Equation Automata

Cellular Automaton Outputs a Trigonometry Equation

by

GOKTUG ISLAMOGLU

goktug.islamoglu@cibss.uni-freiburg.de

GOKTUG ISLAMOGLU from Freiburg, Germany

CIVIL ENGINEER / OHS EXPERT

B.Sc Istanbul Technical University

M.Sc Karadeniz Technical University

Research Data Manager at CIBSS

INFORMS 2018 Session Chair

ITS SUMMIT 2019 Presentation

NECSI ICCS 2020 Presentation

NERCSS 2021 Presentation

Model: Equation Automaton[1]

Runs upon: PyCX 0.3 Realtime Visualization Template

PyCX 0.3 Realtime Visualization Template: Written by Chun Wong

Revised by Hiroki Sayama

Requires PyCX simulator to run

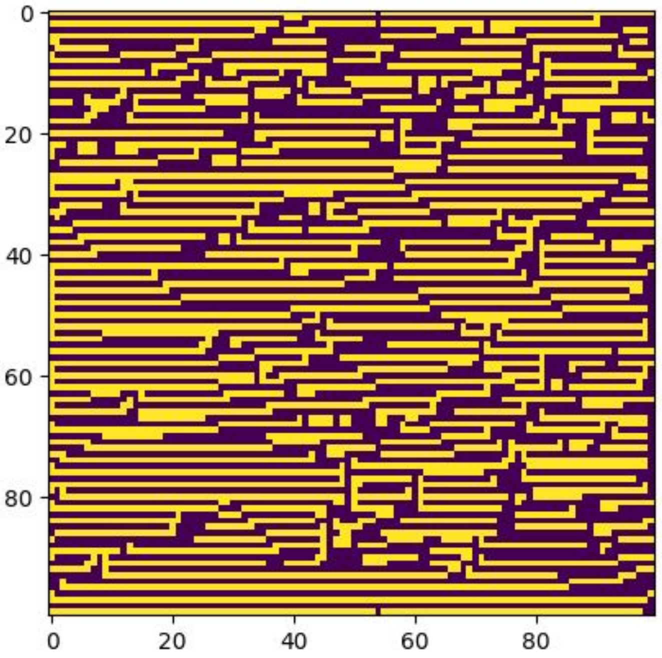
PyCX available from: https://github.com/hsayama/PyCX

1. https://github.com/goektug/Equation-Automata/

Idea

Critical behavior by extending cells of an automaton uniaxially

This is named the “driving motion”, aimed at maximizing total cell count



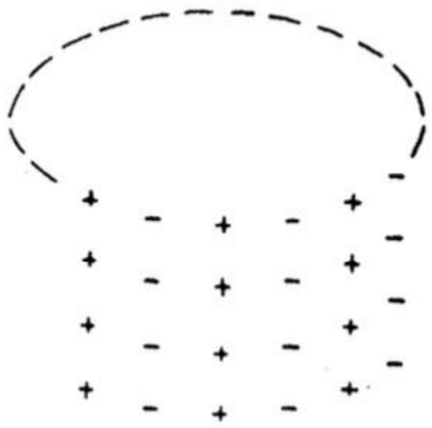
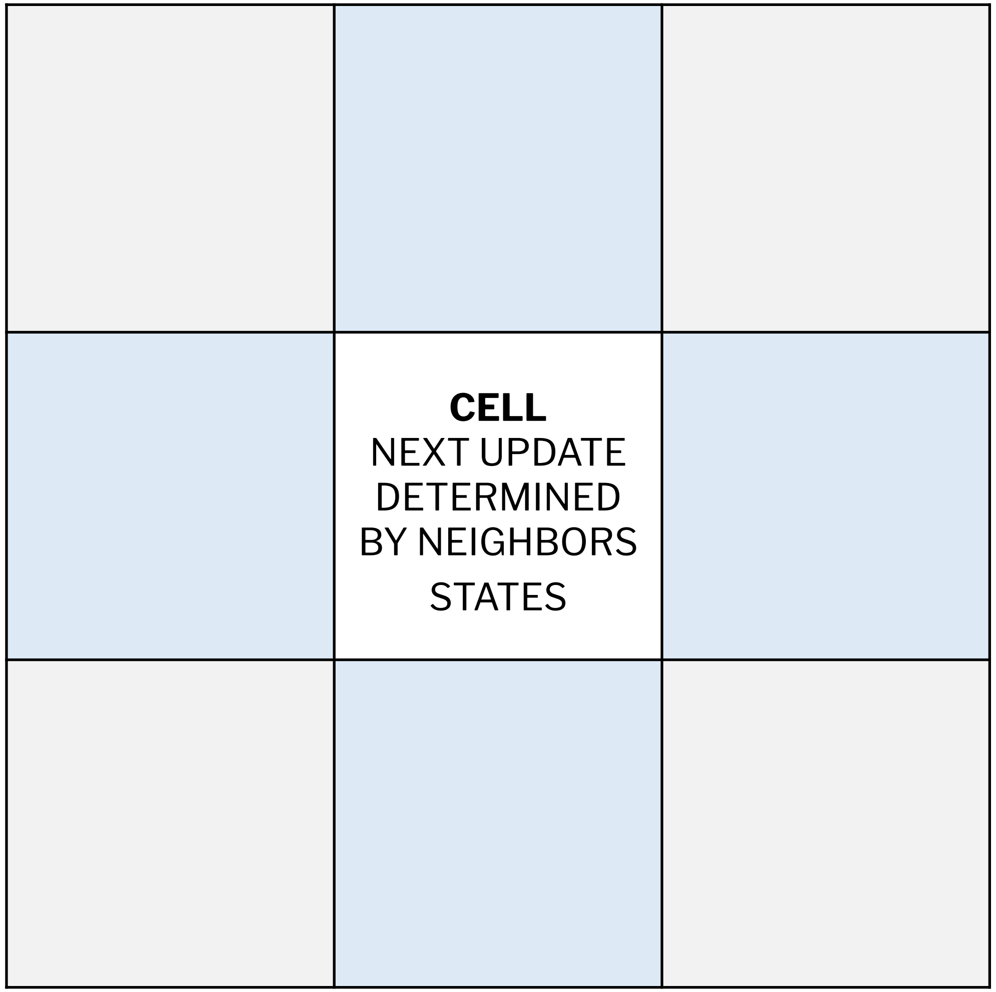


FIG. 8. Superlattice at low temperatures in crystal for which the interaction energy between neighbors of opposite spin is $J > 0$ lengthwise and $-J' < 0$ transversely.

Figure: Onsager, Lars. Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition. Phys. Rev. 65, 117 – Published 1 February, 1944, doi:https://doi.org/10.1103/PhysRev.65.117

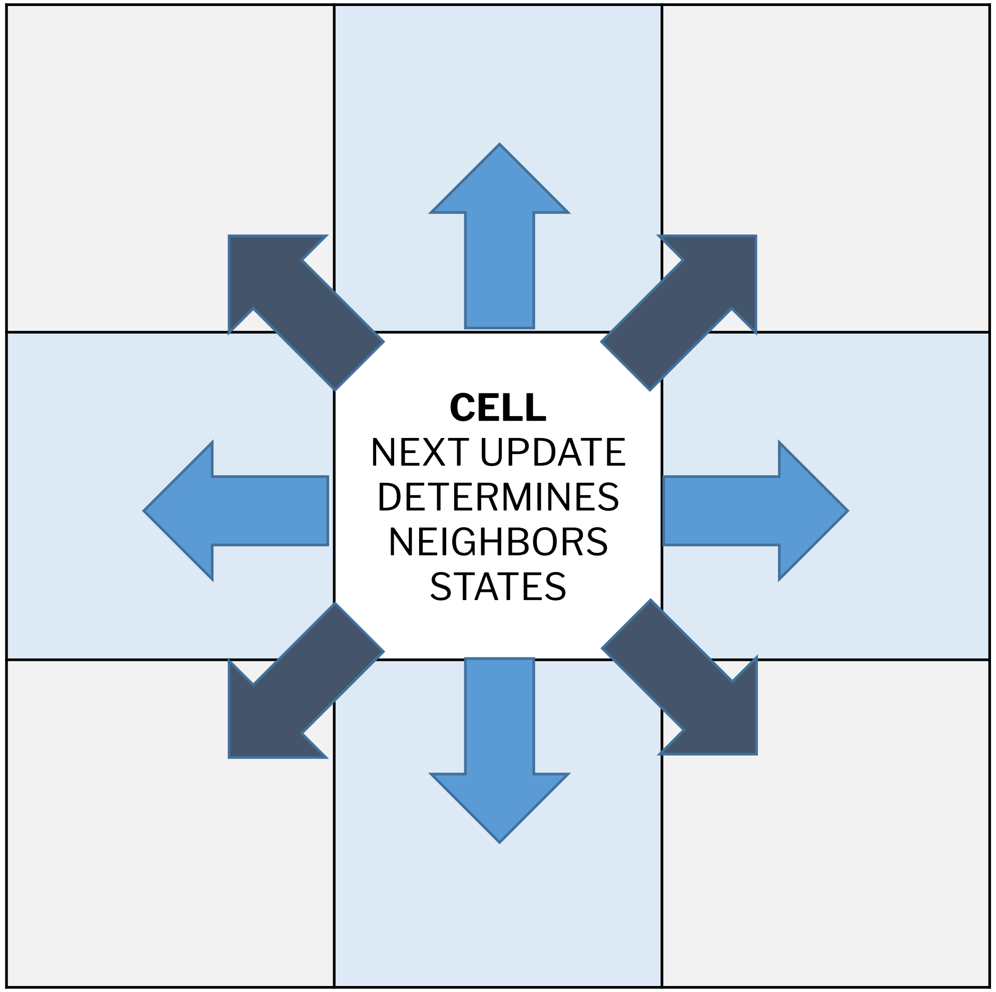
Conventional Cellular Automaton Update



A cellular automaton is a search function around a cell.

