

Exact Rational Integration over Constrained Simplices

Symmetric Polynomial Bases, Marginal Geometry, and Lean 4 Verification

Reading Point (translatepi.ai)

Draft v0.4 — 19 October 2026

Abstract

We present a formalized collection of integration identities for polynomial weights on simplexes constrained by coordinate thresholds and cardinality-dependent caps. The development includes exact simplex moments, a rational polynomial subspace of dimension 846, cap-parametric formulas for monomial integrals, rational Gram-entry representations, fibre decompositions, and two-aggregate Dirichlet integration. Exact polygon-cell formulas now cover all branches of the marginal endpoint, yielding a kernel-checked rational value for the first marginal Gram entry $J_T[1, 1]$ under Table 3 parameters. These statements are formalized in Lean 4 and supported by independently replayed Comparator suites and boundary controls. The constructions arise from a bounded-prime-gap argument, but this manuscript studies their integration and polynomial mathematics independently. General marginal Gram entries involving power sums, variational positivity, and remaining analytic obligations lie outside the completed results.

1 Introduction and scope

Exact integration on constrained regions is useful where a variational inequality must ultimately be checked using rational arithmetic. A conventional simplex has a closed monomial moment formula; the present support also depends on which coordinates cross a threshold and on a cap for the sum of those coordinates. This introduces a finite stratification, but preserves a route to exact, rational expressions.

We study the mathematical constructions implemented in the public Lean repository [thinkthoughts/lean-bounded-prime-gap](https://github.com/thinkthoughts/lean-bounded-prime-gap). The motivating bounded-prime-gap argument originates with Axiom Math [1]; the statements below are an independent formalization of specified components and do not amount to a complete formal proof of the unconditional prime-gap bound. References to Lean declarations are included to delimit precisely what has been checked. The short arguments given here are *proof outlines*, not substitutes for the complete Lean proof terms.

2 Polynomial moments on a simplex

Definition 2.1. For $m \in \mathbb{N}$ and $R \in \mathbb{R}$, define

$$\Delta_m(R) = \left\{ x \in \mathbb{R}^m : x_i \geq 0 \text{ for all } i, \quad \sum_{i=1}^m x_i \leq R \right\}.$$

Theorem 2.2 (Simplex monomial moments). For $m \in \mathbb{N}$, $R \geq 0$, and $e \in \mathbb{N}^m$,

$$\int_{\Delta_m(R)} \prod_{i=1}^m x_i^{e_i} dx = \frac{R^{m+\sum_i e_i} \prod_i (e_i!)}{(m + \sum_i e_i)!}.$$

Here the dimension-zero case uses the corresponding Lean integral convention.

Lean: `simplex_integral`; definitions `simplex`, `momentR`.

Source: `Solution/SimplexMoment.lean:160`.

Proof outline. Induct on dimension and use Fubini to isolate a coordinate. The inner integral is a Beta integral; the factorial expression then simplifies by induction. The formal development additionally treats negative and zero radii, simplex volume, and rational-polynomial integrals (`poly_integral`).

3 An indexed rational polynomial space

Consider rational polynomials in 45 coordinates. Write P_j for the j th power sum, P_λ for the product over parts of a multiset λ , and set

$$G_{a,\lambda} = (L - P_1)^a P_\lambda, \quad \lambda \text{ has positive even parts}, \quad a + |\lambda| \leq 21.$$

The index type `B21Idx` enumerates the admissible pairs.

Theorem 3.1 (Independence and rank). *The indexed family $G_{a,\lambda}$ is linearly independent over $\mathbb{Q}$. Its rational span B_{21} has*

$$\dim_{\mathbb{Q}} B_{21} = 846.$$

Lean: `genIdx_linearIndependent, finrank_B21 (Solution/Variational.lean:398,410)`.

Proof outline. The indexing is injective; distinct indexed monomials are linearly independent. Injective polynomial transformations preserve this independence. A separate count establishes $|B_{21}\text{Idx}| = 846$, and the span-rank identity gives the result. The theorem concerns this designated subspace, not the full polynomial ring.

