Factorized Harmonic Numbers and Logarithms by a Forward Arithmetic Recursion

Michael T. M. Emmerich

26 October 2026
Revised 8 November 2026 at 06:08 UTC

Abstract

The fifth harmonic number is 137/60. Its denominator is the least common multiple of the first five integers, but how does a construction starting at 1 discover the prime 137 in its numerator? We give a forward arithmetic recursion that produces the prime factorization of every harmonic number in one uninterrupted run. Its purpose is to make visible how the factors of a sum of reciprocals arise and cancel. The starting point is to view a reciprocal as the frequency with which a divisor occurs. Counting all divisors up to a given bound over a common period then gives the corresponding harmonic number as the mean count per step. The total count over that period has a second role. It tells the recursion how far to continue until it has generated every prime it needs for the numerator. For the fifth harmonic number, the count is 137 after 60 steps. At step 137, the recursion recognizes the count as prime. It discovers every prime it needs in the same way, without a supplied list. Each step uses integer counters and prime exponents from rows stored at earlier steps. On these valuation rows the recursion needs only componentwise addition, subtraction and comparison. The same divisor counts reveal a finite arithmetic identity behind the multiplication law for logarithms. When two adjacent blocks of reciprocals are joined, their sums add and the ratios of their endpoints multiply. The starting index of the second block matters at every finite stage. In the limit, the block sum depends only on its endpoint ratio and gives the natural logarithm of that ratio. At successive powers of two, signed divisor counts give a decreasing sequence of rational approximations to the Euler-Mascheroni constant, with their factorizations already available when counting ends. The divisor sum and harmonic number in Lagarias's inequality also arise from the same divisibility data, and we offer a brief interpretation.

Keywords: harmonic numbers; divisor count; prime valuations; discrete dynamics; logarithms; Euler–Mascheroni constant; Lagarias's inequality; Riemann hypothesis.

We deliberately slow down computation to make structure visible.

1 Introduction

For a positive integer N, the harmonic number $H_N = \sum_{m=1}^{N} 1/m$ is a sum of reciprocals with two complementary aspects. Its size grows logarithmically, while its reduced numerator and denominator record prime factors and their cancellation. We seek a construction that explains how these aspects arise together from divisibility. In particular, we want to see how reciprocal contributions accumulate, how the primes needed for a harmonic value become available, and how addition of finite harmonic blocks gives rise to the logarithmic laws.

The purpose of proceeding slowly is to expose these intermediate steps. The familiar update $H_{N+1} = H_N + 1/(N+1)$ determines the next rational value, but it does not specify how to generate the primes in its factorization. Here the construction is supplied with no prime list or factorizations. It must generate its own arithmetic information and use each item only after that item has become available. We distinguish the harmonic index N from the integer time $t = 1, 2, \ldots$ at which this information is produced. A *tick* is one step $t \mapsto t + 1$.

To see what arithmetic information is required, write $\nu_p(x)$ for the exponent of a prime p in a positive rational number x. In the reduced fraction, a positive exponent belongs to the numerator, a negative one to the denominator, and zero means that p is absent. The list $(\nu_p(x))_p$ is its *signed valuation row*. For instance, $12 = 2^2 \cdot 3$ has entries $+2$ and $+1$ in the columns 2 and 3, whereas $H_3 = 11/6$ has entries $-1, -1, +1$ in the columns $2, 3, 11$. Table 1 places the rows of N and H_N next to each other. The integer row has no prime factor greater than N; the harmonic row already requires 11 at $N = 3$, 137 at $N = 5$, and 761 at $N = 8$. Generating the next integer row and obtaining the next harmonic row therefore involve different requirements on the available primes.

	Integer: $\nu_p(N)$							Harmonic number: $\nu_p(H_N)$							
N	2	3	5	7	11	137	761	2	3	5	7	11	137	761	H_N
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	+1	0	0	0	0	0	0	−1	+1	0	0	0	0	0	$\frac{3}{2}$
3	0	+1	0	0	0	0	0	−1	−1	0	0	+1	0	0	$\frac{11}{6}$
4	+2	0	0	0	0	0	0	−2	−1	+2	0	0	0	0	$\frac{25}{12}$
5	0	0	+1	0	0	0	0	−2	−1	−1	0	0	+1	0	$\frac{137}{60}$
6	+1	+1	0	0	0	0	0	−2	0	−1	+2	0	0	0	$\frac{49}{20}$
7	0	0	0	+1	0	0	0	−2	+1	−1	−1	+2	0	0	$\frac{363}{140}$
8	+3	0	0	0	0	0	0	−3	0	−1	−1	0	0	+1	$\frac{761}{280}$

Table 1: Signed valuation columns for N and H_N, $1 \le N \le 8$. Blue cells show positive exponents, orange cells negative exponents; the signs also make the table readable without colour. The same prime labels are used in both panels, and every omitted prime has valuation zero throughout. Each line compares values at the same index N; it does not assert that H_N is available at clock time $t = N$. The construction below generates the prime labels itself and produces the harmonic rows at later clock times (except $H_1 = 1$).

We now explain the clock terminology. A *clock of period* p is a counter whose hand advances through $0, 1, \ldots, p-1$ and then returns to zero. A return to zero is called a *ring*. To *install* a clock means to add this counter to the running collection. Initially, at $t = 1$, the collection is empty. At each new tick $t \geq 2$, all existing hands advance. If none rings, a clock of period $p = t$ is installed with its hand at zero; it rings again at $2p, 3p, \ldots$. This rule discovers precisely the primes: a composite tick has an earlier prime divisor whose clock rings, whereas a prime tick has none. For example, clocks are installed at 2 and 3, but not at 4, when the period-2 clock rings.

The same process records the valuation row of each integer it reaches. A count of completed cycles, together with previously stored rows, supplies the exponents by a recursion specified below. The term *native generator* refers to this common process for generating primes and integer valuation rows. For a rational number u/v, subtracting the row of v from that of u records cancellation and gives the signed row of the reduced fraction: $\nu_p(u/v) = \nu_p(u) - \nu_p(v)$.

We now attach a counting process to a fixed harmonic index $N \geq 1$. This process, called an *observer*, counts one *divisor event* for each integer $m \in \{1, \ldots, N\}$ that divides the current tick t. Several events may occur at the same tick. These divisibility tests can be made from the generated rows: $m \mid t$ exactly when every prime exponent of m is at most the corresponding exponent of t. Only $m \leq t$ can contribute, so the rows needed at tick t are already available. Divisor 1 contributes once at every tick through its zero valuation row; it is not installed as a prime clock. The first tick divisible by every integer from 1 to N is

$$L_N = \operatorname{lcm}(1, \ldots, N),$$

the least common multiple. We call this first common multiple the *synchronization time*. It is detected by the divisibility tests; its value is not supplied in advance.

During the ticks $1, \ldots, L_N$, each possible divisor m contributes exactly L_N/m events. Its average contribution per tick is thus $1/m$. Adding all contributions, including $m = 1$, gives the full event count

$$B_N := \sum_{m=1}^{N} \frac{L_N}{m} = L_N H_N, \qquad H_N = \frac{B_N}{L_N}.$$

The equality identifies the count produced by divisibility tests; it does not presuppose a known harmonic value. For $N = 3$, the first six ticks contain six multiples of 1, three of 2, and two of 3. The eleven events give the average $11/6 = 1 + 1/2 + 1/3$.

This counting identity alone does not settle when the complete prime factorization of H_N is available. If $H_N = a_N/b_N$ is reduced, then b_N divides L_N, but a prime factor of a_N can exceed L_N:

$$H_3 = \frac{11}{6}, \quad 11 > L_3 = 6, \qquad H_5 = \frac{137}{60}, \quad 137 > L_5 = 60.$$

The native generator discovers prime p only at tick p. Stopping it at synchronization would therefore leave some required numerator primes undiscovered. This is the reason for continuing the clock after the event count is complete.

