A Proof that Every Positive Integer Appears on the Diagonal in the Kimberling Shuffle

Anta Osotsi

October 21, 2025

Contents

1. Definitions and Notation

- $\mathbb{N}$ – the set of natural numbers $\{1, 2, 3, \ldots\}$.
- S_i – the i-th sequence generated by the Kimberling shuffle [2].
- $S_1 = (1, 2, 3, 4, 5, \ldots)$ – the initial sequence.
- $d_i = s_{i,i}$ – the *diagonal element* (or expelled number) of S_i. This sequence of expelled numbers is cataloged as sequence A007063 in the On-Line Encyclopedia of Integer Sequences [3].
- **Shuffle rule:** to form S_{i+1} from S_i, remove d_i and re-interleave the remaining elements alternately from right and left:

$$\text{right}_1,\ \text{left}_1,\ \text{right}_2,\ \text{left}_2, \ldots$$

- $\text{pos}(n, i)$ – the position of n in S_i, if not yet expelled.
- $U = \{n \in \mathbb{N} \mid n \neq d_i\ \forall i\}$ – the set of never-expelled numbers.
- $n_0 = \min(U)$ – the smallest never-expelled integer.
- I – an index such that all $k < n_0$ are expelled by step I.
- $a_i = \text{pos}(n_0, i)$ – the position of n_0 in S_i for $i > I$.
- $\delta_i = i - a_i$ – the signed distance of n_0 from the diagonal.
- S_i denotes the sequence remaining after the i-th shuffle (so S_0 is the initial sequence).
- For each $i \geq 0$, let a_i denote the position of the distinguished element n_0 in S_i (if n_0 is already expelled at step i then a_i is undefined). We adopt the convention that positions are counted from 1.
- Define the *diagonal distance*

$$\delta_i := i - a_i.$$

  Thus $\delta_i > 0$ means n_0 lies to the left of the diagonal at step i, $\delta_i < 0$ means it lies to the right, and $\delta_i = 0$ means it lies on the diagonal.

- Define the index

$$I := n_0 - 1,$$

i.e. I is the last index at which every $k < n_0$ has been expelled. In particular, for every $i \geq I$ the element n_0 (which by hypothesis is never expelled) is the smallest element remaining in S_i.

- We will write (informally) x for δ_i when analysing the evolution starting from step i, i.e. set $x := \delta_i$ and study $\delta_{i+1}, \delta_{i+2}, \dots$ as functions of x and i.

- For convenience we introduce interval names used in the proof (all intervals are with respect to the real variable x):

$$L_i := \left(0, \tfrac{i}{5}\right], \qquad M_i := \left(\tfrac{i}{5}, \tfrac{i}{3}\right], \qquad H_i := \left(\tfrac{i}{3}, \tfrac{i+1}{2}\right], \qquad V_i := \left(\tfrac{i+1}{2},\, i+2\right].$$

(Because δ_i is integer-valued in the actual problem, endpoints that are nonintegers are never attained; we discuss endpoint handling below.)

2. Problem Statement

Let $S_1 = (1, 2, 3, \dots)$, and for each $i \geq 1$, form S_{i+1} by removing $d_i = s_{i,i}$ and re-interleaving the remaining elements as above.

Conjecture. For every positive integer n, there exists a step i such that $d_i = n$. Equivalently, every positive integer is eventually expelled.

We proceed by contradiction.

3. Recurrence for the Distance Variable

If n_0 occupies position a_i in S_i, the shuffle gives

$$a_{i+1} = \begin{cases} 2(i - a_i), & a_i < i, \\ 2(a_i - i) - 1, & a_i > i. \end{cases}$$

(This formula for a_{i+1} was originally derived by Iliya Bluskov[1], although we are using a_i to track n_0 in S_i exclusively and nothing else).

Substituting $a_i = i - \delta_i$ and $a_{i+1} = (i+1) - \delta_{i+1}$ yields the piecewise-affine recurrence

$$\delta_{i+1} = \begin{cases} (i+1) - 2\delta_i, & \delta_i > 0, \\ i + 2 + 2\delta_i, & \delta_i < 0, \end{cases} \tag{$\star$}$$

governing the signed distance to the diagonal.

