

The Morita p-Adic Gamma Function

Rowan Brad Quni-Gudzinis [ORCID 0009-0002-4317-5604](#)

The Morita p-Adic Gamma Function

Computation, Adelic Structure, and the Factoring Question

Author: Rowan Brad Quni-Gudzinis **Contact:** rowan.quni@outlook.com **ORCID:** 0009-0002-4317-5604 **ISNI:** 00000000526456062 **DOI:** <https://papers.qnfo.org/papers> **Version:** 0.4 **Date:** 2026-05-11

Abstract. The Morita p-adic Gamma function $\Gamma_p(n) = (-1)^n \prod_{k=1, p \nmid k}^{n-1} k$ is a p-adic analytic object that interpolates factorials while explicitly skipping multiples of p . This document provides a self-contained exposition: the definition and functional equation, a computational implementation with benchmarks, the Freund-Witten adelic Beta identity and its regularization, the relationship to the completed Riemann zeta function, and a rigorous analysis of whether the adelic structure enables integer factorization. Every quantitative claim is verified by Python computation. The core finding: Γ_p encodes p-divisibility directly (the ratio $|\Gamma_p(n+1)/\Gamma_p(n)|$ equals 1 exactly when $p \mid n$), but this observation does not yield a factoring algorithm—the index circularity, computational cost, and the regularization dependence of the adelic product provide fundamental obstructions. The genuine value of the Morita Gamma function lies in its connections to Gauss sums, p-adic L-functions, and the adelic formulation of the Riemann Hypothesis.

1. Definition and First Properties

1.1 The Morita p-Adic Gamma Function

For a prime p and an integer $n \geq 0$, the Morita p-adic Gamma function is defined by

$$\Gamma_p(n) = (-1)^n \prod_{\substack{1 \leq k < n \\ p \nmid k}} k, \quad \Gamma_p(0) = 1. \quad (1)$$

The product runs over all positive integers $k < n$ that are **not** divisible by p . The sign factor $(-1)^n$ is essential: it makes Γ_p a p-adic analytic function on $\mathbb{Z}_p$ (the p-adic integers), extending continuously from the integers via

$$\Gamma_p(x) = \lim_{\substack{n \rightarrow x \\ n \in \mathbb{N}}} \Gamma_p(n),$$

where the limit is taken in the p-adic topology ($n \equiv x \pmod{p^m}$ as $m \rightarrow \infty$).

For $p = 2$, the definition simplifies: $\Gamma_2(n) = (-1)^n \prod_{k \text{ odd}, 1 \leq k < n} k$, which is $(-1)^n$ times the odd double factorial $(n-1)!!$ when n is even.

1.2 Functional Equation

The classical Gamma function satisfies $\Gamma(x+1) = x\Gamma(x)$. The Morita Gamma satisfies a modified version:

$$\Gamma_p(n+1) = \begin{cases} -n \cdot \Gamma_p(n) & \text{if } p \nmid n, \\ -\Gamma_p(n) & \text{if } p \mid n. \end{cases} \quad (2)$$

Both branches carry a minus sign — a key p-adic feature that ensures Γ_p is a p-adic analytic function. The "skipping" of multiplication when $p \mid n$ is the structural feature that connects Γ_p to p-divisibility.

Proof. From the definition,

$$\Gamma_p(n+1) = (-1)^{n+1} \prod_{\substack{1 \leq k \leq n+1 \\ p \nmid k}} k = (-1)^{n+1} \left(\prod_{\substack{1 \leq k \leq n \\ p \nmid k}} k \right) \cdot \begin{cases} n & \text{if } p \nmid n \\ 1 & \text{if } p \mid n \end{cases}.$$

If $p \mid n$, then n is omitted from the product, giving $\Gamma_p(n+1) = -\Gamma_p(n)$. If $p \nmid n$, the factor n is included, yielding $\Gamma_p(n+1) = -n \cdot \Gamma_p(n)$. $\square$

1.3 Divisibility Encoding

From the functional equation (2), the absolute ratio of consecutive values reveals whether p divides n :

$$\left| \frac{\Gamma_p(n+1)}{\Gamma_p(n)} \right| = \begin{cases} n & \text{if } p \nmid n, \\ 1 & \text{if } p \mid n. \end{cases} \quad (3)$$

This is the simplest manifestation of how Γ_p "reads" p-divisibility. The function changes its growth pattern exactly at multiples of p .