At synchronization, the observer freezes the full count B_N as its *target*. Since divisor 1 alone contributes L_N events, this target is at least the current time. The native clock now continues until $t = B_N$, called the *release time* T_N. During this *realization phase*, the observer represents t/L_N by the current integer row minus the stored row of L_N. Each

successor tick changes the vector by $\mathcal{V}(t+1) - \mathcal{V}(t)$, where $\mathcal{V}(u)$ denotes the valuation row of integer u. Every required prime is already available, because the two integers t and L_N are at most the current time. At release,

$$T_N = B_N, \qquad \frac{T_N}{L_N} = H_N.$$

The remaining number of ticks is the derived integer $R_N(t) = B_N - t$; it need not be stored as a separate state. Thus N specifies the harmonic number, L_N ends collection, and the count B_N itself is the destination of the forward clock. For H_3, the observer knows the integer 11 as a count at tick 6, but its valuation row is generated only at tick 11. At that tick the prime 11 is discovered and 11/6 is released in factorized form.

Section 2 develops this construction and its invariants. Theorem 2.3 shows that all harmonic indices can share a single uninterrupted run. The underlying prime and valuation generator is the prime-clock mechanism of [4, §§2–3]. The present construction adds event collection, realization, and a shared initialization: observer N is created at tick N with a count that already includes all contributions through that tick. No earlier tick needs to be replayed. Their integration makes the availability of the required arithmetic information explicit.

Section 3 turns from individual harmonic values to the logarithmic structure already visible between finite blocks. For integers $q \geq 2$ and $N \geq 1$, define

$$D_q(N) = H_{qN} - H_N = \sum_{m=N+1}^{qN} \frac{1}{m}.$$

Splitting a block at the intermediate endpoint aN gives

$$\boxed{D_{ab}(N) = D_a(N) + D_b(aN)} \qquad (a, b \geq 2).$$

For example, $D_6(N) = D_2(N) + D_3(2N)$ splits the block at $2N$. Doubling takes the index from N to $2N$, and tripling takes it from $2N$ to $6N$. The two sums add, and their endpoint ratios multiply. For fixed q, the limit of $D_q(N)$ as $N \to \infty$ is $\log q$. The finite identity thus yields $\log(ab) = \log a + \log b$. The section gives error bounds for this passage and derives the quotient and integer-power laws.

Section 4 uses blocks between 2^k and 2^{k+1} for $k \geq 0$, called *dyadic blocks*, to obtain decreasing rational approximations to Euler's constant $\gamma_{\mathrm{EM}} = \lim_{N\to\infty}(H_N - \log N)$. Signed divisor counts generate these approximations directly: at synchronization both integer rows needed for their factorization have already appeared. We identify their exact relation to partial sums of a known series involving binary digit counts [7, equation (6)]. Section 5 relates the same divisibility information to sums of positive divisors. Lagarias's inequality [6, Theorem 1.1] provides a setting for further questions about these arithmetic structures. Section 6 compares the information flow with Boyd's second method, which works at finite precision for a selected prime and can require recomputation at higher precision [1, §5.2]. Appendices A and B give the complete proofs of the native generator and the simultaneous observer schedule.

2 Factorized harmonic numbers on one clock

2.1 Reciprocals as event frequencies

For $N \geq 1$, define the number of divisor events at tick s by

$$d_N(s) = \sum_{m=1}^{N} \mathbf{1}_{\{m|s\}}. \tag{1}$$

Here $\mathbf{1}_{\{m|s\}}$ is 1 if m divides s, and 0 otherwise. In particular, divisor 1 contributes at every tick. Several divisors may contribute simultaneously, and every contribution is counted. For $N = 3$, the first six ticks give

Tick s	1	2	3	4	5	6	Total
Divisor 1	1	1	1	1	1	1	6
Divisor 2	0	1	0	1	0	1	3
Divisor 3	0	0	1	0	0	1	2
All events	1	2	2	2	1	3	11

Tick 6 contributes three events. Counting by columns or by rows gives

$$\frac{1+2+2+2+1+3}{6} = \frac{6+3+2}{6} = 1 + \frac{1}{2} + \frac{1}{3}.$$

The fractions add because the contributions of the divisors add. The average is taken over six ticks, not over the three possible divisors.

Lemma 2.1 (The event average)**.** *For $N \geq 1$ and every integer $a \geq 0$,*

$$\sum_{s=a+1}^{a+L_N} d_N(s) = L_N H_N. \tag{2}$$

Write $C_N(t) = \sum_{s=1}^{t} d_N(s)$ for the running collection count. Its value at synchronization is the positive integer

$$\boxed{B_N := C_N(L_N) = L_N H_N.} \tag{3}$$

It satisfies $B_N \geq L_N$, with equality exactly when $N = 1$.

Proof. Since each $m \leq N$ divides L_N, exactly L_N/m integers in an interval of L_N consecutive ticks are multiples of m. Interchanging the finite sums gives

$$\sum_{s=a+1}^{a+L_N} d_N(s) = \sum_{m=1}^{N} \frac{L_N}{m} = L_N H_N.$$

The divisor 1 supplies L_N events, and $N > 1$ supplies further events. □

The counts $d_N(1), \ldots, d_N(L_N)$ form row N of OEIS A138553 [8]; the row sum B_N is term N of A025529 [9]. Operationally, B_N is obtained by counting events. The product $L_N H_N$ identifies that count after the fact; the process does not evaluate a harmonic number or perform addition on its factorization. Including divisor 1 gives the direct quotient $H_N = B_N/L_N$.

2.2 The count as a forward target

For a prime p and a positive integer u, let $\nu_p(u)$ be the exponent of p in u, and put

$$\mathcal{V}(u) = (\nu_p(u))_p, \qquad \mathcal{V}(1) = \mathbf{0}.$$

For a positive rational $r = u/v$, write $\boldsymbol{\nu}(r) = \mathcal{V}(u) - \mathcal{V}(v)$. Unique factorization makes this independent of the expression for r. A finitely supported integer vector $w = (w_p)_p$ represents the reduced fraction

$$\frac{\prod_{w_p>0} p^{w_p}}{\prod_{w_p<0} p^{-w_p}}. \tag{4}$$

Its numerator and denominator are coprime because their prime supports are disjoint. Negative entries therefore record denominator factors directly.

Suppose for the moment that row $\mathcal{V}(t)$ is published at integer tick t. The next subsection explains how these rows and the synchronization signal are generated. At the first synchronization, freeze the collection count as the target and initialize the signed row:

$$B_N = C_N(L_N), \qquad \mathbf{W}_N(L_N) = \mathbf{0}. \tag{5}$$

While $t < B_N$, publish the next native row and apply

$$\boxed{\begin{aligned} t &\longmapsto t+1, \\ \mathbf{W}_N(t+1) &= \mathbf{W}_N(t) + \mathcal{V}(t+1) - \mathcal{V}(t). \end{aligned}} \tag{6}$$

At $t = B_N$, emit $\mathbf{W}_N(t)$ and stop the observer; the common clock continues. The target is an integer counter value. Its factorization is not requested when the count is frozen.

Proposition 2.2 (Native realization of the event average)**.** *The recursion terminates at $T_N = B_N = L_N H_N$. At every realization tick $L_N \le t \le B_N$,*

$$\mathbf{W}_N(t) = \mathcal{V}(t) - \mathcal{V}(L_N) = \boldsymbol{\nu}(t/L_N). \tag{7}$$

In particular, $\mathbf{W}_N(B_N) = \boldsymbol{\nu}(H_N)$.

Proof. Lemma 2.1 gives $B_N \ge L_N$, so the target is reached by $B_N - L_N$ successor steps. The vector increments telescope from zero:

$$\mathbf{W}_N(t) = \sum_{u=L_N+1}^{t} \Big(\mathcal{V}(u) - \mathcal{V}(u-1)\Big) = \mathcal{V}(t) - \mathcal{V}(L_N).$$

At the target, the represented quotient is $B_N/L_N = H_N$. For $N = 1$, the target equals the synchronization time, and the initial zero vector already represents $H_1 = 1$. □

For interpreting the trajectory, it is useful to retain the optional *remainder* $R_N(t) := B_N - t$. It is a derived integer balance, with

$$t + R_N(t) = B_N, \qquad R_N(L_N) = B_N - L_N. \tag{8}$$

It decreases by one at each realization tick, and certifies the unfinished contribution:

$$\frac{t}{L_N} + \frac{R_N(t)}{L_N} = \frac{B_N}{L_N} = H_N. \tag{9}$$

Each tick adds $1/L_N$ to the represented value. In multiplicative coordinates the same step multiplies by $(t+1)/t$, which explains the valuation difference in (6). There is no assertion that valuation vectors add when rational numbers are added.