Update rule is the determination of a cell’s state based on the states of its neighboring cells following a rule.

1st Rule: Inverse Cell Update

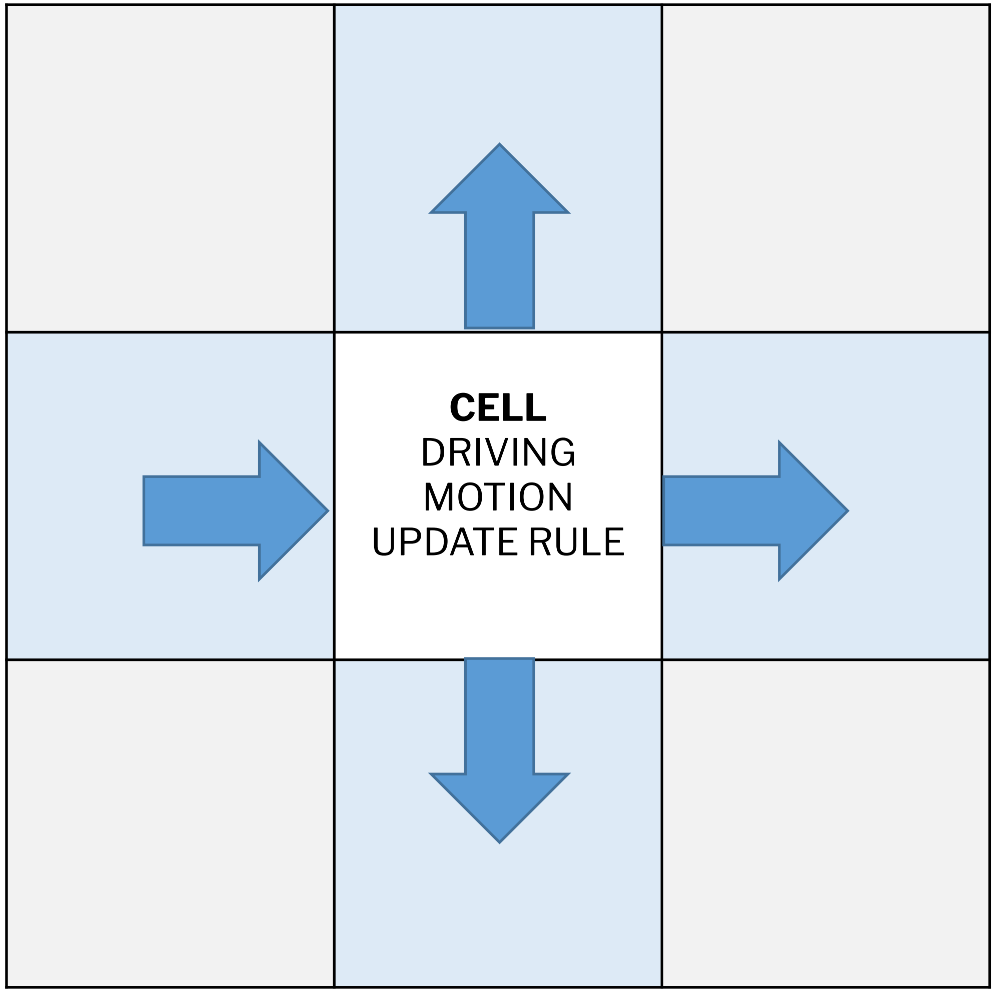


To achieve driving motion, update of a cell is reversed. Cell's state dictates its neighborhood.

```
elif c[x, y] == 1:
    array1.append(c[x, y])
    for z in range(-1, 2):
        # block generation from randomly distributed points

        #neighbor updating from cell(x,y)
        m = number_of_upper_neighbors(x, y)
        if m == 1:
            nc[x, (y + 1) % L] = 1
```

1st Rule: Inverse Cell Update

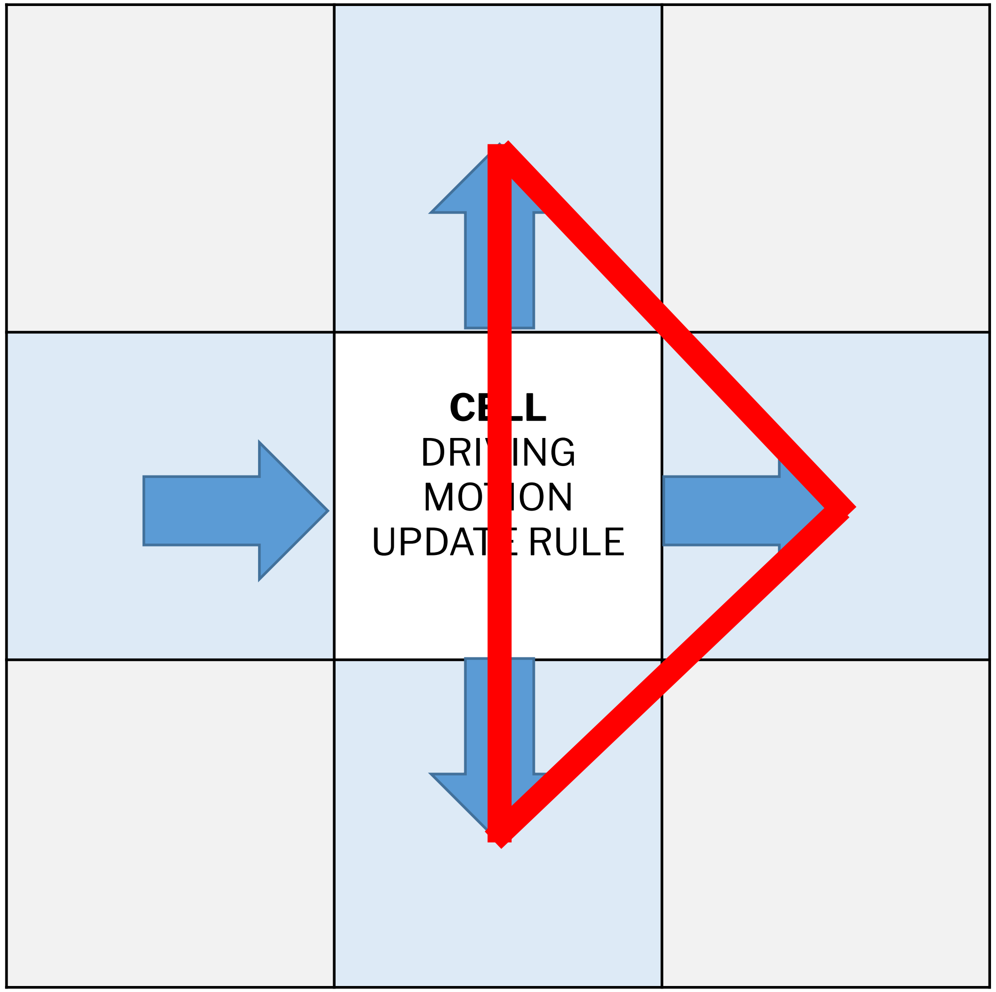


To achieve driving motion, update of a cell is reversed. Orthogonal neighbors are either updated to state “1” (arrow out), or updated to state “0” (arrow in)

```
elif c[x, y] == 1:
    array1.append(c[x, y])
    for z in range(-1, 2):
        # block generation from randomly distributed points

        #neighbor updating from cell(x,y)
        m = number_of_upper_neighbors(x, y)
        if m == 1:
            nc[x, (y + 1) % L] = 1
```