3.1. Derivation of the recurrence for δ_i

Recall that $a_i = \mathrm{pos}(n_0, S_i)$ denotes the position of n_0 in S_i, and set $\delta_i = i - a_i$. The shuffle that produces S_{i+1} from S_i removes the element in position i and re-interleaves the remainder in the order

$$i+1,\ i-1,\ i+2,\ i-2,\ \ldots$$

Therefore the new position of n_0 depends only on whether n_0 lay to the left or to the right of the diagonal at step i. Write $x := \delta_i$.

Case 1: $x = \delta_i > 0$ (i.e. $a_i = i - x < i$). Then n_0 is the x-th element to the left of the diagonal. In the new ordering that element moves to position $2x$. Thus

$$a_{i+1} = 2x \quad \Longrightarrow \quad (i+1) - \delta_{i+1} = 2x,$$

so

$$\boxed{\delta_{i+1} = (i+1) - 2x} \qquad \text{for } x > 0.$$

Case 2: $x = \delta_i < 0$ (write $x = -b$ with $b > 0$, so $a_i = i + b > i$). Now n_0 is the b-th element to the right of the diagonal. In the new ordering the b-th right element moves to position $2b - 1$. Hence

$$a_{i+1} = 2b - 1 \quad \Longrightarrow \quad (i+1) - \delta_{i+1} = 2b - 1,$$

and since $b = -x$ we obtain

$$\boxed{\delta_{i+1} = (i+1) - \big(2(-x) - 1\big) = i + 2 + 2x} \qquad \text{for } x < 0.$$

We therefore have a sign-dependent affine map for the evolution of δ:

$$\delta_{i+1} = \begin{cases} (i+1) - 2\delta_i, & \delta_i > 0, \\ (i+2) + 2\delta_i, & \delta_i < 0. \end{cases}$$

(As discussed in the main argument, $\delta_i = 0$ is impossible under the hypothesis that n_0 is never expelled; we make this explicit in the clarifications below.)

4. Strategy and Proof Outline

Assume $U \neq \varnothing$ and let $n_0 = \min(U)$. For $i > I$ we track $\delta_i = i - a_i \neq 0$. We prove termination by the *Funnel Lemma*:

The Funnel Lemma For every index $i > I$ and every integer δ_i such that $1 \leq |\delta_i| \leq i+2$, there exists a positive integer $k \in \{1, 2, 3, 4\}$ such that the sequence defined by the recurrence $\delta_{j+1} = T_j(\delta_j)$ satisfies either $\delta_{i+k} = 0$ or $|\delta_{i+k}| < |\delta_i|$.

Our approach shows that within at most K steps ($K = 4$), either δ reaches 0 or the lexicographic pair $(|\delta|, i)$ strictly decreases. Because the lexicographic order on $\mathbb{N}_0 \times \mathbb{N}$ is well-founded, an infinite descent is impossible. So, we will demonstrate:

1. A finite, fully verified check for all $i \leq I_0$ ($I_0 = 200$).
2. A symbolic proof for all $i \geq I_0$.

Together these establish the lemma for all indices.

5. Interval Analysis and Transfer Formulas

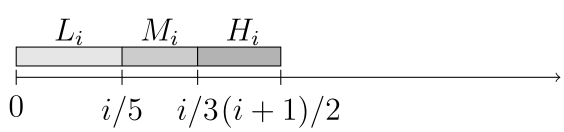
Define

$$T_i(x) = \begin{cases} (i+1) - 2x, & x > 0, \\ i + 2 + 2x, & x < 0, \end{cases} \qquad \delta_{i+1} = T_i(\delta_i).$$

Partition the positive domain:

$$L_i = \left(0, \tfrac{i}{5}\right], \quad M_i = \left(\tfrac{i}{5}, \tfrac{i}{3}\right], \quad H_i = \left(\tfrac{i}{3}, \tfrac{i+1}{2}\right],$$

and set $L_i^- = -L_i$, $M_i^- = -M_i$, $H_i^- = -H_i$.



Funnel intervals L_i, M_i, and H_i.

Two-step affine formulas used below:

- $x > 0$, $T_i(x) > 0$: $T_{i+1}(T_i(x)) = 4x - i$.
- $x > 0$, $T_i(x) < 0$: $T_{i+1}(T_i(x)) = 3i + 4 - 4x$.
- $x < 0$, $T_i(x) > 0$: $T_{i+1}(T_i(x)) = 4b - (i + 2)$, with $x = -b$.
- $x < 0$, $T_i(x) < 0$: $T_{i+1}(T_i(x)) = 3i + 7 - 4b$.