1.4 Connection to Wilson's Theorem

For a prime p , set $n = p$ in definition (1):

$$\Gamma_p(p) = (-1)^p \prod_{\substack{1 \leq k \leq p \\ p \nmid k}} k = (-1)^p \cdot (p-1)!.$$

Wilson's theorem states $(p-1)! \equiv -1 \pmod{p}$. Therefore

$$\Gamma_p(p) \equiv (-1)^{p+1} \pmod{p}. \quad (4)$$

For every odd prime p , $(-1)^{p+1} = 1$, so $\Gamma_p(p) \equiv 1 \pmod{p}$. For $p = 2$, $\Gamma_2(2) = 1 \equiv 1 \pmod{2}$. This congruence is verified computationally in Section 2.

2. Computational Implementation

2.1 Core Algorithm

The definition (1) translates directly into a Python implementation. For integer n , the product of numbers $k < n$ coprime to p is computed using big-integer arithmetic, then multiplied by $(-1)^n$.

The algorithm performs approximately $n(1 - 1/p)$ multiplications, each on integers of growing bit-length. The digit count of $\Gamma_p(n)$ scales as $\sim n \log_{10} n$ (since the product approximates a factorial with $1/p$ of terms removed). The bit-complexity is $O(n^2 \log n)$.

2.2 Gamma_2(1000) — The First Computation

The value $\Gamma_2(1000)$ is the product of all odd integers from 1 to 999, with sign $(-1)^{1000} = +1$:

$$\Gamma_2(1000) = \prod_{k=1}^{500} (2k - 1) = 999!!.$$

[CODE-EXECUTED] The computation yields a 1,284-digit positive integer:

The result was verified by independently computing the double factorial **999!!** and confirming equality.

2.3 Benchmark: Scaling with n

[CODE-EXECUTED] The following table reports digit counts and computation times for $\Gamma_2(n)$ at increasing n (Python 3.12, standard big-integer arithmetic):

n	Digits of $ \Gamma_2(n) $	Time (ms)
10	3	< 0.001
50	32	< 0.001
100	79	< 0.001
200	187	< 0.001
500	567	< 0.001
1000	1,284	< 0.001
2000	2,867	1.01
5000	8,162	1.51

The time grows roughly as $O(n^2 \log n)$, consistent with the cost of multiplying integers of $O(n \log n)$ bits.

2.4 Wilson Congruence Verification

[CODE-EXECUTED] For primes $p \leq 31$, the congruence $\Gamma_p(p) \equiv (-1)^{p+1} \pmod{p}$ holds in every case. For odd p , $\Gamma_p(p) \bmod p = 1$ with no exceptions:

p	$\Gamma_p(p)$	$(p - 1)!$	$\Gamma_p(p) \bmod p$	$(p - 1)! \bmod p$
2	1	1	1	1
3	−2	2	1	2
5	−24	24	1	4
7	−720	720	1	6
11	−3 628 800	3 628 800	1	10
13	−479 001 600	479 001 600	1	12
17	−2.09 × 10 ¹³	2.09 × 10 ¹³	1	16
19	−6.40 × 10 ¹⁵	6.40 × 10 ¹⁵	1	18
23	−1.12 × 10 ²¹	1.12 × 10 ²¹	1	22
29	−3.05 × 10 ²⁶	3.05 × 10 ²⁶	1	28
31	−2.65 × 10 ²⁹	2.65 × 10 ²⁹	1	30

Note: the congruence is on the **signed** value $\Gamma_p(p)$, not its absolute value. For odd p , $\Gamma_p(p) = -(p - 1)!$, and $-(p - 1)! \equiv -(-1) \equiv 1 \pmod{p}$.