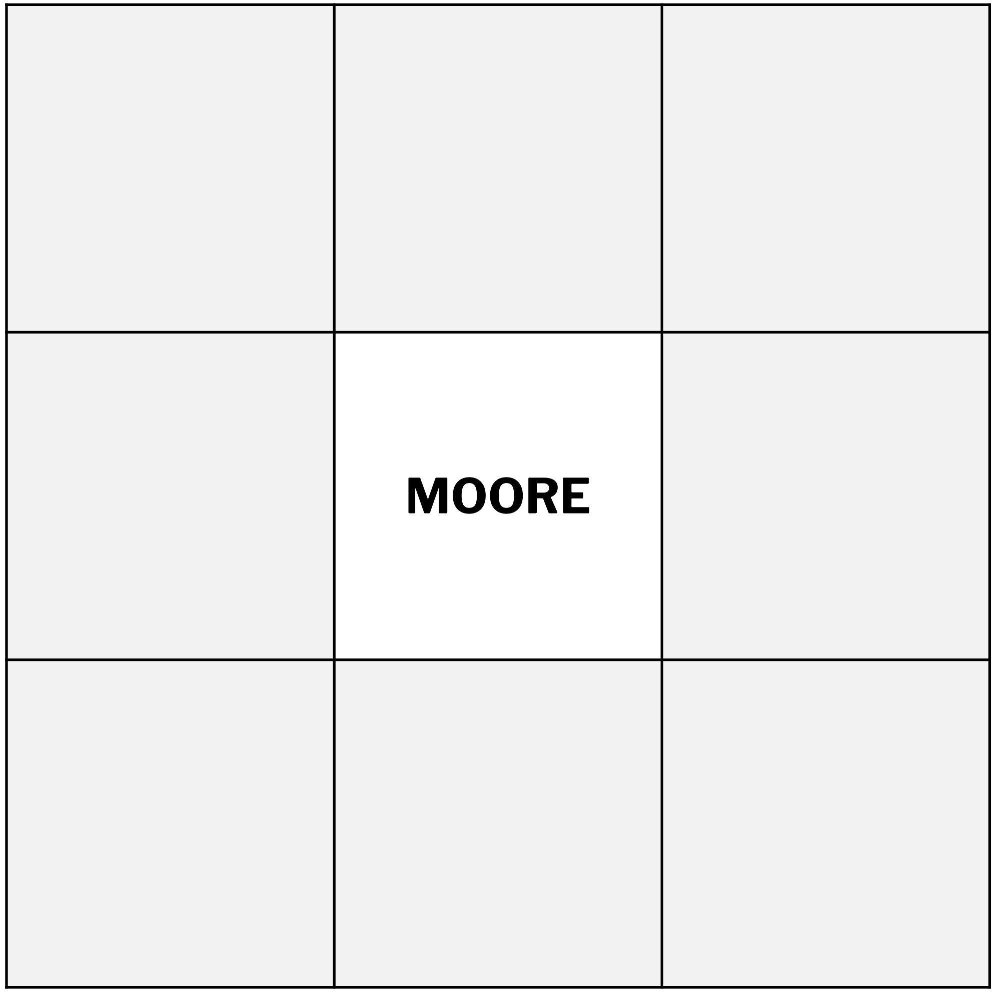
Hypothesis: Right-Angle Wavefront

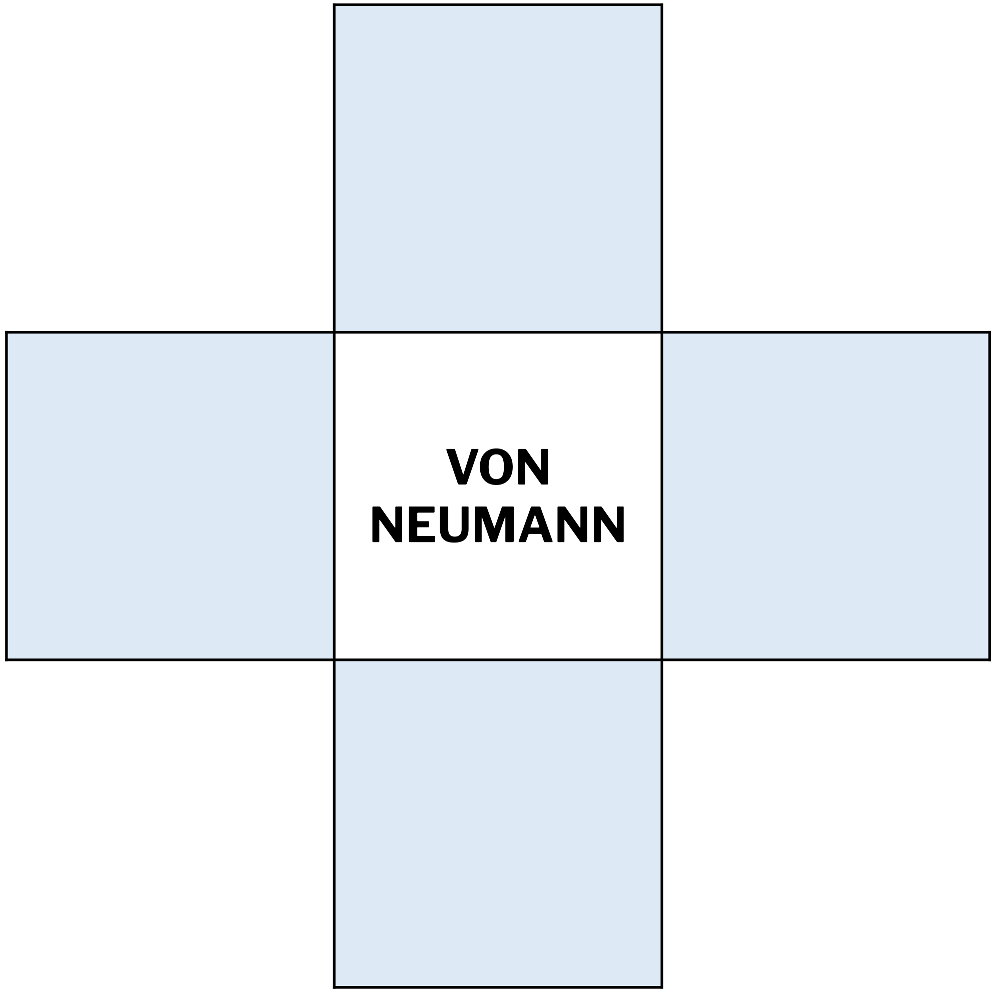


With the removal of the left-hand cell, the remaining cells form a "right-angle wavefront".

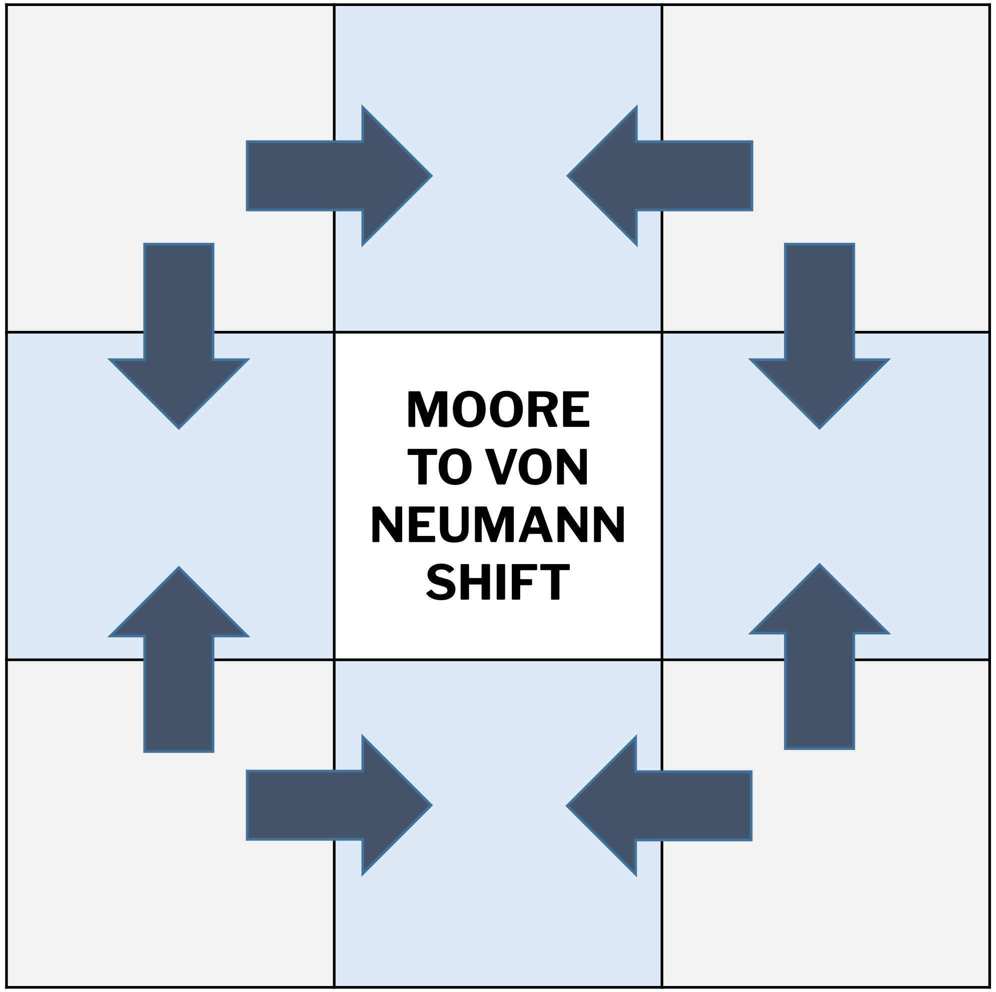
This wavefront is hypothesized to create the $\sqrt{2}$ constant that will appear often in the model.