The separation of counting and factorized realization is essential. The running count already has $C_3(3) = 5$ at tick 3; requesting its factorization immediately would require a prime not yet generated. An integer count can therefore be known before its valuation row is available. Continuing to the target makes the numerator the current native integer itself, with all of its prime coordinates available.

2.3 What is generated, and when

Here *from scratch* means that at time 1 there are no prime clocks and the only native row is $\mathcal{V}(1) = \mathbf{0}$. The clock advances and resets integer counters and reads exponents from rows stored at earlier ticks. On valuation rows it needs only componentwise addition, subtraction and comparison. No primes, factorizations, least common multiples or harmonic values are supplied. Memory and work per tick may grow.

A clock of period p stores a hand position $r_p \in \{0, \ldots, p-1\}$ and a counter c_p for completed cycles. At each tick, the hand advances by one. When it would reach p, it resets to zero and c_p increases by one; this reset is a ring, also called a carry. If none of the existing clocks rings at a new tick $t \geq 2$, install a clock of period $p = t$ in state $(r_p, c_p) = (0, 1)$, recording its first multiple. These installation times are precisely the primes. The counters satisfy $t = pc_p(t) + r_p(t)$, and the exponent of p is read out as

$$v_p(t) = \begin{cases} 0, & r_p(t) \neq 0, \\ 1 + v_p(c_p(t)), & r_p(t) = 0. \end{cases} \tag{10}$$

In the second line, $v_p(c_p(t))$ is read from the previously stored row at tick $c_p(t) < t$. An entry whose historical tick is below p is zero, since no positive integer below p is divisible by p. Appendix A proves that this produces $v_p(t) = \nu_p(t)$.

Divisibility is then a comparison of available rows:

$$m \mid t \quad \Longleftrightarrow \quad \mathcal{V}(m) \leq \mathcal{V}(t) \quad \text{componentwise.} \tag{11}$$

Only $m \leq t$ need be tested. Comparing entries gives the running maximum at tick N,

$$E_N = \max_{1 \leq m \leq N} \mathcal{V}(m). \tag{12}$$

The first row $\mathcal{V}(t)$ dominating E_N is exactly the row at L_N: it is the first integer divisible by every $m \leq N$. Since $L_N \geq N$, starting this test at tick N misses no synchronization.

Theorem 2.3 (Factorized harmonic numbers from scratch)**.** *There is a single deterministic forward process, starting with no supplied primes, that creates observer N at tick N and emits $\boldsymbol{\nu}(H_N)$ at tick $T_N = L_N H_N$ for every $N \geq 1$. Collection ends by a detected synchronization at L_N, and realization ends when the clock reaches the frozen full count B_N. Every datum used has already been generated, either at an earlier tick or earlier in the current tick. Each finite tick requires only finite state and finite work. No earlier tick is recomputed to obtain additional precision.*

The only initialization issue is that observer N is born after ticks $1, \ldots, N-1$ have passed. Maintain the shared count $S(t) = \sum_{s=1}^{t} \tau(s)$, where $\tau(s)$ counts all positive divisors. For $s \le N$, every divisor of s is at most N, so

$$S(N) = \sum_{s=1}^{N} d_N(s).$$

Thus $S(N)$ supplies exactly the elapsed collection count, including the birth tick. Later ticks add their own $d_N(t)$. This is the reason all observers can share one run. Appendix B gives the precise update order and completes the proof of the theorem, including $N = 1$.

2.4 Two examples

For $N = 3$, synchronization occurs at $L_3 = 6$, and the full event count is $B_3 = 6+3+2 = 11$. Thus the target is tick 11, and the initial remainder is $R_3(6) = 11 - 6 = 5$. The table and Figure 1 give every realization tick; $\mathbf{e}_p$ denotes the unit vector in prime coordinate p.

t	$R_3(t)$	$t/6$	$\mathbf{W}_3(t)$
6	5	1	0
7	4	7/6	$\mathbf{e}_7 - \mathbf{e}_2 - \mathbf{e}_3$
8	3	4/3	$2\mathbf{e}_2 - \mathbf{e}_3$
9	2	3/2	$\mathbf{e}_3 - \mathbf{e}_2$
10	1	5/3	$\mathbf{e}_5 - \mathbf{e}_3$
11	0	11/6	$\mathbf{e}_{11} - \mathbf{e}_2 - \mathbf{e}_3$

The quotient $t/6$ increases at every tick, but its prime exponents need not: the factor 7 appears at tick 7 and disappears at tick 8. At tick 11 the generator discovers the prime 11 before the observer uses its row to emit $H_3 = 11/6$.

In both figures, the side panel counts unmatched events: $C_N(t) - t$ during collection and $R_N(t) = B_N - t$ during realization. Subtracting t pairs each tick with its divisor-1 event; the target remains B_N.

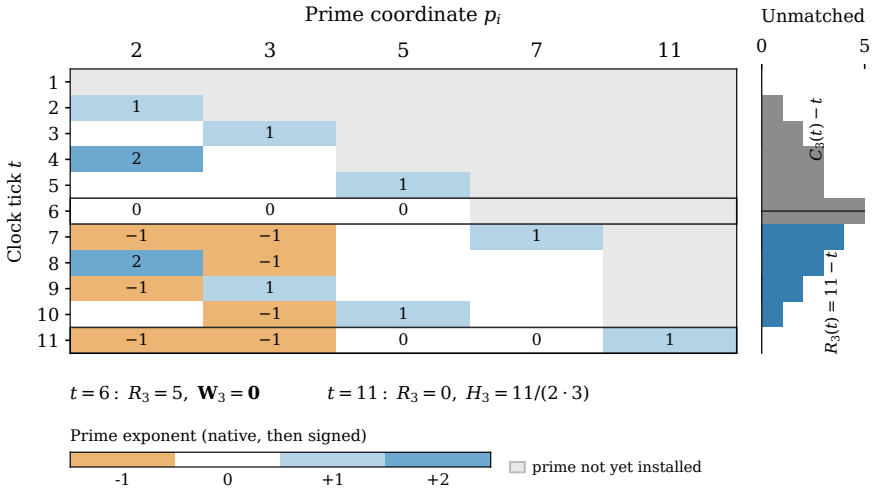


Figure 1: The H_3 process. Before synchronization, pixels show $\nu_p(t)$; from $t = 6$ onward they show $W_{3,p}(t) = \nu_p(t) - \nu_p(6)$. Blue denotes positive exponents, orange negative exponents, white zero, and gray primes not yet installed. At $t = 6$ the signed row is zero and $R_3 = 5$; at $t = 11$ the remainder is zero and the row gives $H_3 = 11/(2 \cdot 3)$.

For $N = 6$, the event contributions are

$$L_6 = 60, \qquad B_6 = 60 + 30 + 20 + 15 + 12 + 10 = 147.$$

The target is therefore 147, with initial remainder $147 - 60 = 87$. At the target tick the difference of the native and stored denominator rows is

$$\mathcal{V}(147) - \mathcal{V}(60) = (\mathbf{e}_3 + 2\mathbf{e}_7) - (2\mathbf{e}_2 + \mathbf{e}_3 + \mathbf{e}_5) = 2\mathbf{e}_7 - 2\mathbf{e}_2 - \mathbf{e}_5.$$

The factor 3 cancels, giving $H_6 = 49/20$. No separate factorization of a terminal harmonic numerator is requested: its unreduced numerator has become the current native integer. Figure 2 displays this evolution one tick at a time.