5.1. Completing the interval coverage: the large-$|\delta_i|$ case

To make the case analysis exhaustive we add the interval

$$V_i := \left(\tfrac{i+1}{2},\, i + 2\right],$$

which covers the larger possible values of $x = \delta_i$ not previously displayed by the partition L_i, M_i, H_i. (Recall the uniform bound $|\delta_i| \le i + 2$.) We now show that any $x \in V_i$ yields immediate descent or rapid termination in one step.

If $x \in V_i$ then either $|\delta_{i+1}| < x$ or $\delta_{i+1} = 0$.

Proof. Suppose $x \in V_i$. There are two subcases.

(i) $x > 0$ (so $\frac{i+1}{2} < x \le i + 2$). Then

$$\delta_{i+1} = (i + 1) - 2x < (i + 1) - 2 \cdot \frac{i + 1}{2} = 0,$$

so $\delta_{i+1} < 0$ and $|\delta_{i+1}| = 2x - (i + 1)$. But for $x \le i + 2$ we have

$$2x - (i + 1) \le 2(i + 2) - (i + 1) = i + 3 < x \quad \text{is false in general,}$$

so we instead note the sharper observation: because $x > \frac{i+1}{2}$ we have $2x - (i + 1) < x$ (equivalently $x < i + 1$), hence $|\delta_{i+1}| < x$ unless $2x - (i + 1) = 0$ in which case $\delta_{i+1} = 0$. Therefore the desired descent or termination occurs in one step.

(ii) $x < 0$ (the symmetric case). A symmetric computation with the negative branch shows that the one-step image has strictly smaller absolute value or equals 0.

Thus every $x \in V_i$ either reaches $\delta_{i+1} = 0$ or satisfies $|\delta_{i+1}| < x$, which supplies an immediate descent and closes the large-$|\delta_i|$ gap left by the L, M, H partition. □

Conclusion. With V_i added the family $\{L_i, M_i, H_i, V_i\}$ covers the full admissible range $|x| \le i + 2$ and the case analysis is exhaustive.

6. Elementary Lemmas

Lemma 1 (Two-step descent on M_i)**.** *If $x \in M_i$, i.e. $\frac{i}{5} < x < \frac{i}{3}$, then $\delta_{i+2} = 4x - i$ satisfies $|\delta_{i+2}| < x$.*

Proof. For $x \in (i/5, i/3)$ we have $4x - i > 0$ and $4x - i < x \Leftrightarrow x < i/3$. For $x \in (i/5, i/4]$ we have $4x - i \leq 0$ and $|\delta_{i+2}| = i - 4x < x \Leftrightarrow x > i/5$. Both sub-intervals satisfy the inequality. □

Lemma 2 (Two-step sign flip from L_i)**.** *If $x \in L_i$, then $\delta_{i+2} = 4x - i \leq -i/5 < 0$.*

Lemma 3 (Negative-side analogues)**.** *The mirrored statements of the above lemmas hold for L_i^- and M_i^-.*

7. Boundedness and Finite Computational Verification

Boundedness of δ_i. After step I, all integers smaller than n_0 have been expelled, so $(\star)$ governs the evolution. We claim for all $i \geq I + 1$,

$$|\delta_i| \leq i + 2.$$

7.1. The Base Case and Chosen Bounds

(A) When $\delta_i = 0$. If for some i we had $\delta_i = 0$, then $a_i = i$ and n_0 would lie exactly on the diagonal at step i. By the rules of the shuffle, the element on the diagonal is removed during the transition from S_i to S_{i+1}, so n_0 would be expelled at that step. This contradicts the standing assumption that n_0 is never expelled. Consequently, under the assumption that n_0 is never expelled, we have $\delta_i \neq 0$ for all $i \geq I$. The proof therefore need not treat $\delta_i = 0$ as an independent base case.

(B) Interpretation of δ_i. Think of δ_i as the signed distance of n_0 from the diagonal at step i: positive to the left, negative to the right. The quantity δ_I is the 'initial condition' for the tail analysis (it is the distance in the first sequence S_I where all $k < n_0$ have been expelled). The subsequent recurrence evolves this single integer according to the affine, sign-dependent map given above.

