

The density of oscillating agars in Conway's Game of Life

Results, methods and open problems. Revised version, September 2026.

The problem

An **oscillating agar** is a pattern that fills the whole plane periodically and returns to itself after p generations. How large can its average density be?

- The conjectured maximum is **$1/2$** .
- For still lifes this was proved by Noam Elkies (*The still-Life density problem and its generalizations*, 1998, arXiv:math/9905194). He showed more: any set in which every live cell has at most 3 live neighbours has density at most $1/2$.
- For oscillators, the best published upper bound we found is **$1176/2087 \approx 0.563$** , due to Keith Amling (2024; reported on LifeWiki). Earlier bounds were $8/13$, $17/28$ and $320/559$.

Contribution

1. A new upper bound. Every oscillating agar, of any period, has density at most **$205847955/400000000 \approx 0.51462$** .

2. An exclusion theorem. Every oscillating agar in which none of a fixed list of 769 two-generation 3×3 patterns occurs has density at most **$1/2$** . So any counterexample to the conjecture must contain some of them.

Both results come from one linear program that works for all periods at once, combined with certificates checked in exact arithmetic.

The conjecture itself is still open.

Status of the computations. The computer-assisted results rely on the scripts described in Section 8. They have not yet been checked by an independent implementation, and the SAT results in Section 4 were not accompanied by checkable proof files. Both are listed as open tasks in Section 9.

Contents

1. Definitions
2. Summary of results
3. The all-period linear program and its certificates
4. The exclusion theorem
5. Results proved by hand
6. Agars with a small spatial period
7. Approaches that did not lead anywhere
8. Reproducing the results
9. Open problems

1. Definitions

Basic notions

- **Life** is the rule B3/S23 on the square grid. Each cell has 8 neighbours, labelled as in Figure 1.
- An **agar** is a pattern that repeats in both directions, equivalently a pattern on a finite torus. Every agar repeats along some rectangle, so rectangular tori are enough.
- An **oscillating agar of period p** has phases $A_0, \dots, A_{p-1}$. Each phase is the Life image of the one before, and $A_p = A_0$. Oscillators whose period divides p are included.
- Its **average density** is the mean, over the p phases, of the fraction of live cells.

NW	N	NE
W	x	E
SW	S	SE

Figure 1. The eight neighbours of a cell x .

Cells

- A live cell with at least 4 live neighbours dies in the next generation. We call it **overcrowded**.
- In a period-2 oscillator every cell has one of four **types**: S (always alive), D (always dead), X (alive in phase A only) or Y (alive in phase B only). X and Y cells are called **rotor** cells.

■ always alive (S) ■ alive in phase A only (X) ■ alive in phase B only (Y) □ always dead (D)

Windows

- A **window** is the 3×3 block around a cell, recorded at time t and at time $t+1$. Larger $h \times w$ windows are defined the same way.
- Each cell of a window is in one of four states: dead at both times, alive then dead, dead then alive, or alive at both times.
- A window is **reachable** if some pattern at time t produces it under Life. Of the 262,144 possible 3×3 windows, exactly **87,851** are reachable (found by running all 2^{25} patterns of size 5×5).

The patterns that matter

- A **vanishing block** is a 2×2 block with at least 3 live cells at time t and no live cells at time $t+1$.
- A **culprit window** is a window centred on a cell of a vanishing block that has a surviving neighbour directly above, below, left or right of it, outside the block (Figure 2).

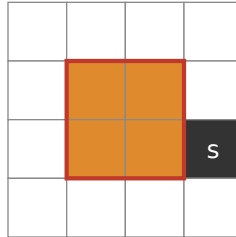


Figure 2. A vanishing block (orange: alive at t , dead at $t+1$). The cell s survives from t to $t+1$ and touches the block along an edge, so the windows centred on the block cells next to s are culprit windows.