Moore vs Von Neumann Neighborhoods





2nd Rule: Moore to Von Neumann Shift



To achieve criticality, Moore neighborhoods and and von Neumann neighborhoods are competed.

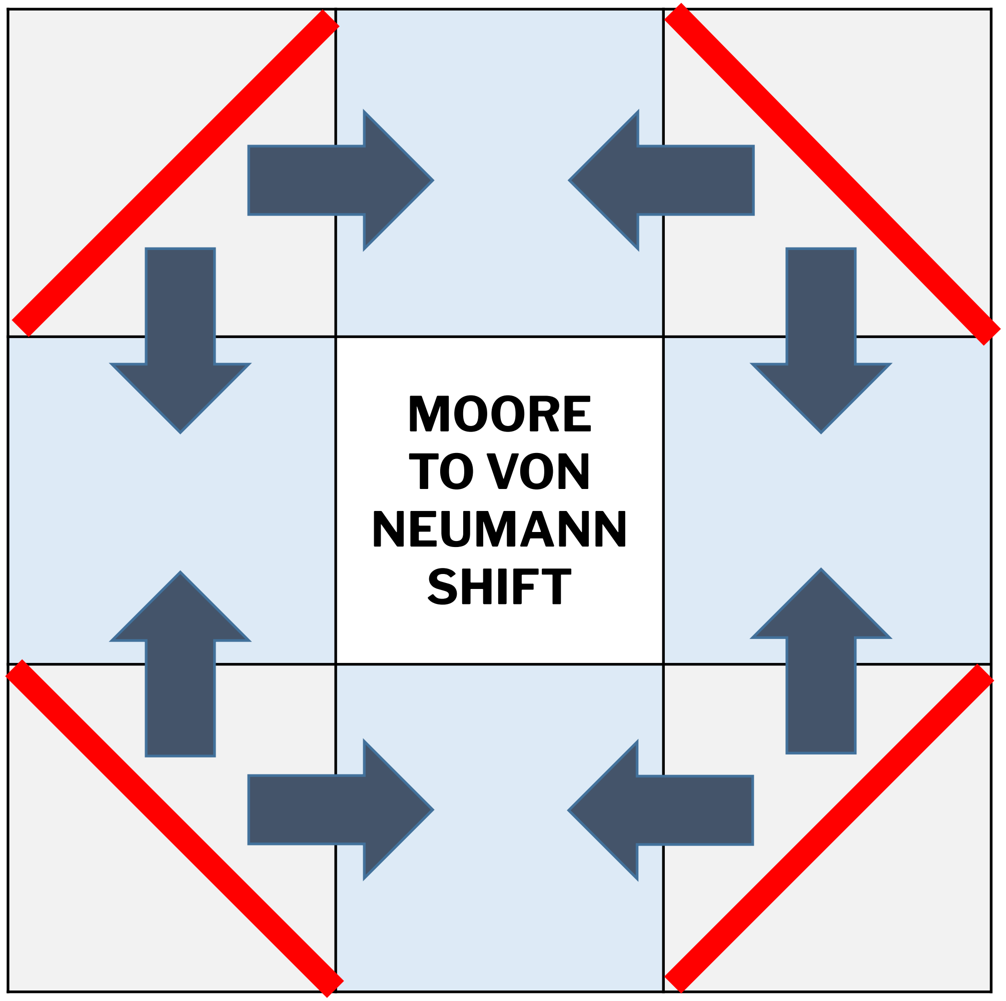
```
#neighbor updating from cell(x,y)
m = number_of_upper_neighbors(x, y)
if m == 1:
    nc[x, (y + 1) % L] = 1

n = number_of_lower_neighbors(x, y)
if n == 1:
    nc[x, (y - 1) % L] = 1
```

```
k = number_of_right_neighbors(x, y)
if k == 0 and (m <= 1 or n <= 1):
    nc[(x + 1) % L, (y + z) % L] = 1

l = number_of_left_neighbors(x, y)
if l == 1 and (m > 1 or n > 1):
    nc[(x - 1) % L, (y + z) % L] = 0
```

Neighborhood coupling



The same right-angle relation is observed in the Moore-von Neumann neighborhood competitions, allowing a coupling equation to be written.

This coupling only occurs in a single cell, as opposed to the wavefront in the previous example.

3rd Rule: Coupled Cellular Automata

$$\rho(t+1) = (1-p)\varphi\big(\rho(t)\big)$$

Left: Stochastic coupling mechanism evolution equation.[2]

The critical roots are realized when 1 is taken as the Moore neighbor count.

1/8: Inverse Moore Neighborhood

$$p(1-p) = \frac{1}{8}$$

$$p^2 - p + \frac{1}{8} = 0$$

$$p = \frac{1}{2} \pm \frac{1}{2\sqrt{2}}$$

```
if g / 8 > (1 - p) * p:  # coupling function
    nc[(x + 1) % L, y] = 1
elif g / 8 < (1 - p) * p:
    nc[(x - 1) % L, y] = 1
else:
    nc[x, y] = 1
```

2. Synchronization of stochastically coupled cellular automata. Luis G. Morelli and Damian H. Zanette. Phys. Rev. E 58, R8(R) – Published 1 July 1988

4th Rule: Neighborhood Configuration

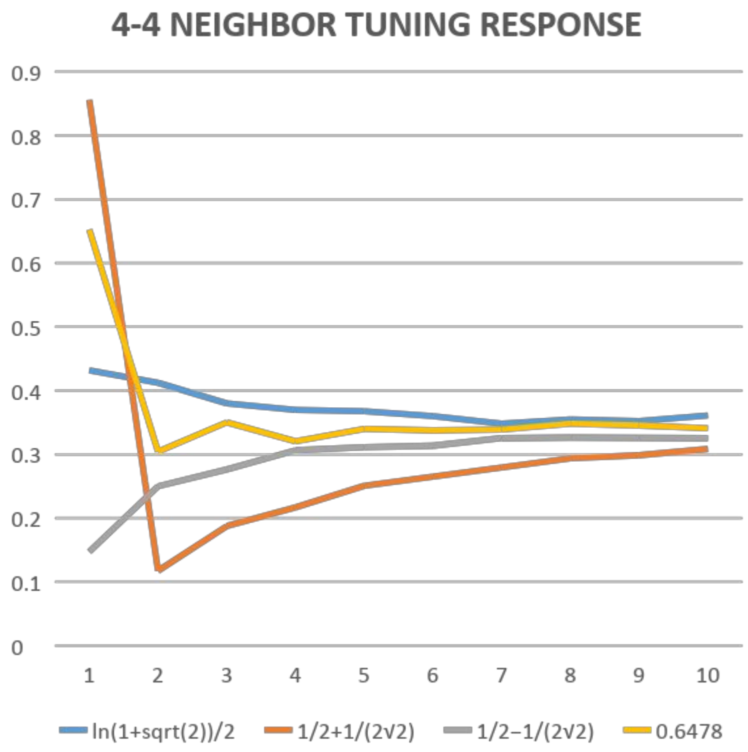
1	**1**	**1**
0	**TUNING THE NUMBER OF NEIGHBORS WITH STATE 1**	**0**
1	**1**	**1**

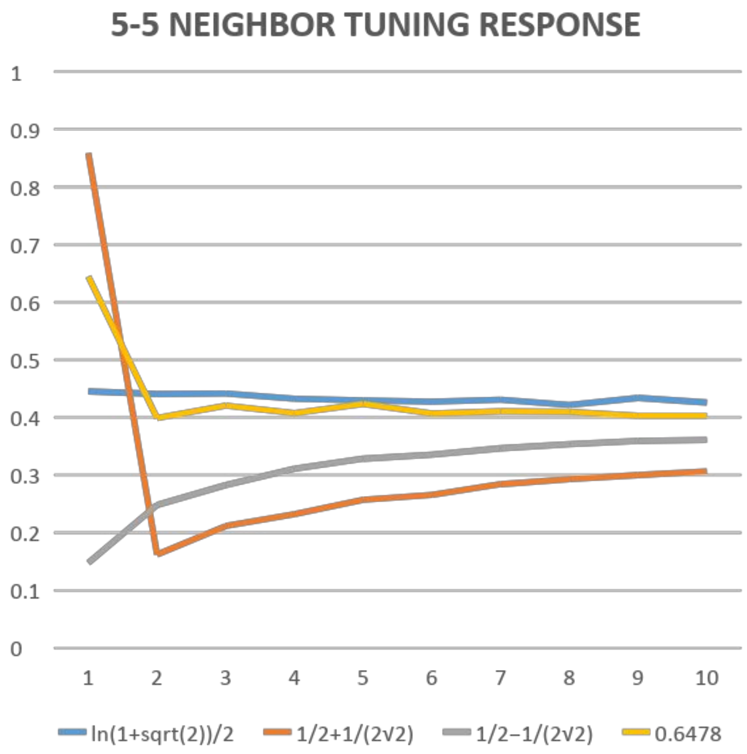
Based on cell's value, Moore and von Neumann neighborhoods are tuned. Critical values are set to 6.