4 Cap-constrained supports and general integrals

Let $N \in \mathbb{N}$, $g \geq 0$, $L \in \mathbb{R}$, and $C : \mathbb{N} \rightarrow \mathbb{R}$. For $x \in \mathbb{R}^N$ write

$$R_g(x) = \{i : x_i > g\}, \quad s_g(x) = |R_g(x)|, \quad \rho_g(x) = \sum_{i \in R_g(x)} x_i.$$

Define the capped simplex support

$$T(N, g, L, C) = \left\{ x \in \mathbb{R}^N : x_i \geq 0, \quad \sum_i x_i \leq L, \quad \rho_g(x) \leq C(s_g(x)) \right\}. \quad (1)$$

Theorem 4.1 (General monomial integral). *For $N \in \mathbb{N}$, $g \geq 0$, $L \in \mathbb{R}$, $C : \mathbb{N} \rightarrow \mathbb{R}$, $e \in \mathbb{N}^N$, and $p \in \mathbb{N}$,*

$$\int_{T(N,g,L,C)} \left(L - \sum_i x_i \right)^p \prod_i x_i^{e_i} dx = \text{gramGen}(N, g, L, C, e, p). \quad (2)$$

The right-hand side is the explicit, finite expression defined in Lean.

Lean: `integral_supportT_gen; gramGen`.

Sources: `Solution/GramGeneral.lean:344;`

`BGP212/GramGeneral.lean`.

Proof outline. Stratify by the rough-coordinate set, apply inclusion–exclusion to threshold conditions, and translate the resulting pieces to weighted simplexes. Use Theorem 2.2 and Fubini to evaluate the polynomial weights. Placement-counting identities `sig_insert`, `sig_union`, and `sig_card` reduce the sums to the finite closed form. The reduction `gramGen_extend` transports calculation to the coordinates having nonzero exponents. This is an algebraic efficiency result, not a claim that all entries are computationally cheap to certify.

5 Rational Gram representations

Specialize to the fixed rational cap values and support parameters of Table 3 in [1], with $N = 45$. For indices $i, j \in B_{21}\text{Idx}$ define

$$I_T(i, j) = \int_T G_i(x) G_j(x) dx.$$

Theorem 5.1 (Explicit rational representation). *For every $i, j \in B_{21}\text{Idx}$, $I_T(i, j)$ equals the real embedding of a finite rational sum indexed by placements φ of the combined power-sum parts:*

$$I_T(i, j) = \left(\sum_{\varphi} \text{gramGen}_{\mathbb{Q}}(45, g, L, C, e_{\varphi}, a_i + a_j) \right)_{\mathbb{R}}.$$

Here e_{φ} is the exponent vector obtained from the placement, exactly as specified by `expo` in the Lean source.

Lean: `gram_entry_B21_rat` (Solution/GramGeneral.lean:522); `gram_entry_gen_list` and `gram_entry_example`.

The theorem gives each entry an explicit rational representation; it does not supply an evaluated and certified 846×846 matrix. One worked entry, corresponding to $(L - P_1)P_2^2$, has its value certified using a collection of small exact rational lemmas. This separation between an algebraic identity and its numerical certificate is important to reproducibility.

6 Marginal-fibre geometry

Let $\hat{x} \in \mathbb{R}^n$ and append $u \in \mathbb{R}$ as the last coordinate. Define $S = \sum_i \hat{x}_i$, $s = s_g(\hat{x})$, $\rho_1 = \rho_g(\hat{x})$, and

$$\hat{T} = \{\hat{x} \geq 0 : S \leq \tau, \rho_1 \leq C(s)\}, \quad U(\hat{x}) = \min\{L - S, \max(g, C(s + 1) - \rho_1)\}.$$

Theorem 6.1 (Fibre equivalence). *Suppose $g \geq 0$, $\tau \leq L$, and $\hat{x} \in \hat{T}$. Then for every $u \in \mathbb{R}$,*

$$(\hat{x}, u) \in T(n + 1, g, L, C) \iff 0 \leq u \leq U(\hat{x}).$$

Lean: `fibre_equiv` (Solution/Fibre.lean:115).