2. Summary of results

Main results (computer-assisted)

	Result	How it is checked
A	Every oscillating agar has density at most $205847955/400000000 \approx 0.5146199$ (3×4 windows)	exact integer arithmetic
B	Every oscillating agar has density at most $51612905/100000000 \approx 0.5161291$ (3×3 windows)	exact integer arithmetic
C	Exclusion theorem: if none of 769 listed 3×3 window types occurs, the density is at most $1/2$	exact rational arithmetic
D	If none of 3,350 listed 3×4 window types occurs, the density is at most 0.500000010	exact integer arithmetic
E	Density at most $1/2$ for all agars with a small spatial period in one direction (Section 6)	exact

Supporting results (proved by hand)

	Result	Remark
F	Any set in which every live cell has at most 3 live neighbours has density at most $1/2$	Elkies' theorem; we give a short discharging proof
G	A period-2 agar in which no two rotor cells of the same type are adjacent has density at most $1/2$	narrow, but a genuine consequence of F
H	If the live cells have k live neighbours on average, the density is at most $(k + 4)/12$	standard eigenvalue bound
I	The family-1 band forces the rows beside it, and together they have density below $1/2$	period 2, band as a strip
J	The edge rows of the family-2 strip cannot occur	strips repeating every 2 columns

3. The all-period linear program and its certificates

A single linear program (LP) gives an upper bound for oscillators of every period at once.

3.1 Variables and constraints

There is one variable $x_w \geq 0$ for each reachable window w : 87,851 variables for 3×3 windows and 2,175,813 for 3×4 windows. Take an oscillator of period p on a torus with n cells, and let x_w be the fraction of the $n \cdot p$ pairs (cell, time) at which the window is w . These numbers satisfy:

- **Reachability.** Only reachable windows occur.
- **Spatial consistency.** For every pattern P , the windows whose left part is P and the windows whose right part is P have the same total frequency, because shifting by one column permutes the cells of the torus. The same holds for top and bottom parts.
- **Stationarity.** The patterns seen at time t and those seen at time $t+1$ occur with the same frequencies, because $t+1$ runs through the same p phases as t .
- **Normalisation.** The x_w add up to 1.

The density of the oscillator is the total frequency of windows whose centre is alive at time t . None of this depends on p , so the maximum of the LP bounds the density of every oscillator.

LP	Variables	Maximum
period 2 only, 3×3	66,320	≈ 0.5161290 (16/31)
period 2 only, 3×4	1,124,513	≈ 0.5146199 (88/171)
all periods, 3×3	87,851	≈ 0.5161290 (16/31)
all periods, 3×4 (using symmetry)	547,016 groups of 2,175,813	≈ 0.5146198 (88/171)

The fractions in brackets are the values the floating-point optima agree with, to within solver accuracy (about 10^{-7}); they were not proved exact. The proven upper bounds are those listed in Section 2.

3.2 Turning the LP into a proof

Choose any number u_r for each consistency or stationarity constraint. These constraints are zero for every valid solution x , so

$$\text{density} = \sum_w x_w (c_w - (A^T u)_w) \leq \max_w (c_w - (A^T u)_w),$$

where c_w is 1 if the centre of w is alive at time t and 0 otherwise. Any choice of u gives a valid bound, and checking it only needs a maximum over finitely many windows. Two kinds of check

were used:

- **Rounded check** (results A, B, D). Take the multipliers found by the solver, multiply by 10^8 , round to integers and compute the maximum in exact integer arithmetic. All constraint coefficients are 0 or ± 1 .
- **Exact check** (result C). Take an optimal basis from the solver HiGHS, compute the multipliers exactly as fractions with the library FLINT (a system of 6,764 equations), and check every window with exact fractions. The bound came out as exactly $1/2$.

3.3 Symmetry

Reflecting windows left-right or up-down maps windows to windows and each constraint to another constraint, possibly with its sign reversed. This follows from how the constraints are defined, and we also checked it numerically for window sizes 3×3 , 2×3 , 3×2 and 2×4 .

Consequences:

- There is an optimal solution unchanged by reflections, so windows can be grouped with their mirror images. This makes the LP up to 4 times smaller (3×3 : about 38 s $\rightarrow$ 3 s) and made 3×4 practical.
- For the rounded check, the multipliers are averaged over the four reflections. The averaged multipliers still satisfy the weak-duality argument above, because each reflection only permutes and re-signs constraints.
- The objective reads density off the centre cell and its mirror images. Spatial consistency makes all interior cells equally likely to be alive, so this does not change the density.

4. The exclusion theorem

4.1 The idea

Remove some window types from the LP and solve it again. If the maximum drops to $1/2$, every oscillator with density above $1/2$ must contain one of the removed types. This gives a rigorous theorem about any class of windows, and it shows which local structures are responsible for the gap between the LP bound and $1/2$.