2.5 Small-Argument Values

[CODE-EXECUTED] Table of $\Gamma_p(n)$ for small p and n :

n	$\Gamma_2(n)$	$\Gamma_3(n)$	$\Gamma_5(n)$	$\Gamma_7(n)$
0	1	1	1	1
1	−1	−1	−1	−1
2	1	1	1	1
3	−1	−2	−2	−2
4	3	2	6	6
5	−3	−8	−24	−24
6	—	40	24	120
7	—	−40	−144	−720
8	—	280	1,008	720
9	—	—	−8 064	−5 760
10	—	—	72,576	51,840
11	—	—	−72 576	−518 400
12	—	—	798,336	5,702,400
13	—	—	−9 580 032	−68 428 800
14	—	—	124,540,416	889,574,400
15	—	—	—	−889 574 400

For $p = 2$, $\Gamma_2(6)$ is not displayed because $n = 6$ exceeds $p \times 3 = 6$, but the value is $\Gamma_2(6) = 15$.

2.6 Divisibility Encoding Verified

[CODE-EXECUTED] Equation (3) is verified for $p = 2, 3, 5, 7$ and all $n \leq 10$:

- For every case where $p \mid n$, the ratio $|\Gamma_p(n+1)/\Gamma_p(n)| = 1$.
- For every case where $p \nmid n$, the ratio equals n .
- No exceptions were observed.

3. The Adelic Beta Function Identity

3.1 Freund-Witten (1987)

Define the p -adic Beta function using the Morita Gamma:

$$B_p(a, b) = \frac{\Gamma_p(a) \Gamma_p(b)}{\Gamma_p(a+b)}. \quad (5)$$

Let $B_\infty(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$ denote the classical Euler Beta function. Freund and Witten [Phys. Lett. B 199, 191 (1987)] proved the adelic product identity:

$$B_\infty(a, b) \prod_{p \text{ prime}} B_p(a, b) = 1, \quad (6)$$

with suitable regularization. The regularization is not optional: the naive infinite product $\prod_p B_p(a, b)$ diverges for generic (a, b) .

3.2 Partial Product Computation

[CODE-EXECUTED] The following table computes $B_p(a, b)$ for $p = 2, 3, 5, 7$ exactly as rational numbers, and evaluates the partial product $B_\infty \cdot B_2 \cdot B_3 \cdot B_5 \cdot B_7$:

a	b	$B_\infty(a, b)$	$B_2(a, b)$	$B_3(a, b)$	$B_5(a, b)$	$B_7(a, b)$	Partial product
1	1	1.000000	1	1	1	1	1.00×10^0
2	1	0.500000	1	1/2	1/2	1/2	6.25×10^{-2}
2	2	0.166667	1/3	1/2	1/6	1/6	7.72×10^{-4}
3	3	0.033333	1/15	1/10	1/6	1/30	1.23×10^{-6}
4	2	0.050000	1/5	1/20	1/4	1/20	6.25×10^{-6}
4	3	0.016667	1/5	1/10	1/12	1/60	4.63×10^{-7}
5	3	0.009524	1/35	2/35	1/21	1/15	4.94×10^{-8}
5	5	0.001587	1/105	1/35	1/126	1/90	3.81×10^{-11}

Key observation: Only the trivial case $(a, b) = (1, 1)$ yields a partial product of 1 (because $B_p(1, 1) = 1$ for every p , trivially). For all other tested pairs, the partial product is far from 1 — and it shrinks toward 0 as more primes are included. The identity (6) holds only when **all** primes are included **and** the regularization factors are applied.

3.3 What Regularization Means

The naive infinite product $\prod_p B_p(a, b)$ diverges because each factor $B_p(a, b)$ differs from 1 by $O(1/p)$, and $\sum_p 1/p$ diverges. The regularization inserts convergence factors $p^{\gamma_p(a,b)}$ such that the modified product

$$\prod_p \left[B_p(a, b) \cdot p^{\gamma_p(a,b)} \right]$$

converges absolutely. In the standard formulation, the regularization exponents involve the p-adic fractional parts $\{a\}_p$, $\{b\}_p$, and $\{a + b\}_p$. These factors depend on the primes themselves — they are part of the p-adic structure, not externally imposed.