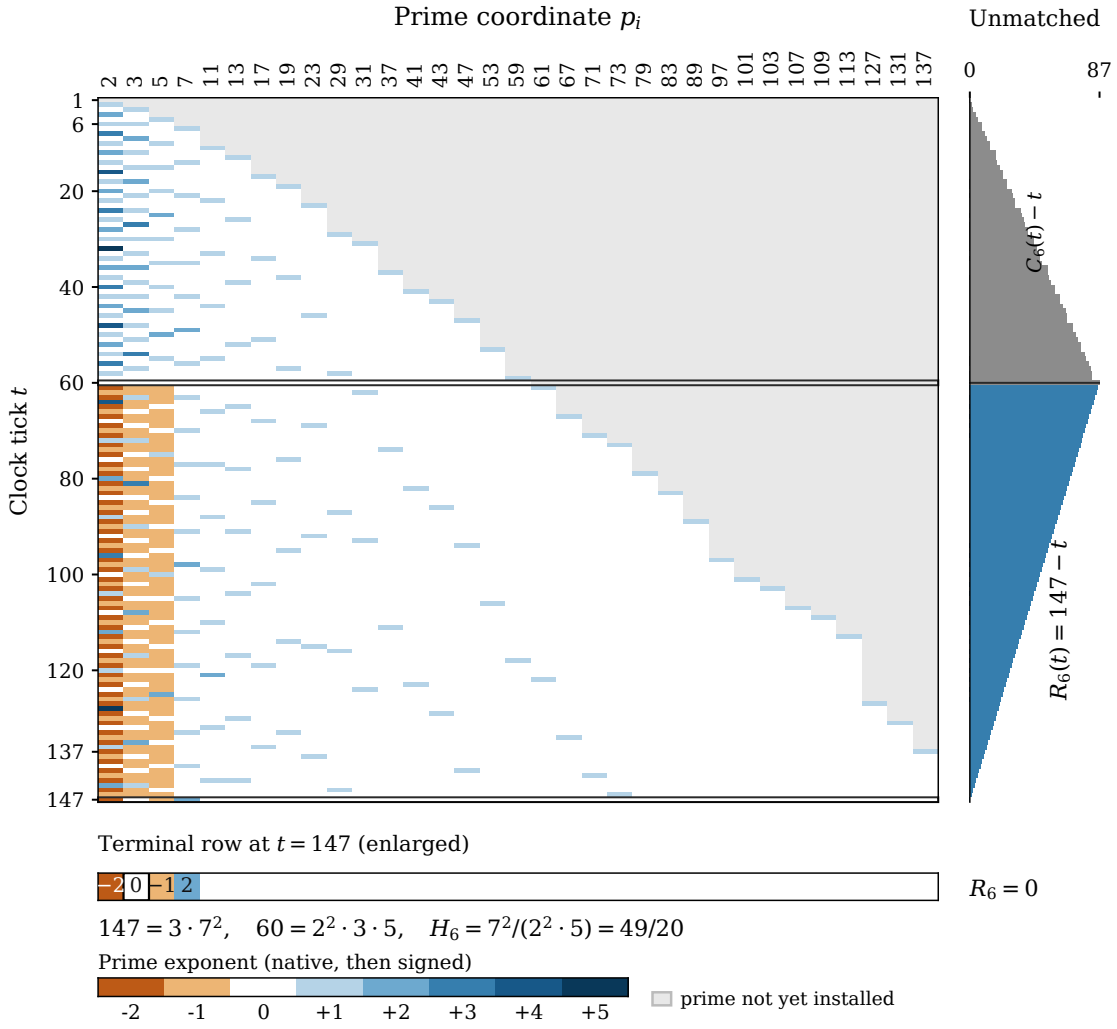


Figure 2: The H_6 process on one clock, with rows indexed by t and columns by the generated primes $p_i \le 137$. Before $t = 60$, pixels show the native exponents $\nu_p(t)$; from $t = 60$ onward they show the signed exponents $W_{6,p}(t) = \nu_p(t) - \nu_p(60)$. Blue denotes positive exponents, orange negative exponents, and white zero; gray marks only primes not yet installed. At the first synchronization $t = 60$, every $m \le 6$ divides the clock value, the full count is $B_6 = 147$, and $\mathbf{W}_6(60) = \mathbf{0}$. The unmatched count is $R_6(60) = 87$; it decreases once per tick and reaches zero at the target $t = B_6 = 147$. The enlarged terminal row makes cancellation visible: the common factor 3 of $147 = 3 \cdot 7^2$ and $60 = 2^2 \cdot 3 \cdot 5$ gives a zero $p = 3$ entry, leaving $H_6 = 49/20$. The prime 139 lies outside the displayed range.

3 Arithmetic structure and logarithmic blocks

3.1 Cancellation and a persistent power of two

If $H_N = a_N/b_N$ in lowest terms, the construction gives

$$a_N = \frac{T_N}{\gcd(T_N, L_N)}, \qquad b_N = \frac{L_N}{\gcd(T_N, L_N)}. \tag{13}$$

Thus denominator factors can only come from L_N, whereas numerator factors may first appear after synchronization. The example H_3 shows that a numerator prime can even be discovered at the release tick. One denominator component has a particularly simple exact behavior.

Proposition 3.1 (The binary denominator)**.** *If $2^h \le N < 2^{h+1}$, then*

$$\nu_2(H_N) = -h. \tag{14}$$

Consequently $B_N = T_N$ is odd, and the full factor 2^h of L_N survives in the reduced denominator. In particular, H_N is not an integer for $N > 1$.

Proof. Among $1, \dots, N$, only 2^h has 2-adic valuation h, and $\nu_2(L_N) = h$. In the integer event count

$$B_N = \sum_{m=1}^{N} \frac{L_N}{m},$$

the contribution with $m = 2^h$ is therefore odd, and every other contribution is even. Thus B_N is odd. The quotient $H_N = B_N/L_N$ has 2-adic valuation $-h$, and $T_N = B_N$ has the same odd parity. □

This is the unique-largest-power argument used in the proof of harmonic nonintegrality; Conrad's exposition states the valuation formula explicitly [3, pp. 1–2]. Here the argument identifies the parity of the full event count and hence of every release time. The release times increase strictly with N, since $L_{N+1} \ge L_N$ and $H_{N+1} > H_N$. They can differ even when the synchronization time is unchanged: $L_5 = L_6 = 60$, but $T_5 = 137 < T_6 = 147$.

3.2 A logarithm is a limiting band average

For integers $q \ge 2$ and $N \ge 1$, consider the finite harmonic block

$$D_q(N) = H_{qN} - H_N = \sum_{m=N+1}^{qN} \frac{1}{m}. \tag{15}$$

It counts the average contribution of periods in the band $(N, qN]$. Indeed, putting $L = L_{qN}$ and counting as in Lemma 2.1,

$$D_q(N) = \frac{1}{L} \sum_{t=1}^{L} \#\{m : N < m \le qN,\ m \mid t\}. \tag{16}$$

All terms on the right are finite integer counts. An equivalent expression is the Riemann sum

$$D_q(N) = \frac{1}{N} \sum_{j=1}^{(q-1)N} \frac{1}{1 + j/N}.$$

Scaling both ends of the band changes its mesh but keeps the ratio q.

Proposition 3.2 (Logarithmic limit and error)**.** *For fixed integer $q \geq 2$, $D_q(N)$ increases strictly to $\log q$, and*

$$0 < \log q - D_q(N) < \frac{1}{N} - \frac{1}{qN}. \tag{17}$$

Proof. The difference of consecutive blocks is

$$D_q(N+1) - D_q(N) = \sum_{j=1}^{q} \frac{1}{qN+j} - \frac{1}{N+1} > 0,$$

because each term in the sum is at least $1/[q(N+1)]$ and some are strictly larger. For every integer $m \geq 2$, monotonicity of $1/x$ gives

$$0 < \int_{m-1}^{m} \frac{dx}{x} - \frac{1}{m} < \frac{1}{m-1} - \frac{1}{m}.$$

Summing from $m = N+1$ to qN and substituting $x = Nu$ in the integral proves (17), with $\log q = \int_1^q du/u$. □

The construction of each rational block uses no logarithm. The integral identifies its real limit and supplies the error bound. Equally, one may first define that limit arithmetically: the preceding difference proves monotonicity, and $D_q(N) < q - 1$ gives boundedness. The integral calculation then shows that this limit has the usual logarithmic normalization.

3.3 Finite concatenation and the logarithm laws

The multiplication rule is already visible as an exact identity between finite blocks:

$$\boxed{D_{ab}(N) = D_a(N) + D_b(aN)} \qquad (a, b \geq 2). \tag{18}$$

The right side divides the interval of denominators at aN:

$$(H_{aN} - H_N) + (H_{abN} - H_{aN}) = H_{abN} - H_N.$$

Endpoint cancellation holds for any sequence F: the differences $C_q(N) = F(qN) - F(N)$ satisfy $C_{ab}(N) = C_a(N) + C_b(aN)$. For harmonic blocks, the essential further fact is their convergence to a nonzero limit depending only on the endpoint ratio. Together, these properties explain how the finite arithmetic of adjacent blocks gives rise to the logarithmic law.