Proof outline. Appending u leaves the rough set unchanged for $u \leq g$, and increments both rough count and rough mass for $u > g$. Combining the resulting cap inequality with $S + u \leq L$ identifies the minimum–maximum endpoint exactly. The empty-fibre theorem for cap-violating outer coordinates additionally needs $C(m + 1) \leq C(m) + g$ for $m \geq 1$, and $C(0) \geq 0$. No constraint on the step $m = 0$ is assumed; the relevant Table 3 facts are checked in Lean.

Theorem 6.2 (Marginal identity). *Assume $g \geq 0$, $\tau \leq L$, $C(0) \geq 0$, and $C(m + 1) \leq C(m) + g$ for every $m \geq 1$. For continuous P ,*

$$\text{marginalJ}(n, g, L, \tau, C, P) = J_T(n, g, L, \tau, C, P).$$

The right-hand functional integrates the squared fibre integral over $\hat{T}$.

Lean: `marginal_agrees` (Solution/Fibre.lean:217).

Proposition 6.3 (Polynomial inner integral). *For arbitrary $U, M \in \mathbb{R}$ and $p, t \in \mathbb{N}$,*

$$\int_0^U (M - u)^p u^t du = \sum_{j=0}^p \binom{p}{j} M^{p-j} (-1)^j \frac{U^{t+j+1}}{t+j+1}.$$

Lean: `inner_integral` (Solution/Fibre.lean:240). The equality follows by expanding the binomial and integrating each power. It holds for arbitrary real U under the oriented interval-integral convention.

7 Two-aggregate Dirichlet integration

Let I and J be disjoint, nonempty finite coordinate groups, with exponent vectors $a \in \mathbb{N}^I$ and $b \in \mathbb{N}^J$. For a nonempty finite group G and an exponent vector e , define

$$\alpha_G(e) = |G| + \sum_{i \in G} e_i - 1, \quad A_G(e) = \frac{\prod_{i \in G} (e_i!)}{\alpha_G(e)!}.$$

Write $\Delta_{I \sqcup J}(M)$ for the nonnegative simplex with total coordinate sum at most M , and

$$\Delta_2(M) = \{(u_1, u_2) \in \mathbb{R}_{\geq 0}^2 : u_1 + u_2 \leq M\}.$$

Theorem 7.1 (Two-aggregate Dirichlet integration). *Suppose I and J are nonempty, and $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ is measurable and bounded on $\Delta_2(M)$. With $u_1 = \sum_{i \in I} y_i$ and $u_2 = \sum_{j \in J} y_j$, the weighted simplex integral satisfies*

$$\begin{aligned} & \int_{\Delta_{I \sqcup J}(M)} \left(\prod_{i \in I} y_i^{a_i} \right) \left(\prod_{j \in J} y_j^{b_j} \right) h(u_1, u_2) dy \\ &= A_I(a) A_J(b) \int_{\Delta_2(M)} u_1^{\alpha_I(a)} u_2^{\alpha_J(b)} h(u_1, u_2) du_1 du_2. \end{aligned} \quad (3)$$

The formalization also gives the version on $\text{Fin } N$ for a rough coordinate set and its complement.

Lean: `two_group`, `two_group_fin`, `one_group`; `Solution/Aggregate.lean`. The theorem is proved for the corresponding measurable, bounded integrands; separate theorems handle empty groups and other boundary cases.

Proof outline. Use Fubini and a sum-coordinate shear to isolate the total in one group; apply the weighted simplex moment calculation to the corresponding slices. Repeat for the second group. The factors A_I, A_J record the Dirichlet normalizations; in particular the -1 in each aggregate exponent is essential. The polynomial cases `tri_poly` and `two_group_poly` yield explicit rational expressions, and `two_group_const` recovers the simplex moment formula when $h = 1$.

Proposition 7.2 (A cut-triangle certificate). *With $y = (y_0, y_1, y_2)$, the exact value is*

$$\int_{\substack{y_i \geq 0, y_0 + y_1 + y_2 \leq 1 \\ y_0 \leq 1/2}} y_0 y_2^2 dy = \frac{19}{7680}.$$

Lean: `tri_cut`, `example_value` (`Solution/Aggregate.lean`). The certificate reduces this integral to $\frac{1}{3} \int_{\Delta_2(1), u_1 \leq 1/2} u_1 u_2^3 du_1 du_2$; the value has also been cross-checked by direct rational integration.

