<https://writings.stephenwolfram.com/2021/05/the-problem-of-distributed-consensus/>

https://www.wolfram.com/events/distributed-consensus/agenda/

30th June:

any reason to choose 77 seeds? Its kinda random

Principal component analysis?

Dimension reduction?

Anyway to categorise / identify if the program runs till a distinctive colour is given?

More repeats, more nearest neighbours, faster categorisation (qualitativey)

If not enough nearest neighbours, the system will be “indecisive”, and converge into two colours with no convergence back to one

Further analysis of algorithm: runtime, efficiency, least amount of steps, least amount of nn, least amount of time

Fixed amount of repeats, varying amount of nn 🡪 some integer of nn will converge to 2 colours

1st July

Consider models with and without noise

1D 🡪 2D 🡪 3D building up

Linkages to phase transitions, varying pressure and temp

Not sure about what what noises are

Maybe use image color classifiers to determine whether the graph is completely filled with that specific colour?

2nd July: Mainly meeting

Renormalisation group? Look at that

General things to bring up:

What I did:

Tried the first part of the code, tweeked it to have it up to 3 colors,

Fixed amount of repeats, varying amount of nn 🡪 some integer of nn will converge to 2 colours even with many repeats

If not enough nearest neighbours, the system will be “indecisive”, and converge into two colours with no convergence back to one

Brainstorming the next steps:

How to “tell” the program that the image has a “phase transition” property?

Update rules via : Brute force, machine learning?

Any advice / any thing I should look out for?

Need a poster?

Meeting with Mano:

Some ideas:

Random sampling over a variety of initial conditions

Colour activated phenomena 🡪 look at percentage of cell colors

Grid or graph cellular automata? 🡪 sld be grid

Statistical mechanics property 🡪 add noise to the system

Look at attack structure

Look at probabilistic cellular automata

Write a function to return interesting properties returned from the applied rule

Use Table / Map to do the sample & result

Machine learning & Multiway systems 🡪 Keep them in mind first, might be useful

Meeting with Hatem:

Think of rules as mappings & substitutions

See other rules except for the one I have at the moment

Add explanation to some rules

Try conduct a systematic investigation of rules

Use the exhaustive method is ok -- > keep an eye on the simple rules, as they form complex behaviours

Finishing the 3-colored case, we can create a general function / solution for up to n-colored systems

Add section linking phas transitions

Use renormalisation groups to explain system behaviours 🡪 ODE explaining evolution of the system

3rd July: NKS talk

Notion: rather than complicated diff Eq, we can use simple rules to compute complex behaviours

Cellular automata: quite difficult syntax

Hexagonal cellular automata to model fluid flow with 10k or more repeated steps

To find the rules: exhaustive search with confined heuristics, eg look for rules with conservation laws: pg 29, 55-56- 378 of NKS book

Elementary classification: 256 kinds, 4 classes:

1. All cells become same colour 128
2. Stable periodic structure that don’t interact 130 🡪 not yet know computationally universal (hypothesis: edge of chaos)
3. Chaotic pattern 30
4. Complex structure that interact 110 🡪 computationally universal

Computationally universal?

Computational Equivalence: Almost all processes that are not obviously simple can be viewed as computations of equivalent sophistication

5th July

Sudden meeting with Hatem:

Screenshot stuff to prep for

Can use the 256 elementary rules

6th July:

1:1 Meeting with Dr. Wolfram & Hatem

Things to bring up:

The notebook

Any things to add

Directions you would suggest

Questions to ask:

how would you construct the rules?

Is my direction ok?

Any things that I can include and improve?

Rules: too little time for exhaustive search, but can do random search 3^32 rules

Look at rules which are based majority based (majority based rules)

Looking for an analog for majority function of the 3-colored case

Aim: Generalise the GKL rule to 3-colored