Letting N tend to infinity in (18) gives $\log(ab) = \log a + \log b$. The change from N to aN is essential at the finite stage. For example,

$$D_6(1) = \frac{29}{20} = \underbrace{\frac{1}{2}}_{D_2(1)} + \underbrace{\left(\frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6}\right)}_{D_3(2)}.$$

At a fixed starting scale, $D_4(1) = 13/12$, whereas $2D_2(1) = 1$. More generally, Proposition 3.2 gives

$$D_{ab}(N) - D_a(N) - D_b(N) = D_b(aN) - D_b(N) > 0.$$

This difference tends to zero as $N \to \infty$. Thus the finite identity records the change of scale explicitly, and its limiting form depends only on the endpoint ratios.

To include quotients, put $\Lambda_{a,b}(N) = H_{aN} - H_{bN}$ for positive integers a, b. Proposition 3.2, or the same integral estimate, gives

$$\lim_{N\to\infty} \Lambda_{a,b}(N) = \log(a/b), \qquad |\Lambda_{a,b}(N) - \log(a/b)| < \frac{|a^{-1} - b^{-1}|}{N} \quad (a \neq b). \tag{19}$$

The finite identity

$$\Lambda_{ac,bd}(N) = \Lambda_{a,b}(cN) + \Lambda_{c,d}(bN) \tag{20}$$

follows by cancelling H_{bcN}. Moreover, $\Lambda_{b,a}(N) = -\Lambda_{a,b}(N)$, and $\Lambda_{ua,ub}(N) = \Lambda_{a,b}(uN)$. The latter equality, together with a common representation of a rational number, shows that the limit depends only on a/b. Thus multiplication, inversion, and iteration give, for positive rationals x, y and integers r,

$$\log(xy) = \log x + \log y, \qquad \log(x/y) = \log x - \log y, \qquad \log(x^r) = r \log x.$$

These laws extend to positive real arguments by continuity of the integral defining log. Their finite antecedents are the block cancellations above, which involve only rational arithmetic.

3.4 A denominator pattern in the dyadic blocks

The real limit does not remove the blocks' arithmetic structure. For every integer $k \geq 0$,

$$\nu_2\Big(D_2(2^k)\Big) = -(k+1). \tag{21}$$

Indeed, in the interval $2^k < m \leq 2^{k+1}$ the terminal denominator is the unique one divisible by 2^{k+1}. Multiplication by 2^{k+1} turns its reciprocal into 1, and every other term has even numerator and odd denominator. The parity argument of Proposition 3.1 applies unchanged. Thus the blocks approach $\log 2$ while their reduced denominators retain exactly the powers $2, 4, 8, 16, \ldots$ of two.

4 A rational descent to Euler's constant

Euler's constant measures the difference between a harmonic prefix and its logarithmic growth. We first record an elementary bound to make the approximation argument self-contained. Set $\alpha_N = H_N - \log N$. Since $H_N > \int_1^{N+1} dx/x$, the sequence is positive. Furthermore,

$$0 < \alpha_N - \alpha_{N+1} = \int_N^{N+1} \frac{dx}{x} - \frac{1}{N+1} < \frac{1}{N} - \frac{1}{N+1}. \tag{22}$$

It therefore has a limit, denoted γ_{EM}, and telescoping yields

$$0 < H_N - \log N - \gamma_{\mathrm{EM}} < \frac{1}{N}. \tag{23}$$

The bounds remain strict after summation: each summand is positive, and the first already leaves a positive gap below its telescoping upper bound.

At $N = 2^k$, the logarithmic term is $k \log 2$. Replacing $\log 2$ by its arithmetic block approximation gives the rational number

$$\boxed{G_k = H_{2^k} - kD_2(2^k) = (k+1)H_{2^k} - kH_{2^{k+1}}}, \qquad k \geq 0. \tag{24}$$

This formula identifies the approximation through two harmonic values. Below we construct it directly from signed divisor counts, including its reduced valuation row. No supplied value of $\log 2$ or γ_{EM} is needed.

Theorem 4.1 (Rational approximation from above)**.** *The sequence G_k decreases strictly to γ_{EM}, with*

$$0 < G_k - \gamma_{\mathrm{EM}} < \frac{k+2}{2^{k+1}}. \tag{25}$$

Its successive decrements have the exact positive expression

$$G_k - G_{k+1} = (k+1) \sum_{j=2^k+1}^{2^{k+1}} \frac{1}{2j(2j-1)}. \tag{26}$$

Proof. Put $N = 2^k$, $e_N = H_N - \log N - \gamma_{\mathrm{EM}}$, and $\delta_N = \log 2 - D_2(N)$. Equations (17) and (23) give $0 < e_N < 1/N$ and $0 < \delta_N < 1/(2N)$. Hence

$$G_k - \gamma_{\mathrm{EM}} = e_N + k\delta_N,$$

which proves (25) and convergence.

To obtain the decrement, write $D(N) = D_2(N)$. Since $H_{2N} = H_N + D(N)$, direct subtraction gives

$$G_k - G_{k+1} = (k+1)\Big(D(2N) - D(N)\Big).$$

Pair the terms in $D(2N)$ with denominators $2j-1, 2j$, where $N < j \le 2N$. Their excess over $1/j$ is

$$\frac{1}{2j-1} + \frac{1}{2j} - \frac{1}{j} = \frac{1}{2j(2j-1)}.$$

Summation proves (26) and strict decrease. □

4.1 Native factorization by signed divisor counts

The harmonic formula (24) identifies G_k, but adding its valuation rows would not perform the indicated rational subtraction. Instead, we give the two divisor bands their coefficients before counting. Put $N = 2^k$ and define the signed event count

$$\eta_k(t) = \sum_{m=1}^{N} \mathbf{1}_{\{m|t\}} - k \sum_{m=N+1}^{2N} \mathbf{1}_{\{m|t\}}. \tag{27}$$

Every divisor in the first band contributes $+1$; every divisor in the second contributes $-k$. These are updates of an integer counter, using the same comparisons of generated valuation rows as before. The signed counter need not move monotonically. At the common synchronization of all divisors up to $2N$, freeze its value

$$Z_k := \sum_{t=1}^{L_{2N}} \eta_k(t). \tag{28}$$

Proposition 4.2 (Native dyadic approximation)**.** *For every $k \ge 0$, with $N = 2^k$, the signed count satisfies*

$$\boxed{G_k = \frac{Z_k}{L_{2N}}, \qquad \boldsymbol{\nu}(G_k) = \mathcal{V}(Z_k) - \mathcal{V}(L_{2N}).} \tag{29}$$

Moreover, $1 \le Z_k \le L_{2N}$, with strict upper inequality for $k \ge 1$. The two native rows in (29) *are therefore available at the synchronization tick itself.*

Proof. Counting multiples over the common period, as in Lemma 2.1, gives the integer identity

$$Z_k = \sum_{m=1}^{N} \frac{L_{2N}}{m} - k \sum_{m=N+1}^{2N} \frac{L_{2N}}{m} = L_{2N}\Big(H_N - kD_2(N)\Big) = L_{2N}G_k.$$

Theorem 4.1, $G_0 = 1$, and $\gamma_{\mathrm{EM}} \geq 0$ imply $0 < G_k \leq 1$, strictly below 1 for $k \geq 1$. Since Z_k is an integer, this proves the stated bounds. The native generator has published every row with index at most L_{2N} by synchronization. Subtracting the denominator row from the row at the generated address Z_k gives exactly the reduced valuation vector of their quotient. □

This construction also fits one uninterrupted forward run. Generate the dyadic thresholds by repeated doubling, beginning with $N = 1$, and let k count the doublings. Create observer k at tick $N = 2^k$ with signed counter $S(N)$. This is the correct elapsed count: for $s \leq N$, the second band contributes nothing and every divisor of s lies in the first band, so $\eta_k(s) = \tau(s)$. At each later tick add $\eta_k(t)$, testing only divisors at most t. At tick $2N$, store the running maximum E_{2N}; its first domination by the current row detects L_{2N}. Freeze Z_k there and read the already stored row $\mathcal{V}(Z_k)$. No earlier tick is replayed, and neither a harmonic output nor a new factorization is needed for this readout.