Connection to marginal geometry. On the translated stratum $x = y + g \mathbf{1}_{R \sqcup K}$, define $u_1 = \sum_{i \in R} y_i$ and $u_2 = \sum_{i \notin R} y_i$. The proved correspondence `stratumEndpoint_shift` expresses the fibre endpoint as

$$U = \min(L - (r + k)g - u_1 - u_2, \max(g, C(r + 1) - rg - u_1)), \quad r = |R|, k = |K|.$$

The cap index belongs to the stratum R . Substituting the translated point's rough set $R \cup K$ instead changes the endpoint; the Table 3 control records the distinct values 587/1250 and 461/1250. The cap-branch polygon-cell integration is established in Section 8; all endpoint branches and the first marginal Gram entry are treated in Section 9.

8 Polygon-cell integration and the cap branch

The two-aggregate reduction produces planar regions bounded by coordinate thresholds and a total-sum constraint. For real A, B, D define the clipped cell

$$\mathcal{B}(A, B, D) = \{(u_1, u_2) \in \mathbb{R}^2 : 0 \leq u_1 \leq A, 0 \leq u_2 \leq B, u_1 + u_2 \leq D\}.$$

Theorem 8.1 (Exact polygon-cell moments). *For $\alpha, \beta \in \mathbb{N}$, the Lebesgue integral*

$$\int_{\mathcal{B}(A, B, D)} u_1^\alpha u_2^\beta du_1 du_2 = \text{boxVal}(A, B, D, \alpha, \beta)$$

where `boxVal` is a finite closed form using polynomial antiderivatives. For rational A, B, D its value is rational. The same construction gives rational integrals for arbitrary rational polynomials in two variables. Degenerate and empty cells are treated separately.

Lean: `box_monomial, boxVal_cast, box_poly, boxVal_empty, boxVal_nonpos, boxVal_rect; Solution/CapCell.lean`.

Proof outline. Integrate first in u_2 with upper endpoint $\min(B, D - u_1)$ and split the u_1 interval at $D - B$. The resulting one-variable powers have exact polynomial antiderivatives. In particular, when $A + B \leq D$ the diagonal constraint is redundant and the cell is a rectangle; the formula also covers diagonally truncated cells.

Theorem 8.2 (Cap-branch endpoint and piece integral). *On a translated stratum $x = y + g\mathbf{1}_{R \cup K}$, with $r = |R|$, $k = |K|$ and aggregates u_1, u_2 as in Section 7, the cap branch of the fibre endpoint is*

$$U = C(r + 1) - rg - u_1,$$

subject to its vertical and horizontal branch constraints. Combined with the piece constraints, the aggregate domain is a clipped cell $\mathcal{B}(A, B, D)$ with parameters determined by g, L, τ, C, r, k . The weighted piece integrals

$$\int_{\text{piece}} y^t \mathbf{1}_{\text{cap}} U^m dy$$

therefore equal an explicit expression obtained from the two-aggregate Dirichlet constants and rational cell integral formulas.

Lean: `endpointAgg_on_capCell, cap_branch_piece, cap_branch_r0, cap_inner; Solution/CapCell.lean`. The formalization proves the branch classification and corresponding endpoint equalities. For Table 3 the τ constraint is redundant on the cap branch: $\tau - (L - g) = 1/625 > 0$, and the cap cells become rectangles (`tau_ge_table3, capCell_rect_table3`).

Proposition 8.3 (A Table 3 cap-piece certificate). *For the specified 44-coordinate piece $(R, K) = (\{0\}, \emptyset)$, the cap-branch contribution to the integral defining $J_T(1)$ is*

$$\frac{(L - C(2))^{43}((C(2) - g)^3 - g^3)}{3 \cdot 43!} \approx 1.0464 \times 10^{-70}.$$

Lean: `capPieceT3_value (Solution/CapCell.lean)`. This is *one piece* of the inclusion-exclusion decomposition, not the complete value of $J_T(1)$. The exact rational certificate is kernel-checked. Polygon clipping and lower-dimensional integration provide independent numerical controls; the latter is an approximation, not part of the kernel proof. The remaining branches and their assembly are treated in Section 9.