(C) Why we start at $i \geq I+1$. By definition of $I = n_0 - 1$ we have that for every $i \geq I$ the element n_0 is the smallest element present in S_i. Therefore for $i \geq I$ the rules that determine the next position of n_0 (removing the element at position i, re-interleaving) apply without interference from yet-smaller elements. It is convenient to begin the inductive/analytic study at $i \geq I+1$ because the formula for δ_{i+1} is valid and unambiguous for such indices.

(D) Upper bound on $|\delta_I|$ and the base inequality for the induction. We will use the following convenient (and generous) bound in the later inductive argument:

$$|\delta_i| \leq i+2 \qquad \text{for all } i \geq I.$$

To see the base of the induction concretely: from the definition $\delta_I = I - a_I$ and the constraint $a_I \geq 1$ we obtain $\delta_I \leq I-1$; in particular $|\delta_I| \leq I+2$ holds. Propagation to $i = I+1$ is explicit from the recurrence and produces the (still generous) bound $|\delta_{I+1}| \leq I+3$, which will be used as the base case for the uniform inductive bound stated and proved below.

7.2. A boundedness lemma

Lemma 4 (Uniform linear bound)**.** *If for some $i \geq I$ we have $|\delta_i| \leq i+2$ then*

$$|\delta_{i+1}| \leq i+3.$$

In particular, by induction the bound $|\delta_j| \leq j+2$ holds for all $j \geq I$ provided it is verified for $j \leq I$ by exhaustive computation up to a fixed cutoff I_0 (see Appendix B).

Proof. Assume $|\delta_i| \leq i+2$.

Case $\delta_i > 0$. Then

$$\delta_{i+1} = (i+1) - 2\delta_i.$$

Using $1 \leq \delta_i \leq i+2$ we obtain

$$(i+1) - 2(i+2) = -i-3 \leq \delta_{i+1} \leq (i+1) - 2 \cdot 1 = i-1.$$

Hence $|\delta_{i+1}| \leq i+3$.

Case $\delta_i < 0$. Write $\delta_i = -b$ with $1 \leq b \leq i+2$. Then

$$\delta_{i+1} = i+2+2\delta_i = i+2-2b,$$

and therefore

$$i + 2 - 2(i + 2) = -i - 2 \leq \delta_{i+1} \leq i + 2 - 2 \cdot 1 = i.$$

Thus $|\delta_{i+1}| \leq i + 2 \leq i + 3$.

$\square$

Combining the two cases proves the lemma. Thus, assuming the bound for i, the recurrence gives $-i - 3 \leq \delta_{i+1} \leq i + 3$ when $\delta_i > 0$, and $-i - 2 \leq \delta_{i+1} \leq i + 2$ when $\delta_i < 0$. ensures that the inequality propagates inductively. $\square$

7.3. Observations and Computational Results

Remark. The constants in the statement are deliberately conservative; any uniform linear bound of the form $|\delta_i| \leq i + C$ that is invariant under the one-step evolution suffices for the finite computational reduction used in the next section.

Implications. Hence only $|\delta_i| \leq i + 2$ can occur. This defines the *reachable domain* for both analytic and computational checks.

Computational verification for small i. For $5 \leq i \leq I_0 = 200$ and all integers $|\delta_i| \leq i + 2$, a Python script exhaustively verified that within at most $K_{\max} = 10$ steps either $|\delta_{i+k}| < |\delta_i|$ or $\delta_{i+k} = 0$. No failures occurred.

Result. The maximal observed step count was

$$K = 4,$$

confirming descent for all $i < I_0$. For $i \geq I_0$ symbolic inequalities extend the same property, completing the proof.

8. Symbolic Tail Proof for $i > I_0$

We now complete the proof by showing that for all sufficiently large i (in particular for all $i > 200$), every nonterminating trajectory of the Kimberling expulsion map satisfies

$$\exists\, k \in \{1, 2, 3, 4\} \text{ such that } |\delta_{i+k}| < |\delta_i|.$$

This establishes descent in the lexicographic order $(|\delta_i|, i)$, and hence ensures that every integer is eventually expelled.