For a worked example, take $k = 1$, so $N = 2$. Divisors 1 and 2 contribute $+1$ per event, while divisors 3 and 4 contribute -1:

$$\eta_1(t) = 1 + \mathbf{1}_{\{2|t\}} - \mathbf{1}_{\{3|t\}} - \mathbf{1}_{\{4|t\}}.$$

The observer starts at tick 2 with $S(2) = 1 + 2 = 3$ and adds the signed events at each subsequent tick. At tick 4, it stores $E_4 = 2\mathbf{e}_2 + \mathbf{e}_3$. The first current row dominating this maximum occurs at tick 12, detecting synchronization. Over ticks $1, \ldots, 12$, the four divisors have contributed

$$Z_1 = \underbrace{12}_{m=1} + \underbrace{6}_{m=2} - \underbrace{4}_{m=3} - \underbrace{3}_{m=4} = 11.$$

These are the accumulated events; the value 12 was not supplied as a stopping time. At tick 11, the native generator has already installed prime 11 and stored $\mathcal{V}(11) = \mathbf{e}_{11}$. At tick 12, it first publishes $\mathcal{V}(12) = 2\mathbf{e}_2 + \mathbf{e}_3$. The observer then freezes its count at $Z_1 = 11$ and uses that count as the address of the stored numerator row:

$$\boldsymbol{\nu}(G_1) = \mathcal{V}(11) - \mathcal{V}(12) = \mathbf{e}_{11} - 2\mathbf{e}_2 - \mathbf{e}_3, \qquad G_1 = \frac{11}{2^2 \cdot 3} = \frac{11}{12}.$$

Thus the factorization is read entirely from rows already generated when synchronization is detected.

Corollary 4.3 (The binary valuation of the dyadic approximations)**.** *For every $k \geq 0$, the signed count and its period average satisfy*

$$\nu_2(Z_k) = \nu_2(k+2), \qquad \nu_2(G_k) = \nu_2(k+2) - (k+1).$$

Proof. Put $N = 2^k$ and $L = L_{2N}$. Over one period, let E and O count the events from even and odd divisors in $[1, 2N]$, respectively. Each $j \leq N$ contributes twice as many events as $2j$, since $L/j = 2L/(2j)$. Thus the lower band $[1, N]$ contributes $2E$, leaving $O - E$ events in the upper band. With the signed weights this gives

$$Z_k = 2E - k(O - E) = (k+2)E - kO.$$

Since $\nu_2(L) = k+1$, every odd divisor contributes a multiple of 2^{k+1} to O. Among the even divisors, only $2N$ contributes an odd number, so E is odd. For $k \geq 1$ we have $k+2 < 2^{k+1}$; hence subtracting kO leaves the valuation $\nu_2(k+2)$ of $(k+2)E$ unchanged. For $k = 0$, the divisor 1 contributes twice and the upper band has weight zero, so $Z_0 = 2$. In both cases, $G_k = Z_k/L$ gives the second formula. □

For $k = 2$, the corresponding integers are $L_8 = 840$ and $Z_2 = 684$, giving $G_2 = 684/840 = 57/70$. Exactly two factors of 2 cancel, as predicted by $\nu_2(k+2) = \nu_2(4) = 2$. The native row at 684 already contains the prime 19 needed in this numerator. Harmonic averages use a target $B_N \geq L_N$ reached by continuing the clock; the dyadic approximations use a count $Z_k \leq L_{2N}$ whose row is available when collection ends. In both cases the factorization is obtained from generated integer rows.

The first values illustrate both approaches to their limits:

k	$D_2(2^k)$	Decimal value	G_k	Decimal value
0	1/2	0.500000	1	1.000000
1	7/12	0.583333	11/12	0.916667
2	533/840	0.634524	57/70	0.814286
3	95549/144144	0.662872	175193/240240	0.729242

The blocks increase to $\log 2$, while G_k decreases to γ_{EM}. The bound (25) is a direct certificate of convergence obtained from the same comparison of harmonic blocks.

4.2 Exact relation to a binary-length series

Let $\ell_2(m)$ denote the number of digits in the binary representation of a positive integer m. Equivalently, $\ell_2(m) = k+1$ for $2^k \leq m < 2^{k+1}$. This definition uses integer comparisons with powers of two; no logarithm is needed to compute it.

Corollary 4.4 (The finite partial sums)**.** *For every integer $K \geq 0$,*

$$G_K = 1 - \sum_{m=1}^{2^K-1} \frac{\ell_2(m)}{(2m+1)(2m+2)}. \tag{30}$$

Consequently,

$$\gamma_{\mathrm{EM}} = 1 - \sum_{m=1}^{\infty} \frac{\ell_2(m)}{(2m+1)(2m+2)}. \tag{31}$$

Proof. Telescope (26) from $k = 0$ to $K-1$, using $G_0 = 1$. In its kth block, set $m = j-1$. Then $2^k \leq m < 2^{k+1}$ and $k+1 = \ell_2(m)$, proving (30). Theorem 4.1 gives the limit. Since all summands are positive, convergence of these increasing partial sums also gives convergence of the full series. □

Formula (31) is exactly equation (6), p. 2, of Hessami Pilehrood and Hessami Pilehrood [7]. Their numerator $N_{1,2}(m) + N_{0,2}(m)$ counts binary ones and zeros and is therefore $\ell_2(m)$. Equation (30) makes the relation precise: our G_K is the partial sum stopped immediately before 2^K. The derivation here passes through harmonic blocks and their positive decrements. It does not propose a new infinite series for Euler's constant.

Proposition 4.2 adds a native realization of these finite partial sums: the signed incidence count Z_k selects an already generated integer row. Prime cancellation is then precisely the componentwise difference $\mathcal{V}(Z_k) - \mathcal{V}(L_{2^{k+1}})$. Relating the resulting exponent patterns to the approximation error remains a further question.

5 A research outlook through Lagarias's inequality

The divisibility relation supports both the harmonic event average and the sum-of-divisors function $\sigma(n) = \sum_{d|n} d$. Their connection is especially suggestive in view of Lagarias's theorem [6, Theorem 1.1]: the Riemann hypothesis is equivalent to

$$\sigma(n) \le H_n + \exp(H_n) \log H_n \qquad (n \ge 1), \tag{32}$$

with equality only at $n = 1$. Lagarias obtains this criterion from Robin's results; see [6, §1, equation (1.2), and §3]. Here it serves as a direction for structural investigation.

To make the common arithmetic input explicit, define the truncated reciprocal divisor sum

$$U_N(t) = \sum_{\substack{1 \le m \le N \\ m \mid t}} \frac{1}{m}.$$

This is a different reading of the same incidence data used by the event counters: each successful divisibility test is weighted by $1/m$.

Proposition 5.1 (A divisor profile and its completion)**.** *For every $N \ge 1$,*

$$U_N(N) = \frac{\sigma(N)}{N}, \qquad U_N(L_N) = H_N, \tag{33}$$

$$H_N - \frac{\sigma(N)}{N} = \sum_{\substack{1 \le m \le N \\ m \nmid N}} \frac{1}{m}. \tag{34}$$

Proof. The map $d \mapsto N/d$ permutes the divisors of N, so $\sigma(N)/N = \sum_{d|N} 1/d = U_N(N)$. Every $m \le N$ divides L_N, giving $U_N(L_N) = H_N$. Subtracting the first sum from the full harmonic prefix gives (34). □

For example, $U_6(6) = 1 + 1/2 + 1/3 + 1/6 = 2$, whereas $U_6(60) = H_6 = 49/20$. The difference $9/20 = 1/4 + 1/5$ is exactly the reciprocal mass missing from the divisor set of 6. The completion is a statement about the two divisor profiles, not a claim that $U_N(t)$ increases at every intervening tick. In the harmonic construction, the same H_N is obtained as the average of unweighted divisor events and is subsequently released in factorized form. These descriptions connect a divisor profile, its completion at synchronization, and its realization along the successor clock.