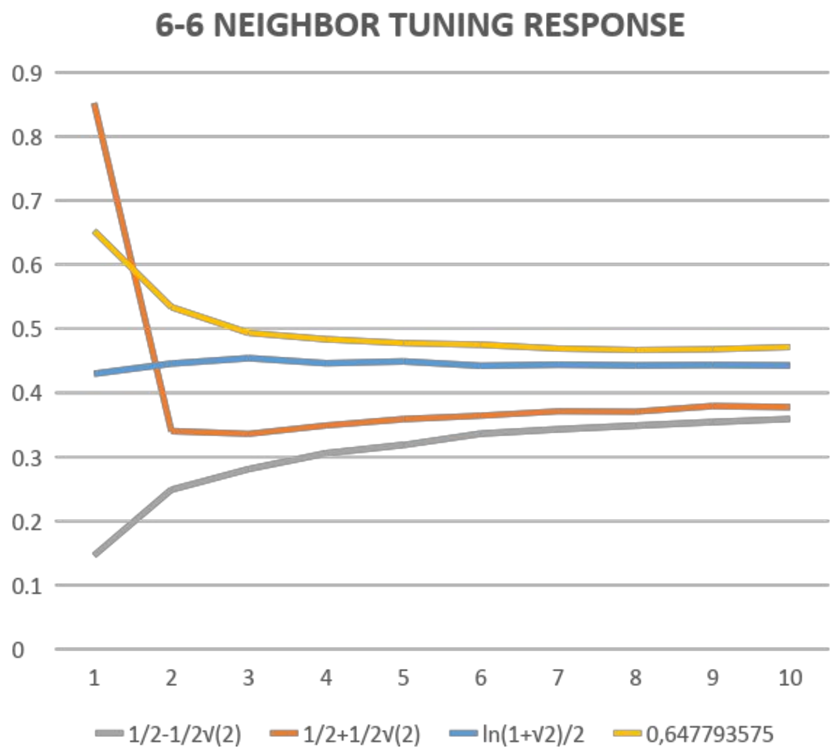
```
g = number_of_Moore_neighbors(x, y) #CA tuning
if c[x, y] == 0:
    nc[x, y] = 0 if g <= 6 else 1
    array0.append(c[x, y])

h = number_of_Neumann_neighbors(x, y) #CA tuning
if h >= 1:
    nc[x, y] = 1 if g <= 6 else 0
```

Neighbor Tuning for Ising Criticality



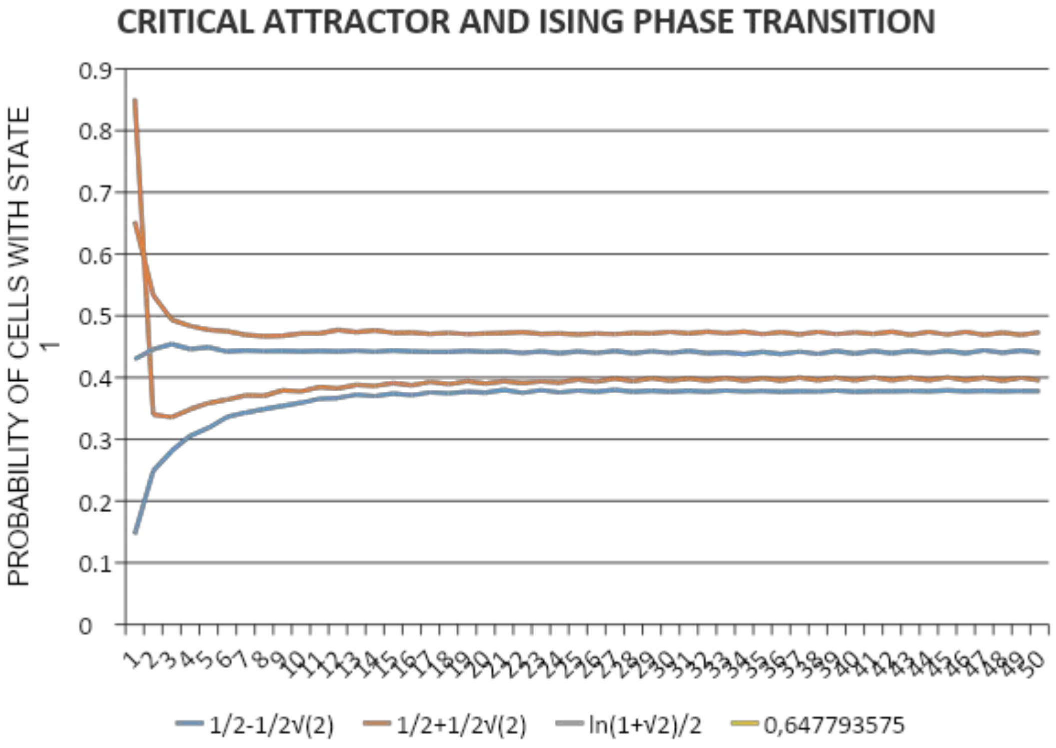


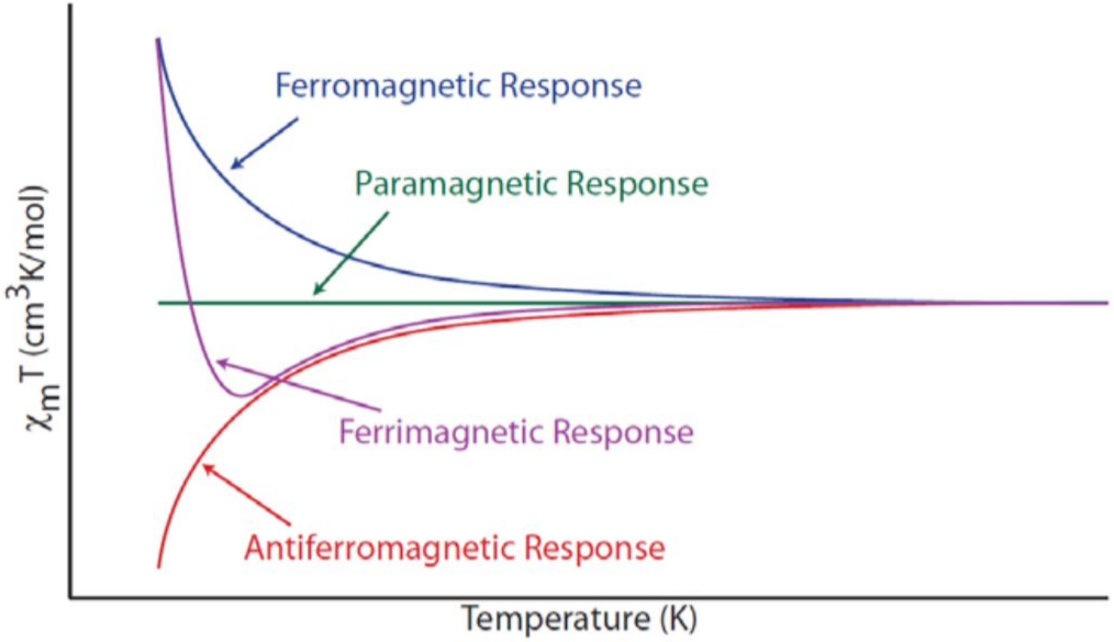


Tuning for **4** and **5** neighbors has **Inverse Ising critical temperature** as the highest state **1** cell count.

For **6** neighbors however, there is another maximum count, corresponding to ferromagnetism.