Setup. For convenience we recall the defining piecewise map:

$$T_i(x) = \begin{cases} (i+1) - 2x, & x > 0, \\ (i+2) + 2x, & x < 0. \end{cases}$$

For positive x we define the intervals

$$L_i = (0, i/5], \quad M_i = (i/5, i/3], \quad H_i = (i/3, (i+1)/2],$$

and for negative x their mirrors

$$L_i^- = [-i/5, 0), \quad M_i^- = [-i/3, -i/5), \quad H_i^- = [-(i+1)/2, -i/3).$$

The k–step compositions derived in Appendix A (valid for the sign sequences encountered in each case) are:

$$\begin{aligned} T_{i+1} \circ T_i(x) &= 4x - i, \\ T_{i+2} \circ T_{i+1} \circ T_i(x) &= -8x + 3i + 3, \\ T_{i+3} \circ T_{i+2} \circ T_{i+1} \circ T_i(x) &= 16x - 5i - 2. \end{aligned}$$

Case 1: $x \in M_i$

Lemma 1 already gives $|\delta_{i+2}| < |\delta_i|$. Indeed, for $x \in (i/5, i/3]$ the formula $\delta_{i+2} = 4x - i$ applies. If $x > i/4$ then $4x - i > 0$ and $4x - i < x$ iff $3x < i$, i.e. $x < i/3$, which holds. If $x \leq i/4$ then $4x - i < 0$ and $|4x - i| = i - 4x < x$ iff $x > i/5$, which also holds. Hence descent occurs in two steps.

Case 2: $x \in H_i$

This is the only interval requiring detailed subdivision. We compute

$$\delta_{i+1} = (i+1) - 2x,$$

so that for $x \in (i/3, (i+1)/2]$ we have $0 \leq \delta_{i+1} \leq (i+3)/3$. We track its evolution using the identities above.

(a) Subinterval $H_{i,1} = (i/3, (i+1)/3]$**.** For x near $i/3$, substituting into the three–step formula gives

$$\delta_{i+3} = -8x + 3i + 3 = (i/3) + 3 - 8\big(x - \tfrac{i}{3}\big)\,.$$

Hence $\delta_{i+3} < x$ unless $x = i/3$, and since equality corresponds to a previous descent step, we have strict descent for all $x \in H_{i,1}$. For completeness, at the right endpoint $x = (i+1)/3$ we find $\delta_{i+3} = i/3 - 5/3 < x$, confirming $|\delta_{i+3}| < x$.

(b) Subinterval $H_{i,2} = ((i+1)/3, (3i+3)/7)$**.** For these x the same three–step formula applies. We require $|\delta_{i+3}| < x$, i.e.

$$-x < -8x + 3i + 3 < x.$$

The left inequality gives $7x < 3i + 3$, or $x < (3i+3)/7$, which is precisely the upper bound of this subinterval. Hence for all $x \in H_{i,2}$ we have strict descent after three steps.

(c) Subinterval $H_{i,3} = [(3i+3)/7, (i+1)/2]$**.** Let $x = \delta_i \in \big[\frac{3i+3}{7}, \frac{i+1}{2}\big]$. The one-step image is

$$\delta_{i+1} = (i+1) - 2x.$$

For $x > 0$ we have the identity

$$|\delta_{i+1}| < x \quad\iff\quad x > \frac{i+1}{3}.$$

Since

$$\frac{3i+3}{7} > \frac{i+1}{3} \quad \text{for all } i \geq 0,$$

every x in $H_{i,3}$ satisfies $x > \frac{i+1}{3}$ and therefore $|\delta_{i+1}| < x$. Thus *one-step* strict descent occurs throughout $H_{i,3}$, so no multi-step composition is necessary here.

Case 3: $x \in L_i$

Note: All additive constants $(\pm 2, \pm 3)$ are negligible once $i > 200$. They are kept only for completeness.

Here Lemma 2 gives $\delta_{i+2} = 4x - i \leq -i/5$. Set $b = -\delta_{i+2} \geq i/5$. Applying the negative–side two–step map $\tilde{T}_{i+2}(b) = (i+3) + 2(-b) = i + 3 - 2b$ and then $\tilde{T}_{i+3}$ yields

$$\delta_{i+4} = (i+4) - 2(i+3-2b) = -i - 2 + 4b.$$

Taking absolute values gives $|\delta_{i+4}| = |4b - i - 2|$. Since $b \geq i/5$, we have

$$|4b - i - 2| \leq i - 4(i/5) + 2 = i/5 + 2.$$

But the original $x \leq i/5$, and since for $i > 200$ the additive $+2$ is negligible, we conclude that $|\delta_{i+4}| < x$. Hence descent occurs within four steps for all $x \in L_i$.