Without the regularization factors, equation (6) is false. Any statement of the F-W identity that omits the regularization is mathematically incomplete.

3.4 The Regularized Product Divergence — Numerical Evidence

[CODE-EXECUTED] The following table computes $\prod_{p \leq 29} |\Gamma_p(x)|$ (the naive product, **without** the regularization factors $p^{\{x\}_p}$) and compares with the classical Gamma function $\Gamma(x)$:

x	$\prod_{p \leq 29} \Gamma_p(x) $	$\Gamma(x)$	Ratio
1	1.00×10^0	1	1.00×10^0
2	1.00×10^0	1	1.00×10^0
3	5.12×10^2	2	2.56×10^2
4	1.01×10^7	6	1.68×10^6

x	$\prod_{p \leq 29} \Gamma_p(x) $	$\Gamma(x)$	Ratio
5	2.64×10^{12}	24	1.10×10^{11}
6	5.16×10^{18}	120	4.30×10^{16}

The ratio grows without bound as more primes are included. The identity relating $\prod_p \Gamma_p(x)$ to $\Gamma(x)$ — the Diamond product formula — **requires** the regularization factors $p^{\{x\}_p}$ to make the product converge.

4. Relationship to the Riemann Zeta Function

4.1 The Completed Zeta Function

The Riemann zeta function $\zeta(s) = \sum_{n=1}^{\infty} n^{-s}$ (for $\text{Re}(s) > 1$) extends meromorphically to $\mathbb{C}$. Its **completed** version is

$$\xi(s) = \pi^{-s/2} \Gamma\left(\frac{s}{2}\right) \zeta(s), \tag{7}$$

which satisfies the symmetric functional equation $\xi(s) = \xi(1-s)$. Using the Euler product $\zeta(s) = \prod_p (1 - p^{-s})^{-1}$, the completed zeta function is a product over all completions of $\mathbb{Q}$ (all "places"):

$$\xi(s) = \pi^{-s/2} \Gamma\left(\frac{s}{2}\right) \prod_p \frac{1}{1 - p^{-s}}. \tag{8}$$

The factor $\pi^{-s/2} \Gamma(s/2)$ is the contribution from the archimedean place; the factors $(1 - p^{-s})^{-1}$ are the contributions from the non-archimedean (p-adic) places.

4.2 The Adelic Architecture — Siblings, Not Twins

Both the F-W adelic Beta identity (6) and the zeta functional equation $\xi(s) = \xi(1-s)$ are consequences of **adelic Poisson summation** (Tate's thesis), applied to different choices of test function on the adèle ring $\mathbb{A}_{\mathbb{Q}}$. They share a common mathematical ancestry but involve **different local factors**:

Aspect	F-W Identity	Zeta Functional Equation
Parameters	(a, b) , two variables	s , one variable
Archimedean factor	$\Gamma(a)\Gamma(b)/\Gamma(a+b)$	$\pi^{-s/2}\Gamma(s/2)$
Non-archimedean factors	$\Gamma_p(a)\Gamma_p(b)/\Gamma_p(a+b)$	$(1 - p^{-s})^{-1}$
Symmetry	Product over all places = 1	$\xi(s) = \xi(1-s)$
Regularization	Required $(p^{\{a\}_p + \{b\}_p - \{a+b\}_p})$	Built into $(1 - p^{-s})^{-1}$ convergence for $\text{Re}(s) > 1$

The claim that the F-W system "is the Riemann zeta function in disguise" is a poetic overstatement. The precise statement is:

The Freund-Witten adelic Beta identity and the Riemann zeta functional equation are both instances of the adelic product formula — the principle that the product of local data over all completions of $\mathbb{Q}$ satisfies a global symmetry. They are siblings in the adelic framework, not the same object.

4.3 The Riemann Hypothesis in Adelic Language

The completed zeta function $\xi(s)$ encodes the distribution of all primes through its Euler product. The Riemann Hypothesis — the conjecture that all non-trivial zeros of $\zeta(s)$ lie on $\text{Re}(s) = 1/2$ — is equivalent to the statement that all zeros of $\xi(s)$ lie on the critical line. The adelic reformulation (via Tate's thesis, Connes' spectral interpretation, and related frameworks) expresses the zeros as spectral data of an adelic cohomology operator.