Window types removed	LP maximum
none	≈ 0.516129
all culprit windows (2,480 types)	$1/2$ (exact certificate)

4.2 Removing only what is needed

Most culprit windows play no role. We therefore removed, in rounds, only the culprit windows that are *tight* at the current optimum: those with zero reduced cost, which the solver could use without loss.

Round	LP maximum	Windows removed
0	≈ 0.516129	405 (family 1)
1	0.505279	364 (family 2)
2	$1/2$	none needed

Exclusion theorem (result C). Let E be the 769 window types removed in rounds 0 and 1. Every oscillating agar, of any period, in which no window of E occurs has average density at most $1/2$.

This removal procedure is a heuristic. It removes whatever happens to be tight at one particular optimum, so different solver runs gave lists of 458, 464 and 769 types. Each run that ends with a checked bound of $1/2$ proves the theorem for its own list. A systematic search, for example for the smallest list that forces the bound down to $1/2$, has not been done.

4.3 The same with 3×4 windows (result D)

Round	0	1	2	3
3×4 windows	0.514620	0.506944	0.503219	0.499999971
3×3 windows	0.516129	0.510634	0.502250	0.4999999998

This removed 3,350 window types, with a checked bound of 0.500000010. They contain 736 different 3×3 culprit patterns, compared with 458 to 769 in the 3×3 runs. So larger windows improve the bound for all agars, but they do not reduce the number of problem patterns.

4.4 What the optimal LP solution looks like, and how this was checked

An optimal LP solution is a list of windows with frequencies, not an actual pattern. To see what structure it describes, we took the 98 windows used by one optimal solution of the 3×3 LP and asked whether they can be fitted together into an actual square or rectangle.

Method. For an $n \times n$ square, every interior cell position gets one variable per allowed window (98 per position), and every cell of the square gets two variables: alive at time t , alive at time $t+1$. The constraints say:

- each interior position uses at least one of the 98 windows;
- if a position uses a window, the 9 cells around that position have exactly the states that window prescribes, at both times.

Overlapping windows therefore have to agree cell by cell. These constraints were given to the SAT solver CaDiCaL (through PySAT). A satisfying assignment is an actual patch; “unsatisfiable” means no patch of that size exists.

Question	Answer
$n \times n$ squares	8×8 satisfiable; 9×9 and 10×10 unsatisfiable
strips 3, 4 or 5 rows high	satisfiable for every width tested (4 to 30)
strips 6 rows high	satisfiable for widths 4–10; unsatisfiable for widths 14, 20 and 30
strips 7, 8 or 9 rows high	satisfiable for widths 4–8; unsatisfiable for widths 10, 14, 20 and 30

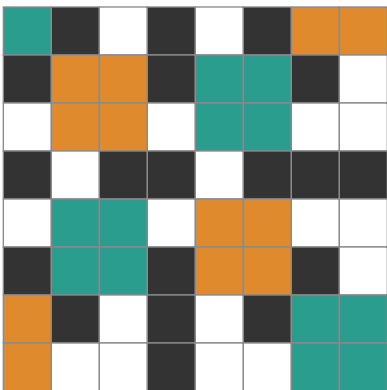


Figure 3. An 8×8 patch built from the 98 windows. Orange: alive at t , dead at $t+1$. Teal: dead at t , alive at $t+1$. Dark: alive at both times. White: dead at both times. No 9×9 patch exists.

Caveats. The check covers the support of one optimal solution; other optimal solutions use other windows. The unsatisfiability results were not accompanied by DRAT proof files, which

solvers such as CaDiCaL can produce and which an independent checker can verify.

A related check used all 6,851 windows that are tight at the optimum. A culprit window can sit at the centre of a 9×9 square of such windows, but not of an 11×11 square. The resulting valid inequality (culprit windows need nearby windows outside the tight set) lowered the LP value only from about 0.516129 to about 0.516119.

4.5 The two families

The strips above describe a band that cannot be stacked. Its two ingredients are the two families of removed windows.

Family 1: the blinking-block band. Two rows of blinking 2×2 blocks, spaced 3 apart and alternating between phases A and B. Between the two rows of blocks is the row .SS.SS of always-alive pairs, and more always-alive cells sit between the blocks. The band is 5 rows high, repeats every 6 columns and has density $8/15$ (Figure 6).