Coupling Function – Magnetizing Automaton





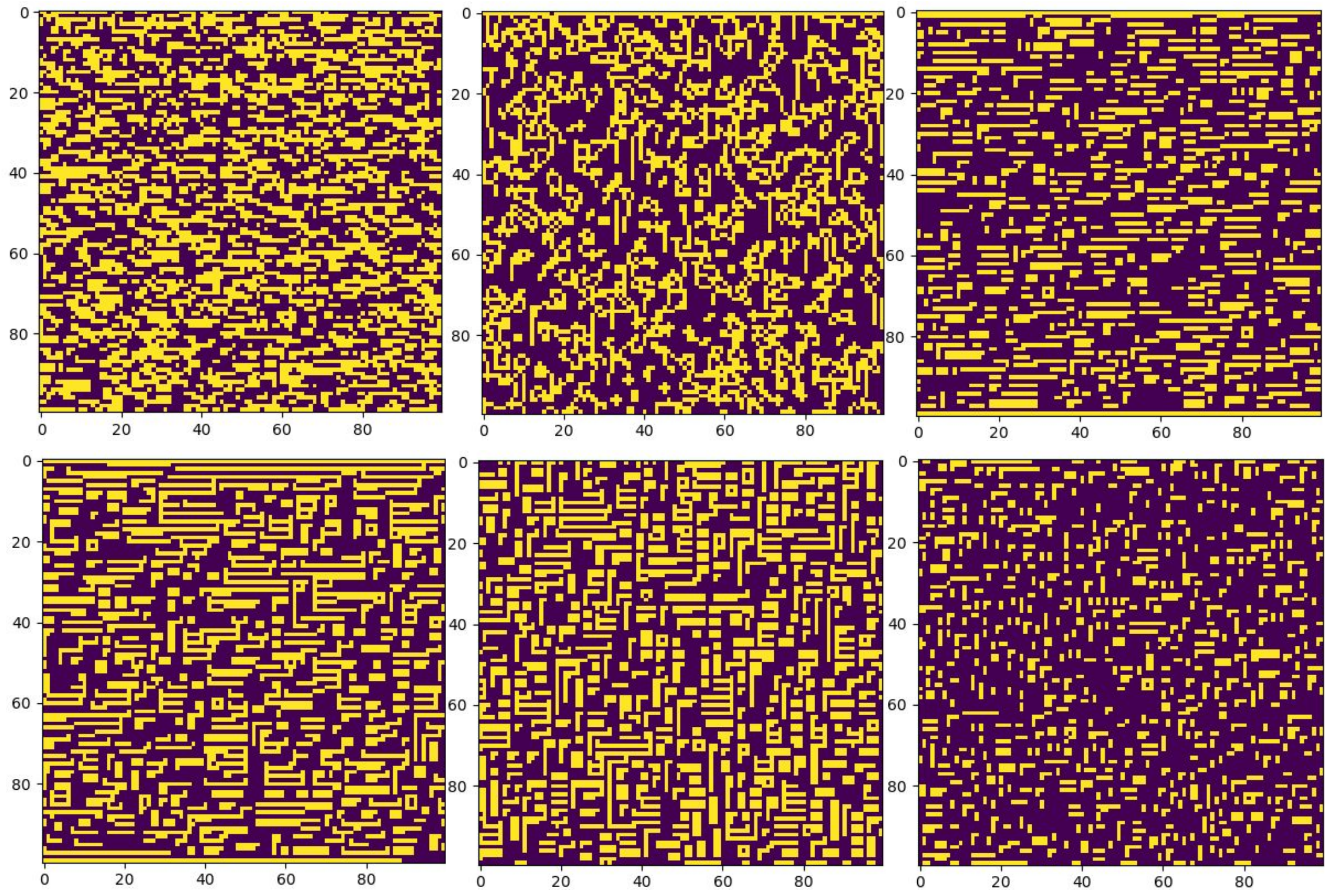
2D Ising square lattice model's critical inverse temperature is the **paramagnetic response.**

Upper coupling is slightly striped, which is weak ferromagnetism (**ferrimagnetism**).

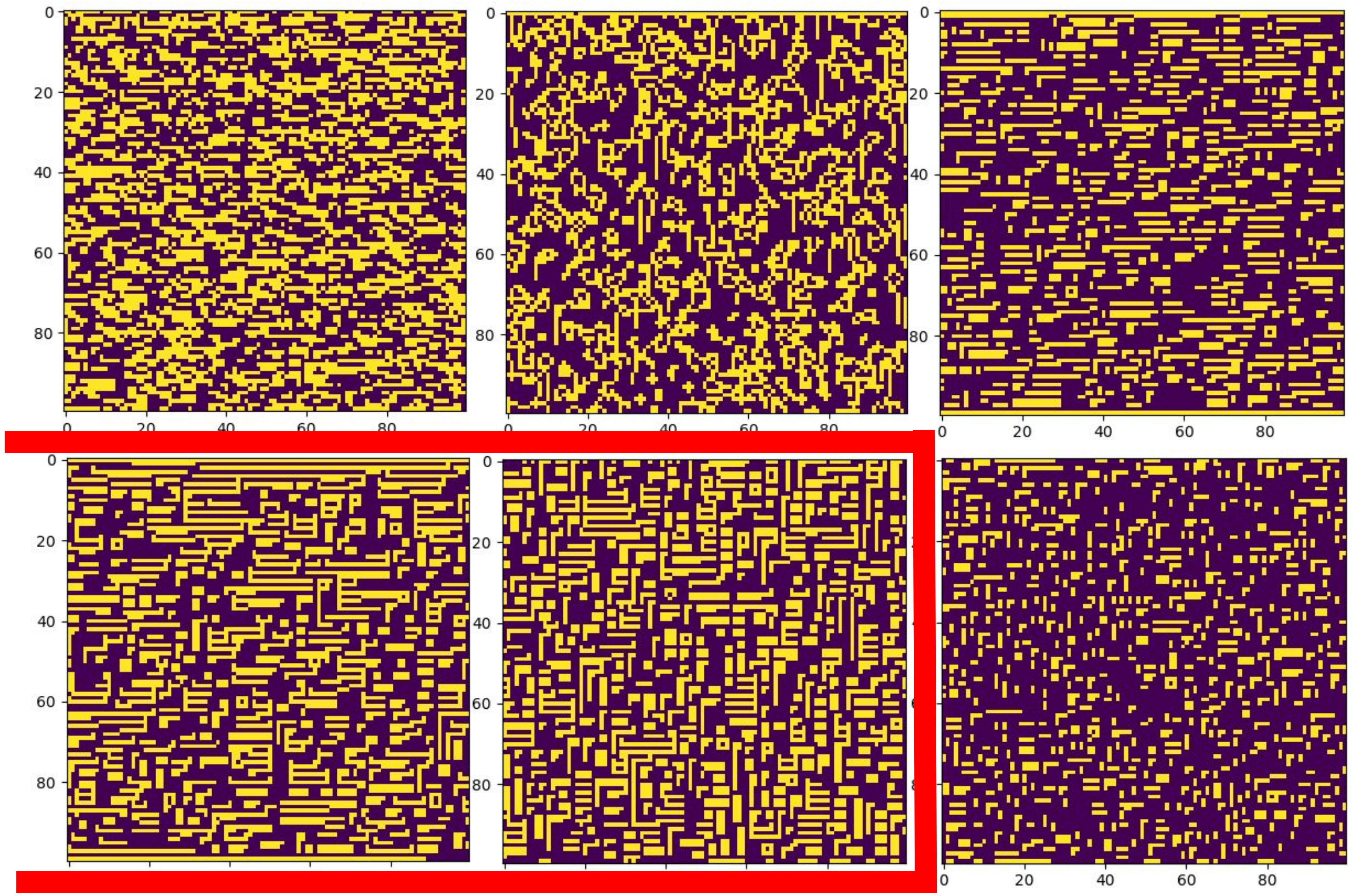
Lower coupling is vortex shaped, which is **antiferromagnetic** behavior.