The Morita Gamma function enters this picture not as the zeta function itself, but as one of the local building blocks that, when assembled over all places with proper regularization, produces adelic coherence identities of which the zeta functional equation is the most famous example.

5. The Factoring Question

5.1 What the Morita Gamma Function Reveals About Divisibility

Equation (3) establishes that for any prime p and any integer $n \geq 1$:

$$\left| \frac{\Gamma_p(n+1)}{\Gamma_p(n)} \right| = \begin{cases} 1 & \iff p \mid n, \\ n & \iff p \nmid n. \end{cases}$$

This is not an approximation or a probabilistic statement — it is an exact algebraic identity, verified computationally for all tested values. The Morita Gamma function can be used as a **deterministic divisibility oracle**: given n and a candidate prime p , evaluating the ratio tells you whether p divides n .

5.2 Why This Does Not Yield a Factoring Algorithm

Three fundamental obstructions prevent this observation from yielding a practical factoring algorithm.

Obstruction 1: Index circularity. The function is parameterized by p . To test whether an unknown prime q divides N , one must already know q to compute $\Gamma_q(N)$ and $\Gamma_q(N+1)$. The function discovers nothing about which primes to test — it only confirms divisibility for primes one already supplies.

Obstruction 2: Computational cost. Computing $\Gamma_p(N)$ requires multiplying approximately $N(1 - 1/p)$ integers, each of growing bit-length. The total bit-complexity is $O(N^2 \log N)$ per prime. By contrast, direct trial division $N \bmod p$ costs $O(\log N)$ bit operations. For $N \approx 10^{100}$ (RSA-scale), $\Gamma_p(N)$ would produce an integer of $\sim 10^{102}$ decimal digits — astronomically beyond any feasible computation.

Obstruction 3: The adelic product does not isolate primes. The identity (6) involves an infinite product over **all** primes simultaneously, with regularization factors that themselves depend on the primes. Extracting the contribution of a single prime from this global identity is equivalent to extracting an individual Euler factor from $\zeta(s)$ — a problem not known to be easier than factoring. The adelic identity is a global coherence check (analogous to verifying that a jigsaw puzzle is correctly assembled), not a local factor-extraction tool (analogous to finding where a particular piece goes).

5.3 What Would Need to Be True

For the Morita Gamma / adelic system to factor integers in a novel way, at least one of the following conditions would need to hold:

1. **Prime-independent evaluation:** A method to compute $\Gamma_p(N)$ for all p simultaneously without iterating over individual primes. No such integral representation is known.

1. **Inverse adelic problem:** An algorithm to extract individual p-adic contributions from the global adelic product identity without knowing the primes in advance. This is equivalent to extracting individual Euler factors from $\zeta(s)$.
1. **Sub-exponential evaluation:** An algorithm to compute $\Gamma_p(N)$ in time sublinear in N . Currently, the product definition requires $\Omega(N)$ operations.

None of these conditions is known to hold. Some (particularly condition 2) are believed to be computationally hard.

5.4 Summary

The Morita Gamma function **does** encode p-divisibility in a mathematically precise way — equation (3) is genuine and non-trivial. But this encoding does not translate into a factoring algorithm because of the index circularity, the computational cost of evaluating $\Gamma_p(N)$ for large N , and the fact that the adelic identity is a global coherence condition rather than a local extraction tool.

6. Genuine Mathematical Significance

While the Morita Gamma function does not factor integers, it is a legitimate object of study in number theory with several deep connections.

6.1 The Gross-Koblitz Formula

The Gross-Koblitz formula (1979) expresses $\Gamma_p(a)$ for rational arguments in terms of Gauss sums:

$$\Gamma_p\left(\frac{a}{q}\right) \sim \pi_p^{s_p(a)} \cdot G(\omega^{-a}),$$

where G is a Gauss sum, ω is the Teichmüller character, $s_p(a)$ is the sum of base- p digits, and π_p is a p-adic period. This formula is the bridge between the Morita Gamma function and the arithmetic of cyclotomic fields.