Family 2. A moving double stripe, in which two full rows die while full rows are born next to them, with half-filled rows at its edges. Section 5.5 shows that these edge rows cannot occur.

5. Results proved by hand

5.1 Elkies' theorem, with a discharging proof (result F)

Theorem (Elkies). Let Z be a periodic set of cells in which every cell of Z has at most 3 neighbours in Z . Then Z has density at most $1/2$.

This is a statement about a single configuration; periodicity is only needed for density to make sense. It is tight: stripes have density $1/2$. It is due to Elkies (1998). Our proof below uses discharging; its rules resemble the area rules of Chu, Stuckey and de la Banda (2009), used there for the maximum-density still-life problem.

Proof. Call the cells of Z live. Give each live cell charge $+1/2$ and each dead cell $-1/2$; in a repeating block of n cells with L live ones the total is $L - n/2$, so we must show the total is at most 0. For a cell y and a direction, the two **side cells** of y are its neighbours on either side, perpendicular to that direction. Charge moves by two rules (Figure 4):

- **Rule 1.** Each live cell sends $1/4$ to each dead cell directly above, below, left or right of it, unless both side cells of that dead cell are live.
- **Rule 2.** A live cell x sends $1/4$ to a live neighbour w directly above, below, left or right of it, if both side cells of x are live and both side cells of w are dead.

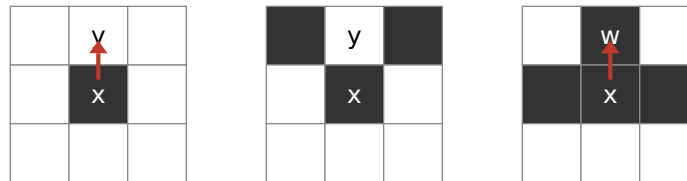


Figure 4. Left: Rule 1, x sends $1/4$ to the dead cell y . Middle: no transfer, because both side cells of y are live. Right: Rule 2, x sends $1/4$ to w .

Charge is conserved, so it is enough to check that every cell ends with charge at most 0.

- **Dead cell.** It only receives, by Rule 1. With 3 live direct neighbours, say N, W, E, the transfer from N is blocked (W and E are live) and it receives $1/2$. With 4, all transfers are blocked. With 2 or fewer it receives at most $1/2$.
- **Live cell x , no live direct neighbours.** At most 3 live diagonal neighbours can block at most 2 of the 4 transfers, so x sends at least $1/2$.
- **One live direct neighbour, say W.** At most 2 live diagonals block at most one of the transfers to N, E, S, so x sends at least $1/2$. If W sends $1/4$ to x by Rule 2, then NW and SW are live, so NE and SE are dead and x sends $3/4$.
- **Two live direct neighbours.** At most one diagonal is live, nothing is blocked, and x sends $1/2$.

- **Three, say N, W, E.** x has no other live neighbours; it sends $1/4$ to S and, by Rule 2, $1/4$ to N.

Every cell ends at most 0, so the total is at most 0. ■

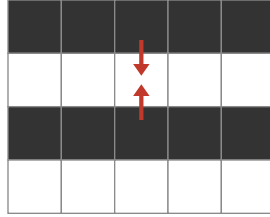


Figure 5. Stripes: every live cell sends $1/4$ up and $1/4$ down, every dead cell receives $1/2$, and every cell ends at 0.

Remarks. (i) The dead-cell case never uses degrees, so in any configuration only overcrowded live cells can end with positive charge. (ii) Applied to each phase, the theorem gives density at most $1/2$ for any oscillator in which no cell ever dies of overcrowding. Both are immediate, but they explain why the LP analysis concentrates on overcrowding.

5.2 Period 2 without adjacent rotors of the same type (result G)

Claim. If no two X cells are adjacent and no two Y cells are adjacent, a period-2 agar has density at most $1/2$.

Proof. An X cell is dead in phase B and alive in phase A, so it is born: in phase B it has exactly 3 live neighbours, all of type S or Y. In particular it has at most 3 S neighbours. In phase A its live neighbours are S or X cells; with no X neighbours it has at most 3, so it is not overcrowded. The same holds for Y cells, and S cells survive both steps, so they have at most 3 live neighbours. No cell is ever overcrowded, and Remark (ii) applies. ■

The hypothesis is strong, so the claim is narrow. It is included because the same counting (X cells have exactly 3 live neighbours of type S or Y in phase B) is used in Section 5.4.