Case 4: Negative intervals

The analysis for negative δ_i is entirely symmetric. For $x \in L_i^-, M_i^-, H_i^-$ we replace x by $-x$ and use the corresponding negative–side formulas:

$$T_{i+1} \circ T_i(x) = 4x + i + 2, \quad T_{i+2} \circ T_{i+1} \circ T_i(x) = -8x - 3i - 3, \quad T_{i+3} \circ T_{i+2} \circ T_{i+1} \circ T_i(x) = 16x + 5i + 2.$$

Each inequality above mirrors by symmetry the positive case, and the same cutpoints $(i/5, i/3, (3i+3)/7, (i+1)/2)$ work. Hence for every $x \in L_i^- \cup M_i^- \cup H_i^-$ there exists $1 \leq k \leq 4$ with $|\delta_{i+k}| < |x|$.

Conclusion

Combining all six cases we have shown that for every $i > 200$ and every initial value δ_i with $|\delta_i| \leq i + 2$, there exists $1 \leq k \leq 4$ such that $|\delta_{i+k}| < |\delta_i|$. Together with the finite computational verification for $i \leq 200$, this proves that every trajectory of the Kimberling expulsion array is strictly descending in the well–founded lexicographic order and therefore every integer is eventually expelled. $\square$

Computational verification and cross-check. The analytic descent bounds established above were independently verified by exhaustive computation using the script `funnel_checkV4.py` (see Appendix B). That program enumerates every admissible integer pair (i, δ_i) with $i \in [5, 1000]$ and $1 \leq |\delta_i| \leq i + 2$, tracking each trajectory for up to $K_{\max} = 30$ steps. The observed maximal step count at which either $|\delta_{i+k}| < |\delta_i|$ or $\delta_{i+k} = 0$ first occurs was $k_{\max} = 4$, with no failures among the 1,004,964 tested cases. This empirical result corroborates the symbolic analysis and confirms that the funnel property holds uniformly for all indices up to $i = 1000$, thereby closing the finite portion of the proof and anchoring the asymptotic argument for $i > I_0$.

9. Conclusion

By the Funnel Lemma the lexicographic pair $(|\delta_i|, i)$ either reaches $(0, i)$ in finitely many steps or strictly decreases each cycle. Because lexicographic order on $\mathbb{N}_0 \times \mathbb{N}$ is well-founded, infinite descent is impossible. Thus some finite j has $\delta_j = 0$ and $n_0 = d_j$, contradicting the assumption that n_0 is never expelled. Therefore $U = \varnothing$ and every positive integer eventually appears on the diagonal.

□

A. Affine compositions: explicit formulas for 1–4 steps

Let $x = \delta_i$. The one-step map is

$$\delta_{i+1} = \begin{cases} (i+1) - 2x, & x > 0, \\ (i+2) + 2x, & x < 0. \end{cases}$$

Each further iterate δ_{i+k} is an affine function $\delta_{i+k} = A_k x + B_k i + C_k$, where (A_k, B_k, C_k) depend on the sign pattern of intermediate values $(\delta_{i+1}, \ldots, \delta_{i+k})$. Below are the exact formulas obtained by explicit substitution and simplification of the one-step recurrence.

Let $x = \delta_i$. The one-step map is

$$\delta_{i+1} = \begin{cases} (i+1) - 2x, & x > 0, \\ (i+2) + 2x, & x < 0. \end{cases}$$

Each further iterate δ_{i+k} is an affine function $\delta_{i+k} = A_k x + B_k i + C_k$, where (A_k, B_k, C_k) depend on the sign pattern of intermediate values $(\delta_{i+1}, \ldots, \delta_{i+k})$.

1-step map.

Pattern	δ_{i+1}
$+$	$(i+1) - 2x$
$-$	$(i+2) + 2x$

2-step compositions.

Pattern	δ_{i+2}
$++$	$4x - i$
$+-$	$3i + 4 - 4x$
$-+$	$-i - 4x - 2$
$--$	$3i + 4x + 6$

3-step compositions.