At native tick n, the generated row $\mathcal{V}(n) = (a_p)_p$ determines

$$U_n(n) = \frac{\sigma(n)}{n} = \prod_{p : a_p > 0} (1 + p^{-1} + \cdots + p^{-a_p}), \tag{35}$$

by expanding the product over divisors; this is the prime-profile formula in [6, §2, equation (2.1)]. At release tick T_n, the harmonic observer supplies $H_n = T_n/L_n$ in reduced

prime-exponent form. Thus both rational inputs to (32) arise from the same generated arithmetic data, at their respective times.

The relation to logarithms and Euler's constant is also explicit. Put $\varepsilon_n = H_n - \log n - \gamma_{\mathrm{EM}}$; equation (23) gives $0 < \varepsilon_n < 1/n$. If $\Delta_n = H_n + e^{H_n} \log H_n - \sigma(n)$, then

$$\frac{\Delta_n}{n} = \frac{H_n}{n} + e^{\gamma_{\mathrm{EM}}+\varepsilon_n} \log H_n - U_n(n). \tag{36}$$

This identity isolates the generated prime profile in the last term. It also explains the connection with Robin's comparison of $\sigma(n)/n$ and $e^{\gamma_{\mathrm{EM}}} \log\log n$, developed quantitatively in [6, §3, Lemmas 3.1–3.2]: $H_n \sim \log n$ and $\varepsilon_n \to 0$ identify that scale. The completion identity gives $\sigma(n)/n \le H_n$, an upper bound of order $\log n$. Controlling (36) at the finer $\log\log n$ scale requires additional estimates on the exponent profiles.

Three questions suggest a concrete continuation of this work.

1. Which features of the generated exponent profile make $U_n(n)$ unusually large? Lagarias's discussion of colossally abundant numbers [6, §2] indicates relevant profiles against which to compare the prime-power thresholds $p^a \le M$ that determine the synchronization integers L_M.

2. Can the missing reciprocal mass in (34) be bounded in terms of those profiles in a way that controls (36)? An average event count alone does not provide the needed pointwise estimate.

3. Can the explicit harmonic-block and Euler-constant error bounds be combined with these profile estimates to obtain certified comparisons on selected families of integers, and identify which estimates would need to hold uniformly?

The common recursion provides an explicit setting for studying the divisor profiles appearing in Lagarias's criterion. Whether this viewpoint leads to additional structural estimates remains an open direction for further research.

6 Relation to earlier work and computational scope

6.1 The native arithmetic and the harmonic observer

The prime-clock generator used here is the one developed in [4, §2.1]; its valuation readout is equation (5) there. That work also describes divisibility as componentwise order and rationals as signed valuation vectors [4, §3.3.3 and §4.1]. We reproduce the required proofs so that the present paper can be read independently. The harmonic-specific step is to count all divisibility events, use the full count as the successor target, and attach the transported valuation vector to that path. The shared initialization in Appendix B then supplies all observers without rerunning the native generator.

The arithmetic consequences have been separated from their interpretation. The binary valuation formula is explicitly documented in [3], and the Euler-constant series is explicitly present in [7]. The contribution is the integration of event collection, synchronization, and valuation transport into one causal process, together with derivations that make the arithmetic structure of its outputs visible. For the dyadic Euler approximations, signed event counts select native rows that are already available at synchronization. In particular,

the logarithmic development exhibits exact finite concatenation and the emergence of a scale-independent limit. The endpoint-cancellation identity itself is general, as explained in Section 3; its role here is to expose this passage from finite arithmetic to the logarithm laws.

6.2 Boyd's second method and precision-driven replay

For a supplied odd prime p, Boyd studies the exceptional set

$$J_p = \{n \geq 1 : \nu_p(H_n) > 0\}.$$

His second computational method advances through a tree of candidate indices, retaining harmonic values at finite p-adic precision [1, §5.2]. Separating denominators divisible by p gives the exact identity

$$H_{pn+k} = \frac{H_n}{p} + F_p(n) + \sum_{j=1}^{k} \frac{1}{pn+j}, \quad 0 \leq k < p, \tag{37}$$

$$F_p(n) = \sum_{\substack{1 \leq j \leq pn \\ p \nmid j}} \frac{1}{j}.$$

Boyd's Theorem 5.2 provides a p-adically convergent series for $F_p(n)$. The division by p in (37) explains the loss of absolute precision in a tree step.

To make that statement precise, write $\widetilde{x} = x + O(p^r)$ when $\nu_p(\widetilde{x} - x) \geq r$, ignoring the sign of a nonzero rational and setting $\nu_p(0) = +\infty$. If H_n is known to precision r, then H_n/p is known to precision $r-1$. If the correction terms are also known to precision at least $r-1$, the resulting child is known to precision at least $r-1$. This bound cannot generally be improved: an admissible input error p^r becomes exactly p^{r-1} even when the correction terms are exact. For instance, $H_2 = 3/2$ and $F_3(2) = 39/20$; replacing H_2 by $H_2 + 27$ in (37) replaces $H_6 = 49/20$ by $H_6 + 9$.

If the last generation available at the chosen precision is nonempty, Boyd prescribes a restart with a larger truncation parameter [1, §5.2, p. 295]. Carofiglio, Cherubini, and Gambini retain this refinement and reduce the repeated work to ancestors of nodes that survive in the last computed block [2, §2, pp. 21–22]. Here *precision-driven replay* means recomputing previously visited states with more p-adic digits. It need not revisit every earlier node. The comparison concerns Boyd's second method; his direct sequential method in §5.1 has no corresponding one-digit loss at each step [1, pp. 293–294].

Our states are exact counters and exact integer exponents. A historical native row never needs additional digits, and a newly discovered prime has zero exponent at all earlier positive ticks. Historical lookup in (10) therefore reads an unchanged value; it does not replay the computation that produced it. In this sense the harmonic construction has no precision-refinement loop. The lookup of $\mathcal{V}(Z_k)$ for the dyadic approximations likewise reads a stored row without recomputing it.

The comparison concerns two different aims. Boyd's tree seeks J_p for one selected prime and can skip vast ranges of harmonic indices. Our purpose is to make arithmetic structure visible. The present process emits the full factorization of H_N at global time $T_N = L_N H_N$; the intervening ticks display how divisor events become increments of a rational value and how its prime exponents change. Retaining these intermediate states is part of the explanation. The distinction between the harmonic index N and the unfolding time t is therefore central to the construction.

6.3 Structural insight and further questions

Computational efficiency is not the aim of this construction. Its purpose is to understand H_N, logarithms, and their rules from an arithmetic viewpoint in which integers are compounds of primes and divisibility is expressed by comparison of their prime exponents. The individual successor steps from L_N to the full count B_N make the passage from divisor counts to rational values explicit. Although the target is known when collection ends, the trajectory shows how that value is realized with its prime structure available at every step. The same attention to finite arithmetic makes logarithmic rules visible as identities between harmonic blocks before any limit is taken.

Two questions remain natural within this viewpoint. First, which further patterns in harmonic numerators and denominators can be explained through the relation between synchronization and release? Second, Corollary 4.3 determines the coordinate at 2 in $\mathcal{V}(Z_k) - \mathcal{V}(L_{2^{k+1}})$; what patterns remain at odd primes? For the latter, a next step is to relate the resulting denominators to the approximation error in (25). Conclusions about the irrationality of γ_{EM} would require additional arithmetic estimates.

Acknowledgments. Reading Julian Havil's *Gamma: Exploring Euler's Constant* [5] ignited my interest in the harmonic numbers. I thank Mutlu Özdemir for discussing the recursive framework with me. I am especially grateful to my beloved wife and children for bearing with me while I played with harmonic numbers on weekends.

A Native generation of primes and valuation rows

This appendix specifies the generator used in Theorem 2.3 and verifies every history access. It is the two-hand prime-clock construction of [4], written in the notation of the present paper.

A.1 Carry, installation, and historical readout

At time 1, there are no installed periods and $\mathcal{V}(1) = \mathbf{0}$. To move from $t-1$ to $t \geq 2$, first advance every previously installed clock by

$$(r_p, c_p) \longmapsto \begin{cases} (r_p + 1, c_p), & r_p < p - 1, \\ (0, c_p + 1), & r_p = p - 1. \end{cases} \tag{38}$$

The second branch is called a ring. If none of the old clocks rings, install period $p = t$ in state $(0, 1)$. Then evaluate (10) for every installed period and store the resulting row. At a historical address $1 \leq j < p$, the value $v_p(j)$ is defined to be zero; at an address $j \geq p$, use the value stored when tick j was processed.