The translated marginal endpoint is the minimum of the remaining total mass and the maximum of a cap expression and g :

Its switching lines partition each two-aggregate piece into four cells: the cap branch, the remaining-mass branch on the cap side, the constant g branch, and the remaining-mass branch on the g side. Each cell is represented by signed differences of the clipped polygon cells $\mathcal{B}(A, B, D)$ in Section 8.

$$u_1^\alpha u_2^\beta (p - u_1)^{m_1} (L' - u_1 - u_2)^{m_2}$$

Lean: box_twoPow, cells_integral, piece_block, piece_r0, indicator_supportT_strat, count_by_card; Solution/MarginalEntry.lean.

$$J_T(1) = (N+1) \int_{\widehat{T}} U(\widehat{x})^2 d\widehat{x} = \text{jSum}(N, g, L, \tau, C, 2, 0).$$
$$J_T[G_{0,\emptyset}, G_{0,\emptyset}] = 45 \int_{\widehat{T}} U(\widehat{x})^2 d\widehat{x} \approx 1.1193487646397893 \times 10^{-55}.$$

The exact value is the ratio of these integers, obtained by concatenating each integer's digit lines (no separators):

196405301342598493830210603911544843582810204
587055080769623409436245894353192939488221872
912132381844932553462651053065161718578698716
871038141

[illegible]

Lean: `jtSum_table3`, `jT_one_table3`, `eval_gen_zero`, `jT_gram_gen00_table3`; `Solution/MarginalEntryCert.lean`, `Solution/MarginalEntry.lean`. The proof evaluates 133 nonempty (r, k) pieces by kernel checking and proves the remaining pieces vanish. The exact certificate is regenerated byte-for-byte by `marginal_entry_v2f_emit.py`. Independent polygon clipping, low-dimensional integration, and Monte Carlo provide additional checks, but are not substitutes for Lean verification. This proves one marginal Gram entry, not the full Gram matrix or a positivity certificate.

10 Verification, limitations, and further mathematics

The project uses Lean `v4.34.0-rc2` and a pinned Mathlib revision `85e3a25`. At the V2f baseline, all 18 Comparator suites were independently accepted on the VPS. Four positive V2d controls were accepted; four intentionally incorrect controls were rejected with unsolved goals. The V2d proof commit is `52fd36a`; its published verification record is commit `90055db`. For V2e-A, five positive controls were accepted and four negative controls were rejected. The V2e-A proof commit is `01644e6`, with VPS verification/provenance commit `38606c0`. For V2e-B, five positive controls were accepted and five deliberately incorrect controls rejected; the V2e-B proof commit is `6a21780`, with verification/provenance commit `25a7376`. For V2f, four positive Lean-file controls were accepted and five negative controls rejected. Its proof commit is `8c50fb6`, with VPS provenance commit `da0c3fd`. The reported kernel axioms for these developments are `propext`, `Quot.sound`, and `Classical.choice`. Results and verification scripts are available in [specs/results](#).

Further work is needed for marginal J_T Gram entries containing power-sum factors and their general rational evaluation, optimized polynomial coefficients, exact variational positivity certificates, and remaining analytic hypotheses. The existing Lean development formalizes the *conditional* implication $\text{DHL}[45, 2] \Rightarrow H_1 \leq 212$; this manuscript does not claim an unconditional formal proof of $H_1 \leq 212$.

Acknowledgment and provenance

The bounded-prime-gap application is motivated by Axiom Math’s mathematical argument [1]. This manuscript records an independent Lean formalization of the specified results, with theorem names and commit identifiers to support reproducible review. Substantive exposition and attribution will be expanded in a subsequent draft.

References

- [1] Axiom Math, *A New Bound for Small Gaps Between Primes*, preliminary draft, 3 September 2026. Project page: <https://primegaps.axiommath.ai/>.
- [2] Reading Point, *lean-bgp212*: Lean source, verification records, and certificates. <https://github.com/thinkthoughts/lean-bgp212>.