5.3 A bound from the average degree (result H)

Claim. If a periodic set has density α and its cells have k neighbours in the set on average, then $\alpha \leq (k + 4)/12$.

Proof. Work on a torus with n cells, let A be its adjacency matrix (8 neighbours per cell) and let z be the indicator vector of the set, which has αn cells. Then:

- $z^T A z = k \alpha n$ (each cell of the set has k neighbours in it on average), $1^T A z = 8 \alpha n$ and $1^T A 1 = 8n$.
- The eigenvalues of A are $(1 + 2\cos \theta)(1 + 2\cos \phi) - 1$ for the torus frequencies θ, ϕ . Each factor $1 + 2\cos$ lies in $[-1, 3]$, so every eigenvalue is at least -4 .

- Let $y = z - \alpha 1$. Then $y^T A y = k\alpha n - 16\alpha^2 n + 8\alpha^2 n = (k - 8\alpha)\alpha n$, and $|y|^2 = \alpha n - \alpha^2 n = \alpha(1 - \alpha)n$.
- Since $y^T A y \geq -4|y|^2$, we get $k - 8\alpha \geq -4(1 - \alpha)$, that is, $12\alpha \leq k + 4$. ■

The bound is tight for stripes ($k = 2$, $\alpha = 1/2$) and for the full plane ($k = 8$, $\alpha = 1$). We also checked it on 20,000 random tori with no violation. It is a standard eigenvalue argument of Hoffman type, and it does not help directly with oscillators: overcrowded phases have k well above 2.

5.4 The band-forcing lemma (result 1, period 2)

Write each cell of the row just above the band as a pair (a_j, b_j) : $a_j = 1$ if it is alive in phase A and $b_j = 1$ if it is alive in phase B, with columns $j = 0, \dots, 5$ counted modulo 6. Each cell in the top row of the band (XXSYYS) gives one condition on the three cells above it. X cells are born into phase A, so they need exactly 3 live neighbours in phase B. Y cells need exactly 3 in phase A. S cells must survive, so they have at most 3 in each phase. Counting the neighbours inside the band gives:

```
cell (col 0, X):  b5 + b0 + b1 = 2      cell (col 1, X):  b0 + b1 + b2 = 2
cell (col 2, S):  a1 + a2 + a3 <= 1    and  b1 + b2 + b3 <= 1
cell (col 3, Y):  a2 + a3 + a4 = 2      cell (col 4, Y):  a3 + a4 + a5 = 2
cell (col 5, S):  a4 + a5 + a0 <= 1    and  b4 + b5 + b0 <= 1
```

Solving. The two Y equations give $a_2 = a_5$. If $a_2 = a_3 = 1$ the first S condition fails, so exactly one of them is 1 and $a_4 = 1$. The second S condition then forces $a_5 = a_0 = 0$, hence $a_2 = 0$, $a_3 = 1$ and $a_1 = 0$. The same reasoning with the X equations gives $b = (1, 1, 0, 0, 0, 0)$. So the row above must be **YY.XX.**, and by the symmetry of the band (swap phases and reflect vertically) the row below must be **XX.YY.** (Figure 6).

This argument assumes that the row above repeats every 6 columns. A SAT check over a 30-column stretch of band, without that assumption, found exactly one possibility for every cell above columns 6 to 24; only cells near the ends of the stretch are free.