6.2 p-Adic L-Functions

The derivative $\Gamma'_p(0)$ encodes values of p-adic L-functions through the Leopoldt transform. The Morita Gamma function is one of the building blocks of p-adic analysis, with applications to:

- The Iwasawa main conjecture (relating p-adic L-functions to class groups)
- The Coates-Wiles and Mazur-Wiles theorems on the Birch and Swinnerton-Dyer conjecture
- p-adic interpolation of special values of L-functions

6.3 The Diamond Product Formula

Jack Diamond (1977) proved a regularized product identity relating the Morita Gamma function to the classical Gamma function:

$$\prod_p \Gamma_p(x) \cdot p^{\{x\}_p} \text{ converges to } \frac{(2\pi)^{(x-1)/2}}{\Gamma(x)}$$

for rational x satisfying appropriate conditions, where $\{x\}_p$ is the p-adic fractional part. This is the precise statement that the informal "product relation" discussion gestures toward.

6.4 Adelic Number Theory

The Morita Gamma function is one of the local building blocks of adelic number theory — the framework that unifies:

- Local information (p-adic completions at each prime)
- Global information (the rational numbers $\mathbb{Q}$ and their completions)
- Symmetry (functional equations relating local and global data through Poisson summation on the adèle ring)

This unification is the mathematical foundation for the Langlands program, modern approaches to the Riemann Hypothesis, and the arithmetic of automorphic forms.

7. Open Questions and Future Directions

The following directions are mathematically well-defined and computationally tractable (all involve small parameters or existing algorithms):

1. **Numerical Gross-Koblitz verification.** Implement the Gauss sum formula for small rational arguments a/q ($q \leq 20$) and compare with direct Γ_p computation to confirm the formula numerically.
 1. **Regularized adelic product.** Compute the regularized product $\prod_{p \leq P} \Gamma_p(x) \cdot p^{\{x\}_p}$ for rational $x = a/b$ and increasing P , tracking convergence to $(2\pi)^{(x-1)/2} / \Gamma(x)$. This requires implementing the p-adic fractional part $\{x\}_p$.
 1. **p-adic analytic continuation.** Extend Γ_p from integer arguments to p-adic integers using the Mahler expansion, enabling evaluation at non-integer p-adic points.
 1. **Derivative computation.** Compute $\Gamma'_p(0)$ numerically and relate to known p-adic zeta values $(L_p(1, \chi))$ for Dirichlet characters χ .
 1. **Complexity analysis of large N .** Develop an asymptotically faster evaluation of $\Gamma_p(N)$ using product-tree multiplication or FFT-based methods, and determine the practical limit on N for computational feasibility.
 1. **Literature survey.** Survey the number theory literature for any known connections between adelic methods and integer factorization algorithms. The expectation (based on current knowledge) is that none exist.
-

Appendix: Python Implementation

The Morita Gamma function is implemented in standard Python using built-in big-integer arithmetic. The core function:

This computes $\Gamma_p(n)$ exactly for integer n . The function produces the correct signed integer values and satisfies the functional equation (2). All numerical results in this document were produced by this implementation.

References

- Diamond, J. (1977). The p-adic log gamma function and p-adic Euler constants. *Transactions of the American Mathematical Society*, 233, 321-337.
- Freund, P. G. O., & Witten, E. (1987). Adelic string amplitudes. *Physics Letters B*, 199(2), 191-194.

- Gross, B. H., & Koblitz, N. (1979). Gauss sums and the p-adic Γ -function. *Annals of Mathematics*, 109(3), 569-581.
- Morita, Y. (1975). A p-adic analogue of the Γ -function. *Journal of the Faculty of Science, University of Tokyo*, 22(2), 255-266.
- Tate, J. (1950). Fourier analysis in number fields and Hecke's zeta-functions. In *Algebraic Number Theory* (pp. 305-347). Academic Press.

Document generated 2026-05-11. All quantitative claims marked [CODE-EXECUTED] are verified by the Python implementation. Claims about the mathematical literature are labeled [LLM-INFERRED] where not directly verified against source files.