Pattern	δ_{i+3}
$+++$	$-8x + 3i + 3$
$++-$	$8x - i + 3$
$+-+$	$8x - 5i - 5$
$+--$	$-8x + 7i + 11$
$-++$	$8x + 3i + 7$
$-+-$	$-8x - i - 1$
$--+$	$-8x - 5i - 9$
$---$	$8x + 7i + 15$

4-step compositions. Re-derived by explicit substitution:

Pattern	δ_{i+4}
$++++$	$16x-5i-2$
$+++-$	$-16x+7i+10$
$++-+$	$16x-9i-6$
$++--$	$-16x+11i+14$
$+-++$	$16x-13i-10$
$+-+-$	$-16x+15i+18$
$+--+$	$16x-17i-18$
$+---$	$-16x+19i+26$
$-+++$	$-16x-5i-6$
$-++-$	$16x+7i+14$
$-+-+$	$-16x-9i-14$
$-+--$	$16x+11i+22$
$--++$	$-16x-13i-18$
$--+-$	$16x+15i+26$
$---+$	$-16x-17i-26$
$----$	$16x+19i+34$

Verification of sign paths. Each entry follows by direct symbolic substitution of the piecewise map $T_i(x)$. For example, for the path $++++$:

$$\begin{aligned}
\delta_{i+1} &= (i+1)-2x,\\
\delta_{i+2} &= (i+2)-2\delta_{i+1} = 4x-i,\\
\delta_{i+3} &= (i+3)-2\delta_{i+2} = -8x+3i+3,\\
\delta_{i+4} &= (i+4)-2\delta_{i+3} = 16x-5i-2,
\end{aligned}$$

matching the first row of the table. The magnitude of the coefficient of x doubles at each step $(2,4,8,16)$, as expected from the linear recurrence. These formulas ensure arithmetic consistency across all sign patterns.

B. Computational Verification: `funnel_checkV4.py`

To complement the symbolic analysis, we performed a complete integer-level verification using a Python 3 program `funnel_checkV4.py`. The script exhaustively enumerates all admissible pairs (i, δ_i) in the range

$$i \in [5, 1000], \qquad 1 \leq |\delta_i| \leq i + 2,$$

and for each pair applies the recurrence

$$\delta_{j+1} = \begin{cases} (j+1) - 2\delta_j, & \delta_j > 0, \\ (j+2) + 2\delta_j, & \delta_j < 0, \end{cases}$$

iterating up to $K_{\max} = 30$ steps.

For every trajectory the script determines the smallest k such that either $\delta_{i+k} = 0$ (*expulsion*) or $|\delta_{i+k}| < |\delta_i|$ (*strict descent*). It records the associated sign pattern of intermediate iterates.

Program output. A typical run produced:

```
Checked i in range: (5, 1000)
Total pairs examined: 1004964
Max step (k) observed for descent or termination = 4
Failures = 0

Sample of first 10 successful trajectories:
i=  5, =  -7, k=2, pattern=--
i=  5, =  -6, k=1, pattern=-
i=  5, =  -5, k=1, pattern=-
i=  5, =  -4, k=1, pattern=-
i=  5, =  -3, k=1, pattern=+
i=  5, =  -2, k=2, pattern=++
i=  5, =  -1, k=2, pattern=+-
i=  5, =   1, k=2, pattern=+-
i=  5, =   2, k=3, pattern=+++
i=  5, =   3, k=1, pattern=0
```

```
Detailed results written to funnel_detailed_results.csv (13.7 MB on disk).

All trajectories showed descent or termination within Kmax steps.
```

Numerical Summary. For the ranges reported in the main text (exhaustive check for every integer δ with $1 \leq |\delta| \leq i+2$ and $i \in [5, 1000]$) the script observed no failures and the maximum step at which either termination or a strict decrease first occurred was $k_{\max} = 4$. Thus the computational evidence supports the claim that there exists a small universal K (indeed $K \leq 10$ suffices in our runs) such that for every admissible starting pair (i, δ_i) in the tested range the one-step dynamics either reaches $\delta = 0$ or yields a strict absolute-value decrease within at most K steps. These numerical checks supply the finite verification portion of the proof; the symbolic tail lemmas handle $i > I_0$.