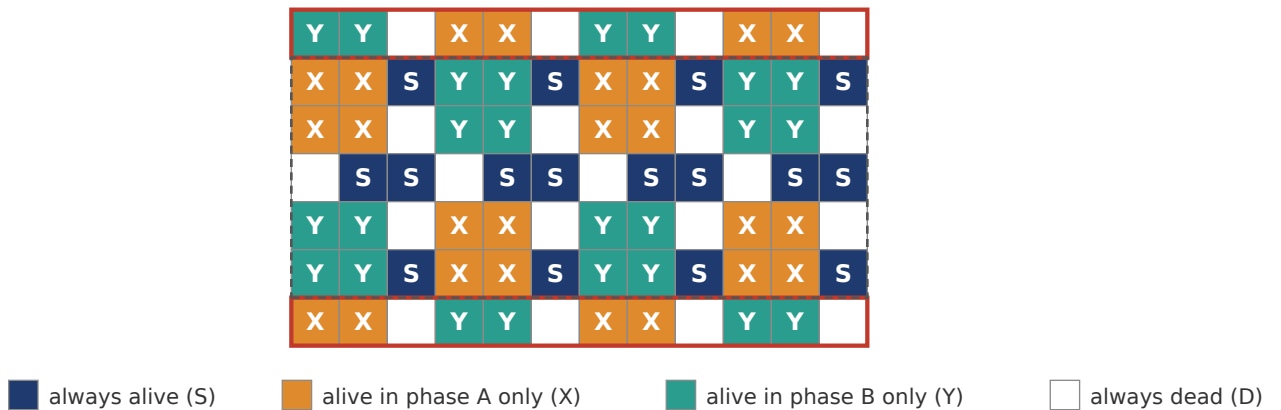


Figure 6. The band (dashed, two periods) and the rows it forces above and below (red).

Per 6 columns	Live cells (average over both phases)	Compared with 1/2
band (5 rows)	16 of 30	+1
forced row above	2 of 6	−1
forced row below	2 of 6	−1

So the band with its two forced rows has density $20/42 < 1/2$. A real period-2 agar on a 6×14 torus consists of two such units and has density exactly $10/21$. RLE for Golly: $x = 6, y = 14$, rule = B3/S23:T6,14 followed by
3o2bo\$2o\$b2ob2o\$3b2o\$2b4o\$2o\$2o\$2b4o\$3b2o\$2ob2o\$2o\$3o2bo\$3b2o\$3b2o!

Further checks (exact SAT results).

Question	Answer
Real period-2 agars 6 columns wide that contain the band	none at heights 7, 8, 10, 12 and 15–18; densities 0.476, 0.456, 0.467 at heights 14, 19, 20 (heights 9, 11, 13 not tested)
A piece of band m columns long: largest excess of live cell-phases over half, in the band plus one row on each side	+8 for $m = 6$, −1 for $m = 12$, −3 for $m = 18$

The last row suggests that long pieces of band are always below $1/2$, with a bounded contribution from their ends. We have only checked $m = 6, 12$ and 18 .

5.5 The edge rows of family 2 (result J)

Suppose the strip repeats every 2 columns, so the row above an edge row alternates two values a and b (Figure 7). An edge cell that must be born has no other live neighbours at time t , so it needs $a + b + a = 3$, which forces $a = b = 1$. Its neighbour in the edge row, which must stay dead, then also has 3 live neighbours and would be born. This is a contradiction. What remains is the ordinary moving double stripe, which has density exactly $1/2$ (Figure 8).

a	b	a	b	a	b
	x				

Figure 7. Top: the row above, alternating a, b . Bottom: cells that must be born (teal) and a cell that must stay dead ($\times$). Both see the same number of live cells above.

X	X	X	X
X	X	X	X
Y	Y	Y	Y
Y	Y	Y	Y

Figure 8. The moving double stripe on a 4-row torus: the X rows are alive in phase A, the Y rows in phase B. Every live cell has 5 live neighbours and dies of overcrowding; density is exactly 1/2.

6. Agars with a small spatial period (result E)

Suppose an agar repeats every h cells in one direction; after rotating, say it repeats every h columns. Then it can be read as a sequence of rows, each row being one of finitely many states (h cells in each of p phases). Its density is at most $1/2$ exactly when a graph on pairs of consecutive rows has no cycle of positive weight, where an edge (a, b, c) has weight $2 \cdot (\text{live cell-phases of } b) - h \cdot p$. When there is no such cycle, the Bellman-Ford algorithm produces integer labels g with $g(a, b) \geq \text{weight} + g(b, c)$ on every edge, which is a checkable certificate.

Period p	1	2	3	4	5	6
Values of h checked	≤ 6	≤ 5	≤ 3	≤ 2	≤ 2	1

The script `strip_certificate.py` does this for a given h (its `--width` option) and period; it also covers every h dividing the given one and every period dividing p . It reproduces the period-2 results for $h = 4$ and 5 , and correctly finds a counterexample (the full plane) for the rule B3/S012345678, in which live cells never die.