Proposition A.1 (Correct native output)**.** *The installed periods are exactly the primes up to the current time. For every installed prime p, the readout satisfies $v_p(t) = \nu_p(t)$. All history addresses are strictly earlier than the current tick.*

Proof. A clock installed at p satisfies

$$t = pc_p(t) + r_p(t), \qquad 0 \leq r_p(t) < p \tag{39}$$

at its installation time. Rule (38) preserves this identity at each subsequent tick: either the remainder increases by one, or it passes from $p-1$ to zero while the cycle count increases by one. Hence an installed period rings precisely at its positive multiples.

Inductively suppose the periods installed before t are exactly the primes below t. If t is prime, none divides it, and a clock is installed. If t is composite, it has a prime divisor $p < t$, whose clock rings, preventing installation. To see the existence needed here without a primality oracle, the least divisor of t greater than 1 is prime: a proper factor of that divisor would contradict minimality. For composite t, this least divisor is smaller than t. The induction starts with the empty array before tick 2.

If $r_p(t) \neq 0$, equation (39) gives $p \nmid t$, so the zero readout is correct. If $r_p(t) = 0$, it gives $t = pc_p(t)$, and therefore

$$\nu_p(t) = 1 + \nu_p(c_p(t)).$$

Here $1 \leq c_p(t) < t$ because $p \geq 2$. Induction on t makes the historical value correct. If its address is below p, its valuation is zero; otherwise that value was stored earlier. This includes the initial ring at $t = p$, whose address is 1 and whose readout is 1. □

The quotient and remainder in (39) describe generated counter values. No division or remainder operation is needed to update the hands. When a new prime is installed, its coordinate is zero in every earlier native row. Such coordinates may be implicit; adding them requires no recomputation of historical values.

A.2 Divisibility and synchronization

Unique prime factorization gives (11): every prime exponent of m must be no larger than the corresponding exponent of t. At tick t, every row required to compute $d_N(t)$ is already available, because only $1 \leq m \leq \min(N, t)$ can contribute. A finite loop of these comparisons therefore produces the events of (1).

The running componentwise maximum of the native rows can be updated from the current row alone. Its value at N is E_N in (12). For every positive integer t,

$$\mathcal{V}(t) \geq E_N \quad \Longleftrightarrow \quad m \mid t \text{ for all } 1 \leq m \leq N \quad \Longleftrightarrow \quad L_N \mid t.$$

Thus its first success is at L_N. The test does not evaluate the product of the prime powers in E_N, and it does not require a numerical value for L_N beforehand. At success, the current native row itself supplies the denominator row $\mathcal{V}(L_N)$.

B A simultaneous schedule and its proof

For completeness, we specify a single run producing all harmonic outputs. An observer has one of three phases: collection, realization, or finished. A phase change at a tick does not cause a second transport update at that same tick. If synchronization already reaches the target, the observer emits its initial zero row immediately.

B.1 The shared initialization

The number $\tau(t)$ is obtained by counting all successful tests $\mathcal{V}(m) \leq \mathcal{V}(t)$ for $1 \leq m \leq t$. Starting with $S(0) = 0$, maintain

$$S(t) = S(t-1) + \tau(t). \tag{40}$$

If observer N is created after this update at tick N, initialize its collection count with $C_N = S(N)$. For every $s \leq N$, the divisors of s lie in $[1, N]$, so

$$C_N = S(N) = \sum_{s=1}^{N} d_N(s). \tag{41}$$

This equality includes the birth tick. The new observer must therefore add $d_N(t)$ only at later ticks $t > N$. If $L_N = N$, the initial count already covers its whole collection period and freezes immediately.

B.2 Order of operations at tick t

Use the initial native row at $t = 1$ and thereafter the update in Appendix A. At every tick, perform the following finite operations in order.

1. Publish $\mathcal{V}(t)$, update $S(t)$ by (40), and update the running componentwise maximum of all native rows so far.

2. Process every observer that existed before this tick, according to its phase at the beginning of the tick. A collector adds $d_N(t)$ to C_N and tests $\mathcal{V}(t) \geq E_N$. On success, it stores $L_N = t$, freezes $B_N = C_N$, and sets $\mathbf{W}_N = \mathbf{0}$. If $t = B_N$, it emits this row and finishes; otherwise it enters realization. An observer already realizing adds $\mathcal{V}(t) - \mathcal{V}(t-1)$ to $\mathbf{W}_N$; if $t = B_N$, it emits $\mathbf{W}_N$ and finishes. Finished observers make no further updates.

3. Create observer $N = t$ with E_N equal to the current maximum and $C_N = S(t)$. Test synchronization immediately and, on success, freeze and test the target as in step 2. Do not add the birth-tick events again.

For $N = 1$, this same rule gives $L_1 = B_1 = T_1 = 1$ and emits $\mathbf{0}$, representing $H_1 = 1$. For every other observer, the first realization update is at $L_N + 1$, precisely as in (6).

B.3 Proof of Theorem 2.3

Proposition A.1 supplies all prime labels and native rows using only generated state. The prefix count and running maximum likewise use available rows. Equation (41) initializes each observer with exactly the events through its birth tick. By induction, while it remains a collector, its count after tick t is

$$C_N(t) = \sum_{s=1}^{t} d_N(s).$$

The first successful synchronization is L_N, by Appendix A. Thus its frozen target is exactly B_N, and its initial realization state agrees with (5).

Proposition 2.2 now applies. The clock reaches the target after $B_N - L_N$ realization ticks, at $T_N = B_N = L_N H_N$, and the emitted vector is $\boldsymbol{\nu}(H_N)$. Every native row required for transport has already been published at the first stage of that tick. Any prime in $\mathcal{V}(t) - \mathcal{V}(L_N)$ divides one of the integers t, L_N, both at most t; its coordinate has therefore been installed. The case $N = 1$ follows from the same target test.

At a finite time t, there are t observers, finitely many installed periods, and t native rows, each with finite support. All loops, counter updates, and historical reads at that time are finite. Each fixed observer has finite L_N and B_N, and hence terminates. The schedule never resets the native clock and never recomputes a historical row. This proves all assertions of Theorem 2.3. □

References

[1] David W. Boyd. A p-adic study of the partial sums of the harmonic series. *Experimental Mathematics* **3**(4), 287–302, 1994. doi:10.1080/10586458.1994.10504298.

[2] Leonardo Carofiglio, Giacomo Cherubini, and Alessandro Gambini. On Eswarathasan–Levine and Boyd's conjectures for harmonic numbers. *Bulletin of the Australian Mathematical Society* **113**, 15–25, 2026. doi:10.1017/S0004972725100154.

[3] Keith Conrad. *The p-adic Growth of Harmonic Sums*. Expository notes, undated; consulted 26 October 2026.

[4] Michael T. M. Emmerich. *The Prime Clockwork: A Dynamic Representation of Modular and Multiplicative Arithmetic*. arXiv:2609.03896v1, 2026.

[5] Julian Havil. *Gamma: Exploring Euler's Constant*. Princeton University Press, 2003.

[6] Jeffrey C. Lagarias. An elementary problem equivalent to the Riemann hypothesis. *The American Mathematical Monthly* **109**(6), 534–543, 2002. Published version; arXiv:math/0008177v2.

[7] Khodabakhsh Hessami Pilehrood and Tatiana Hessami Pilehrood. Vacca-type series for values of the generalized Euler constant function and its derivative. *Journal of Integer Sequences* **13**, Article 10.7.3, 2010.

[8] OEIS Foundation Inc. A138553, table of counts of divisors of an integer up to a given bound. *The On-Line Encyclopedia of Integer Sequences*.

[9] OEIS Foundation Inc. A025529, the sequence $H_n \operatorname{lcm}(1, \ldots, n)$. *The On-Line Encyclopedia of Integer Sequences*.

How to cite. Michael T. M. Emmerich (2026). *Factorized Harmonic Numbers and Logarithms by a Forward Arithmetic Recursion*. Preprint, version dated 8 November 2026.

`https://github.com/emmerichmtm/FactorizedHarmonicNumbers/tree/v2026-11-08`

© 2026 Michael T. M. Emmerich. CC BY-NC 4.0.