Right Figure: http://bh.knu.ac.kr/~leehi/index.files/MPMS_HIL.pdf

Evolution of Coupled CAs



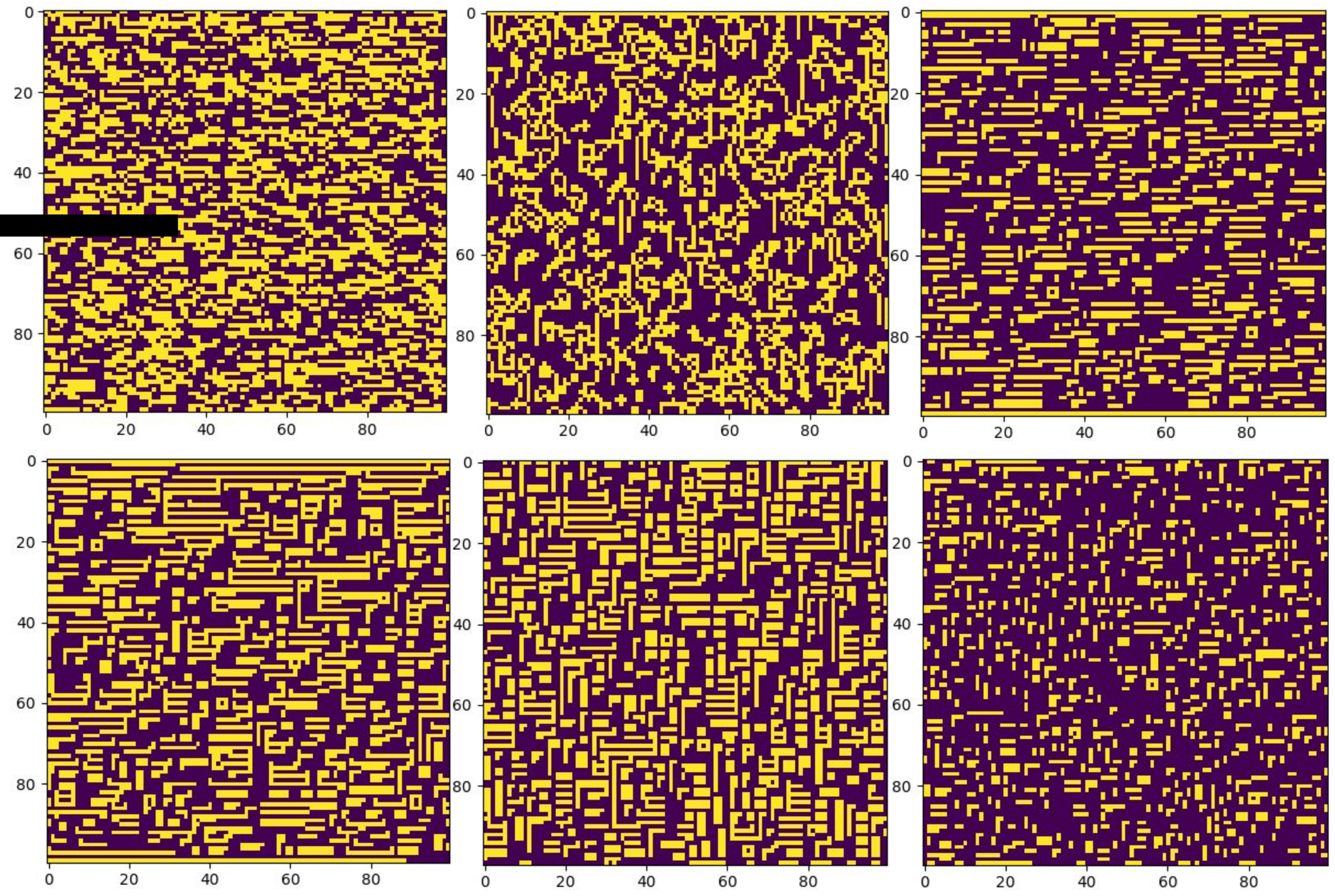
Evolution of Coupled CAs



coupled roots
with driving:
critical

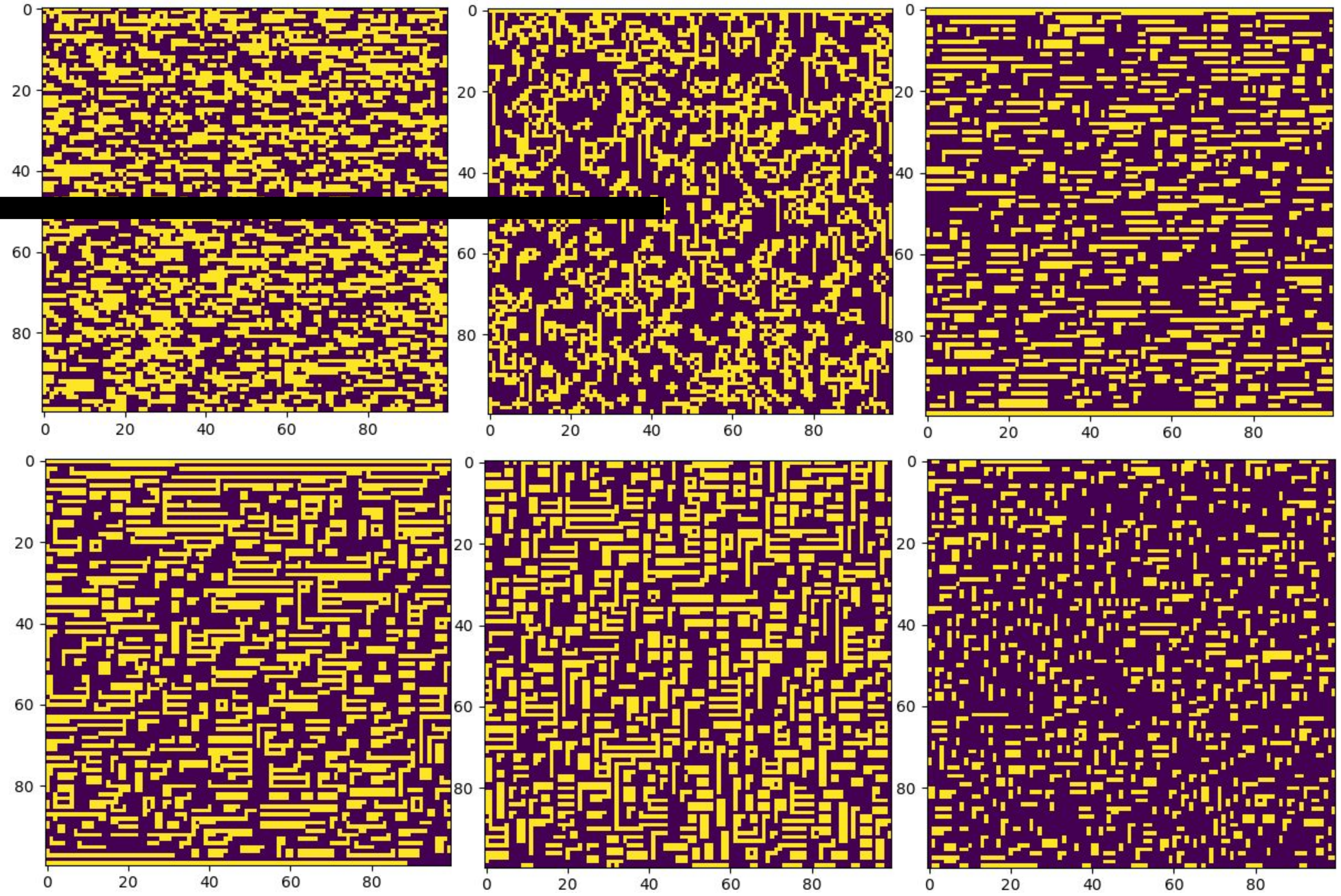
Evolution of Coupled CAs

no
coupling

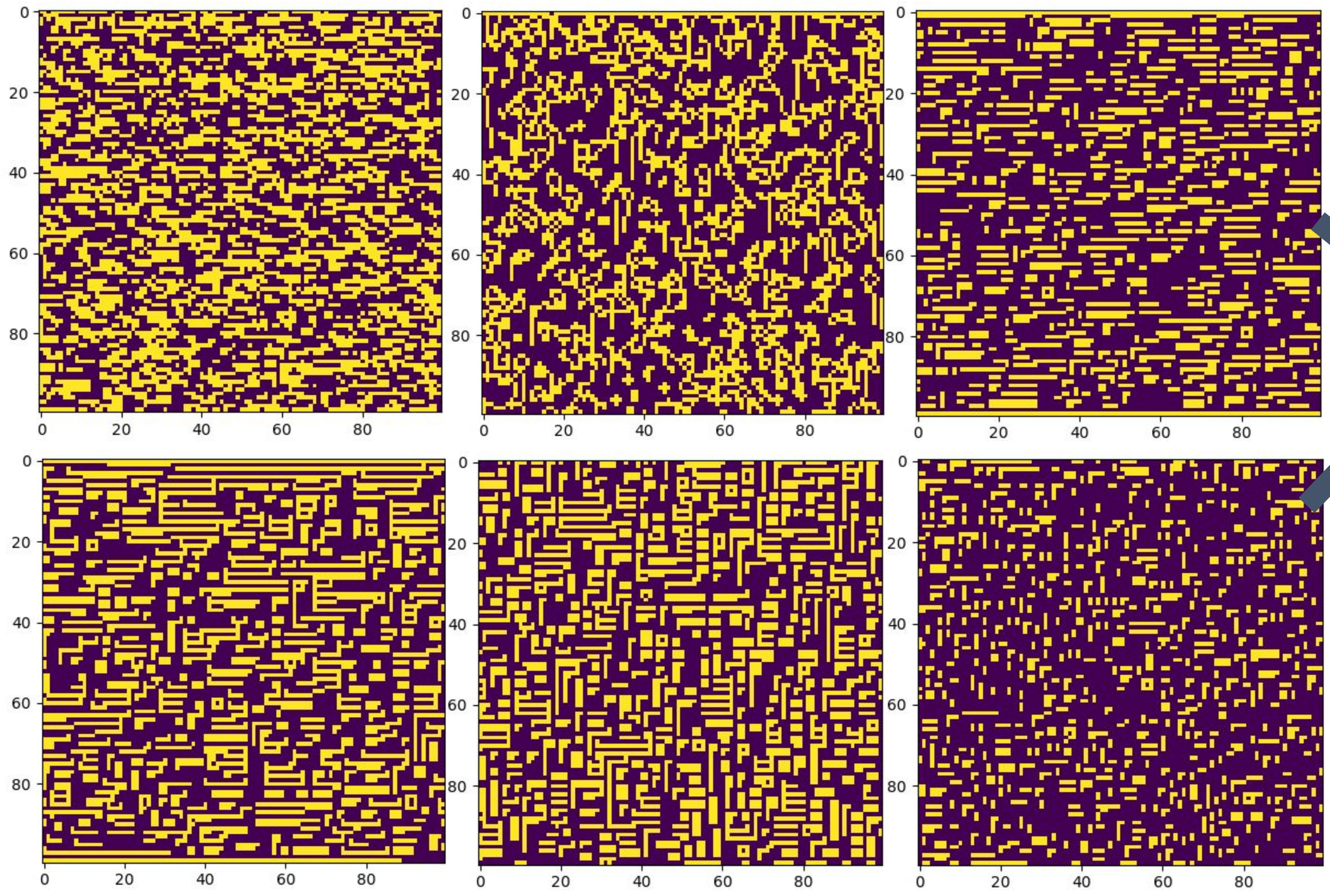


Evolution of Coupled CAs

no drive
motion



Evolution of Coupled CAs



non-critical range

Trigonometric Expression of Coupling

$$\cos^2\left({}^{\pi}/_{8}\right) = \frac{1}{2} + \frac{1}{2\sqrt{2}}$$

$$\sin^2\left({}^{\pi}/_{8}\right) = \frac{1}{2} - \frac{1}{2\sqrt{2}}$$

$$\cos^2\left({}^{\pi}/_{8}\right) - \sin^2\left({}^{\pi}/_{8}\right) = \frac{1}{\sqrt{2}} = 2\sin\left({}^{\pi}/_{8}\right)\cos\left({}^{\pi}/_{8}\right)$$

General Expression:

$$a\cos^2 x - b\sin^2 x = \sin x \cos x$$

5th Rule: Transformation

$a\cos^2 x - b\sin^2 x = \sin x \cos x$

when above equation is divided by $\cos^2 x$:

$\therefore\ b\tan^2 x + \tan x - a = 0$

When the following transformations are applied:

5th Rule: Transformation

$$\cos^2\left({}^{\pi}/_{8}\right) - \sin^2\left({}^{\pi}/_{8}\right) = \frac{1}{\sqrt{2}} = 2\sin\left({}^{\pi}/_{8}\right)\cos\left({}^{\pi}/_{8}\right) = \sin\frac{\pi}{4}$$

$$\frac{j}{i} = \frac{count0}{count1}$$

$$\frac{1}{ratio1} = \frac{{}^{count0}/_{count1}}{{}^{j}/_{i}} = \tan x$$

5th Rule: Transformation

$$-\Delta \cos x \cdot \sin^2 x + \sin x \cdot \cos x - \Delta \sin x \cdot \cos^2 x = 0$$

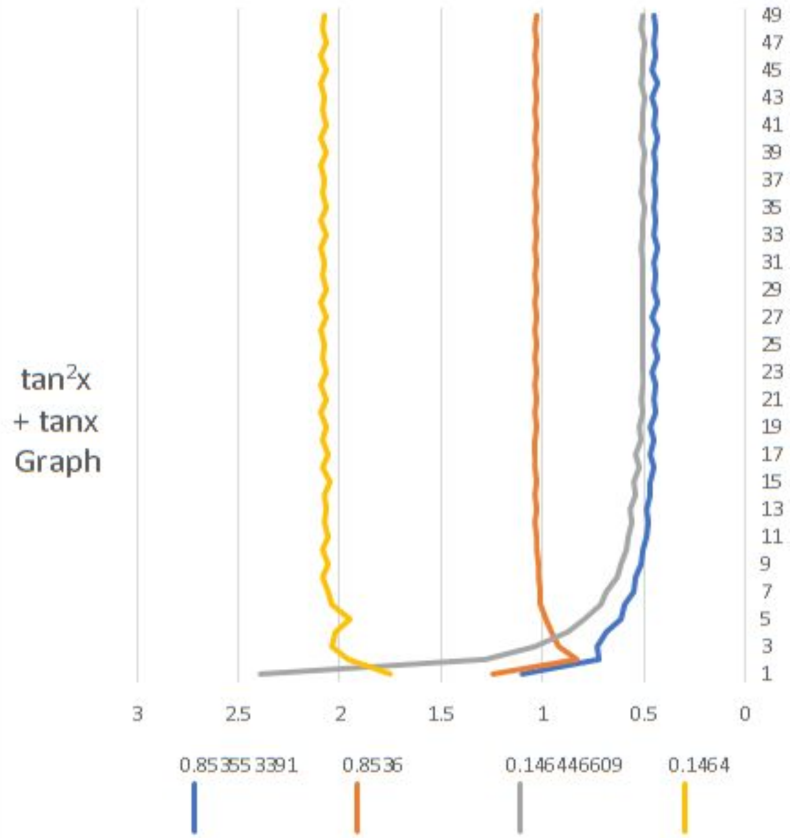
transforms into:

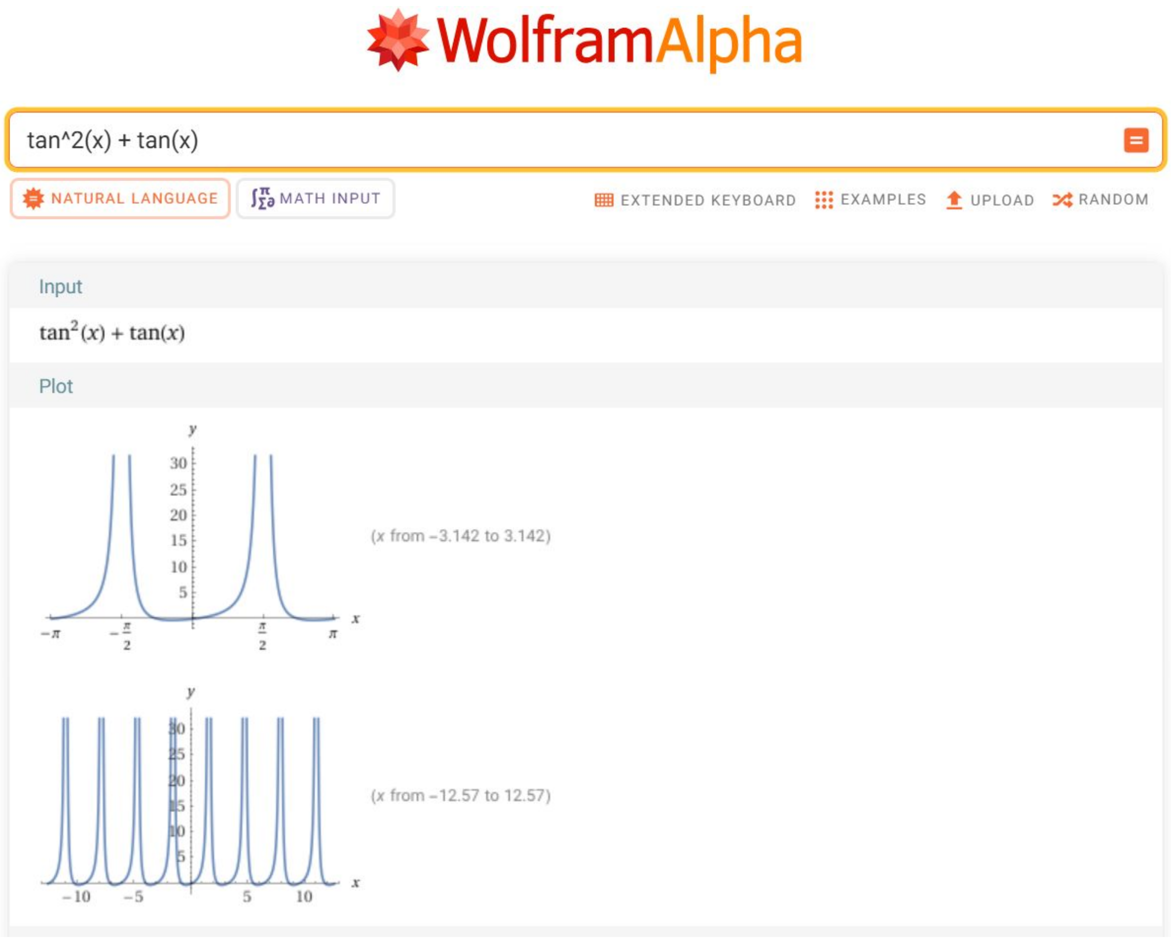
$$\therefore \frac{ratio3}{ratio1 * ratio1} + \frac{1}{ratio1} - ratio2$$

which plots the tangent graph:

$\tan^2 x$ + tanx Graph of CA

Coupled and uncoupled roots are arms of the tangent graph.





Code

https://github.com/goektug/Equation-Automata

Thanks for your time!