7. Approaches that did not lead anywhere

Approach	Result	Reason
Charge rules or LPs using fixed-size windows	≈ 0.5161 (3×3), ≈ 0.5146 (3×4)	the LP optimum is locally consistent up to 8×8 (Section 4.4)
Larger windows only around dying 2×2 triples (4×4)	at least 0.5149	the problem structures are larger than 4×4
Requiring windows to have a long history	no change	all 98 windows of the optimal solution can arise after 4 generations; deeper histories were not checked
Choosing one phase per cell (slicing)	fails	a real agar contains three adjacent rotor cells that cannot be separated
Moving excess charge one generation forward	fails	fails on about 12% of 5×5 patterns and on real agars
Sharing charge with nearby cells (radius 1)	fails	a CP-SAT search found real counterexamples
Local surgery on real oscillators	fails	the moving double stripe cannot be changed locally
Matching S cells to D cells along lines	fails	7 of 17 random period-2 agars have no suitable direction
Local counting around one blinking block	fails	an isolated block's ring can have 4 more S than D cells, and nothing nearby is forced

Approach	Result	Reason
Splicing identical rows of a smallest counterexample	no bound	only bounds each side in terms of the other
Reducing to a bounded period	no proof	cuts must match across the whole width

8. Reproducing the results

Setup

```
conda create -n agar python=3.12 -y
conda activate agar
pip install numpy scipy highspy python-sat python-flint
```

Scripts

Script	What it does
bound_16_31.py	short standalone computation of the 3×3 bound (result B)
life_agar_density.py	3×3 LP, removal rounds, exact certificate (result C)
big_window_lp.py	LP for any window size: solve, removal rounds, checked bounds, symmetry (results A, B, D)
check_windows.py	reports where the excluded windows occur in a given oscillator
band_forcing_check.py	SAT check of the band-forcing lemma
strip_certificate.py	exact results for agars with a fixed spatial period (result E)

Exclusion theorem (about 5 minutes on one core)

```
python life_agar_density.py certify

reachable windows: 87851
peel round 0: LP bound 0.516129; tight culprit windows 405
peel round 1: LP bound 0.505279; tight culprit windows 364
peel round 2: LP bound 0.500000; tight culprit windows 248
excluded 769 window types (saved to excluded_windows.pkl)
basis valid=True; exact system 6764 x 6764
EXACT BOUND = 1/2 (certificate saved to certificate.pkl)
```

Bounds for all agars

The 3×3 run takes about 1 minute. The 3×4 runs were done on a 16-thread machine, about 10 minutes per solve.

```
python big_window_lp.py certify --h 3 --w 3
RIGOROUS bound: density <= 51612905/100000000 = 0.516129050

python big_window_lp.py solve --h 3 --w 4 --sym --threads 16
RIGOROUS bound: density <= 205847955/400000000 = 0.514619887

python big_window_lp.py peel --h 3 --w 4 --sym --threads 16
peel round 3: bound 0.499999971 ... excluded 3350 window types
RIGOROUS bound: density <= 200000004/400000000 = 0.500000010
```

Other checks

- `python band_forcing_check.py`: every cell above columns 6 to 24 of a 30-column band has exactly one possible type, repeating YY.XX.
- `python strip_certificate.py --width 5 --period 2` ends with PROVEN in under a minute.
- The triangle agar used in Section 7 (8×10 torus, period 2), phase A: $x = 10$, $y = 8$, rule = B3/S23:T10,8 b2o\$5bob3o\$4ob2obo\$3b2o\$2b2ob5o\$obobo2bo\$obobo\$o2bo5bo!

9. Open problems

- **The excluded windows.** By the exclusion theorem, the conjecture is equivalent to: in every oscillator, the 769 listed window types never push the density above $1/2$.
- **Arrangements.** Long pieces of band appear to be below $1/2$ once their forced rows are counted, but a single blinking block forces nothing nearby. What prevents short or irregular pieces from being packed densely?
- **A systematic exclusion search.** Find the smallest, or the most structured, set of windows whose removal forces the bound to $1/2$, instead of the greedy rounds used here; repeat with larger windows.
- **Verification.** Re-implement the window enumeration, the constraints and the certificate checks independently, and produce DRAT proofs for the SAT unsatisfiability results in Section 4.4.
- **Bounded period.** If a counterexample exists, must there be one with a bounded period in space or in time? If so, computations like those in Section 6 would finish the proof.