
            total[j] += weight*R[j]
    assert dimension_sum == order
    Q = [[F(0)]*(h+1) for h in range(H+1)]
    for c,R in by_content.items():
        A = [F(1)]
        for j in range(H+1):
            if j:
                t = 2*j-1 if c%2 else 2*(j-1)
                den = 4*(2*j)*(2*j-1)
                A = mul(A,[F(c*c-t*t,den),F(4*c*c,den)])
            for h in range(j,H+1):
                for power,value in enumerate(A):
                    Q[h][power] += value*R[h-j]
    return Q


def verify():
    if not __debug__:
        raise RuntimeError('Run Python with assertions enabled.')
    n=bps_coefficients(43)
    a=467438304194650004605257498534262968215115139192 67
    b=447870695400119839679788896911479468288995454466029 5
    c=429147501803199891064835526788080765663004079176388506
    assert n[23:26] == [a,b,c]
    assert len(n)==904 and min(n)>0 and n[0]==n[-1]==1
    assert 20000*b*b < 19999*a*c
    print('ordinary43 PASS',flush=True)
    Q=contact_polynomials(36,22)
    for h in range(23):
        assert Q[h][0] == (1 if h==0 else 0)
    left,right=mul(Q[21],Q[21]),mul(Q[20],Q[22])
    G=[a-b for a,b in zip(left,right)]
    shifted=[sum(G[i]*comb(i,j)*110**(i-j)
                 for i in range(j,len(G))) for j in range(len(G))]
    assert len(shifted)==43 and all(x<0 for x in shifted)
    den=lcm(*(x.denominator for x in shifted))
    expected=(2**63*3**32*5**16*7**10*11**7*13**6*17**4*19**4
              *(23*29*31*37*41)**2*43)
    assert den == expected
    bounds = [
        (300722,242),(450959,242),(146709,242),
        (253819,241),(292356,240),(248562,239),
        (165463,238),(895750,236),(404867,235),
        (155743,234),(517344,232),(150076,231),
        (383600,229),(870127,227),(176170,226),
        (319860,224),(522779,222),(771483,220),
        (103046,219),(124805,217),(137250,215),
        (137166,213),(124629,211),(102950,209),
        (772822,206),(526728,204),(325506,202),
        (182048,200),(919200,197),(417723,195),
        (170198,193),(618795,190),(199581,188),
        (566917,185),(140542,183),(300596,180),
        (546495,177),(827970,174),(101710,172),
        (973272,168),(680591,165),(309354,162),
        (685924,158)]
    assert len(bounds) == len(shifted)
    for coefficient,(prefix,exponent) in zip(shifted,bounds):
        value=-den*coefficient
        assert value.denominator == 1
```

```
            assert prefix*10**exponent <= value
            assert value < (prefix+1)*10**exponent
        print('uniform36 and table PASS',flush=True)
        moments=[0,0,0]
        for lam in partitions(36):
            hooks,nu,c,f=shape(lam)
            for j,h in enumerate((20,21,22)):
                moments[j] += f*f*c**(2*h)
        moments=[F(t,factorial(36)) for t in moments]
        assert all(t.denominator==1 for t in moments)
        for h,t in zip((20,21,22),moments):
            assert Q[h][h] == t/factorial(2*h)
        a,b,c=moments
        ratio=F(44*43,42*41)*b*b/(a*c)
        assert F(99993,100000)<ratio<F(99994,100000)
        print('transposition moments PASS',flush=True)
        print('ALL CHECKS PASSED',flush=True)
        return n,Q,shifted

if __name__ == '__main__':
    verify()
```