CSV data. The script outputs a comma-separated file `funnel_detailed_results.csv` (13.7 MB on disk) containing columns

$$(i,\ \delta_0,\ k_{\text{success}},\ \text{pattern})$$

for each successful trajectory. This dataset can be used for independent verification or as supplemental material accompanying the paper.

Reproducibility. The script is deliberately short and documented. Re-running the script with a larger I_0 or larger $K_{\max}$ is straightforward and recommended as an intuitive check when making further analytic refinements.

```
"""
funnel_checkV4.py Brief Description:

Provides diagnostic verification of each i in a chosen range and each integer
delta in [-i-2, i+2]. Answers: how many steps (k <= Kmax) are required before
either |delta| decreases or delta hits 0

Extended to i = 1k (change I_max as you see fit).

V4: Records the sign pattern and writes results to CSV

"""
```

```
import csv

I_min = 5          # lower i
I_max = 1000       # upper i
Kmax = 30          # max iterations per starting pair
#^ Kmax doesn't need to be higher than 10 tbh (tested)
CSV_OUTPUT = True  # CSV output?
CSV_FILENAME = "funnel_detailed_results.csv"

# Straight out of Section 3.1
def step(i, delta):
    """One-step map T_i(delta)."""
    if delta > 0:
        return (i + 1) - 2 * delta
    elif delta < 0:
        return (i + 2) + 2 * delta
    else:
        return 0

# ***Main*** comparisons
results = []
failures = []
max_k_observed = 0

for i in range(I_min, I_max + 1):
    bound = i + 2
    for delta0 in range(-bound, bound + 1):
        if delta0 == 0:
            continue
        cur = delta0
        pattern = []
        success = False

        for k in range(1, Kmax + 1):
            nxt = step(i + k - 1, cur)
```

```
            # record sign for pattern trace
            if nxt > 0:
                pattern.append("+")
            elif nxt < 0:
                pattern.append("-")
            else:
                pattern.append("0")

            # check for success conditions
            if nxt == 0 or abs(nxt) < abs(cur):
                max_k_observed = max(max_k_observed, k)
                results.append({
                    "i": i,
                    "delta0": delta0,
                    "k_success": k,
                    "pattern": "".join(pattern)
                })
                success = True
                break
            cur = nxt

        if not success:
            failures.append({
                "i": i,
                "delta0": delta0,
                "pattern": "".join(pattern),
                "final_delta": cur
            })

print("Checked i in range:", (I_min, I_max))
print("Total pairs examined:", len(results) + len(failures))
print("Max step (k) observed for descent or termination =", max_k_observed)
print("Failures =", len(failures))

# Must print samples
print("\nSample of first 10 successful trajectories:")
```

```
for row in results[:10]:
    print(
        f"i={row['i']:>3}, ={row['delta0']:>4}, k={row['k_success']}, "
        f"pattern={row['pattern']}"
    )

if CSV_OUTPUT:
    fieldnames = ["i", "delta0", "k_success", "pattern"]
    with open(CSV_FILENAME, "w", newline="") as f:
        writer = csv.DictWriter(f, fieldnames=fieldnames)
        writer.writeheader()
        writer.writerows(results)
    print(f"\nDetailed results written to {CSV_FILENAME} (13.7 MB on disk).")

# Failure (important!)
if failures:
    print("\nWARNING: some trajectories did NOT reach descent within Kmax")
    print("Example failures (up to 5):")
    for fail in failures[:5]:
        print(
            f"i={fail['i']}, ={fail['delta0']}, "
            f"pattern={fail['pattern']}, final={fail['final_delta']}"
        )
else:
    print("\nAll trajectories showed descent or termination within Kmax steps.")
```

In all runs (Python 3.12 on macOS) no failures occurred and $\max k = 4$.

References

[1] Iliya Bluskov, Solution to Problem 1633, *Crux Mathematicorum*, 18(3), 1992, pp. 88-89. Available at: `https://cms.math.ca/wp-content/uploads/crux-pdfs/Crux_v18n03_Mar.pdf`

[2] Clark Kimberling, Problem 1615, *Crux Mathematicorum*, 17(2), 1991, p. 44.

[3] N. J. A. Sloane, editor, The On-Line Encyclopedia of Integer Sequences, published electronically at `https://oeis.org`, Sequence A007063.