CHALMERS | GÖTEBORGS UNIVERSITET

The Hegselmann-Krause Model of Opinion Dynamics in One and Two Dimensions: Phase Transitions, Periodicity and Other Phenomena

Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Gustav Karlsson
Jesper Johansson
Jimmy Ekström
Mattias Danielsson

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet

Göteborg 2015

The Hegselmann-Krause Model of Opinion Dynamics in One and Two Dimensions: Phase Transitions, Periodicity and Other Phenomena

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk fysik vid Chalmers

Jesper Johansson

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk matematik vid Chalmers

Gustav Karlsson Mattias Danielsson

Kandidatarbete i matematik inom civilingenjörsprogrammet Industriell ekonomi vid Chalmers

Jimmy Ekström

Handledare:	Peter Hegarty
Bi-Handledare:	Edvin Wedin
Examinator:	Maria Roginskaya

Institutionen för matematiska vetenskaper
Chalmers tekniska högskola
Göteborgs universitet
Göteborg 2015

Abstract

In this paper we have investigated the Hegselmann-Krause model of Opinion Dynamics in one and two dimensions.

The first result concerns the critical length of an interval for the phase transition from asymptotically almost surely consensus to almost surely fragmentation. Our simulations support the conjecture that the critical length exists and is slightly greater than 5. We have introduced a continuous agent model and calculations have been made by hand, showing the process for the two first time steps assuming a continuum of agents and a uniform initial distribution.

For various domains in $\mathbb{R}^2$, we have investigated the critical area for dilates of the domain. We make a conjecture which says that there exists no critical area for the disc. Results from our simulations suggest that the critical area is heavily dependent on the shape of the domain of the distribution. These simulations have concerned disc, equilateral polygons and equiareal rectangles with different $\frac{width}{height}$-ratios.

We have also investigated square lattices. We discussed the periodic diagonal movement of the corners of a finite square lattice, stating a conjecture of a lower bound on the freezing time. Further we have proved periodicity for the evolution of a half plane lattice $\mathbb{Z}_{\geq 0} \times \mathbb{Z}$ by treating it as a one dimensional self-weighted model. We prove that from the $10th$ time step, the evolution of this configuration is periodic with period 21. To this end we prove basic theorems regarding distance space.

We end the report with a discussion of a self-weighted version of the model that approaches a continuous time model.

Sammanfattning

I denna uppsats har vi utforskat Hegselmann-Krause-modellen av Opinionsdynamik i en och två dimensioner.

Vi börjar med att presentera ett resultat rörande den kritiska längden av ett intervall, för fasövergången från asymptotiskt nästan säkert konsensus till nästan säkert fragmentering. Våra simuleringar stödjer förmodan om att den kritiska längden existerar och har ett värde strax över 5. Vi har introducerat en kontinuerlig agentmodell, där beräkningar som har gjorts för hand visar processen för de två första tidsstegen om vi antar ett kontinuum av agenter med likformigt fördelade åsikter.

Vi har även för olika områden i $\mathbb{R}^2$ utforskat den kritiska arean för dilatationer av dessa områden. Vi gör en förmodan om att det inte finns någon kritisk area för en cirkelskiva. Resultatet av våra simuleringar tyder på att den kritiska arean är starkt beroende av områdets form. Dessa simuleringar har gjorts på cirkelskivor, liksidiga polygoner och ekviareala rektanglar för olika värden på $\frac{bredd}{höjd}$-kvoten.

Vidare har vi gjort undersökningar på kvadratiska gitter, där vi fört en diskussion om den periodiska rörelsen längs diagonalen. Vi lägger fram en förmodan rörande en undre gräns för frystiden. Vi har bevisat att utvecklingen för halvplanet $\mathbb{Z}_{\geq 0} \times \mathbb{Z}$ är periodisk genom att behandla den som en endimensionell modell med självvikt. Vi bevisar att utvecklingen har en period om 21 tidssteg, med början i tidssteg 10. Till detta bevisar vi grundläggande satser för avståndsrummet.

Vi avslutar rapporten med att diskutera en självviktad version av modellen, som går mot en tidskontinuerlig modell.

Foreword

We would like to thank our supervisor Peter Hegarty for always being available and providing help with mathematics, ideas and the writing of this report. We would also like to thank Edvin Wedin, who, although not actually a supervisor for this project has provided assistance and ideas during our meetings. Their interest in our project has greatly inspired and motivated us in our work.

Now for the bureaucracy. The Swedish summary was written by Jimmy. The abstract was written by Jimmy and Mattias. The introduction was written by Jesper. Section 2, The Critical Length of the Interval was written by Jesper, simulations were done by Gustav while the calculations for the continuous model were done by both Gustav and Jesper. Section 3, Critical Areas was written in whole by Gustav and Mattias. Section 4, Periodicity for equidistant lattices is divided in several parts. Jimmy and Jesper wrote the first part, 4.1, Finite Square Lattices. Jesper wrote and did the mathematics behind 4.2.1, Distances Between Agents. Gustav and Mattias wrote and did the mathematics behind the proof of Section 4.2.2. Jesper, Gustav and Mattias wrote Section 4.3, Discussion and Continuous Time Model. The code in Appendix B was written by Jesper. Almost all of the code used in the actual simulations is more specialised and written by Mattias and Gustav.

Jimmy did a lot of work on square lattices, but since we discovered that the simulations are not reliable most of it did not make it into the report. Thus, it might seem, looking at the above list, like his contributions are small, but he is only a victim of bad luck. We, the authors, do not feel that he has contributed less.

We are also obligated to state that we've logged the hours that each member has put into the project and what they have done with those hours.

Contents

1 Introduction

Opinion Dynamics as a mathematical field of study is a recent development. One of the more popular models is one first put forth by Rainer Hegselmann and Ulrich Krause in the late 1990's, known as the *Hegselmann-Krause bounded confidence model of opinion dynamics* (the HK-model for short). [1] [2]

In this model there are a number n of so-called agents which have opinions represented by real numbers. These opinions then change according to

$$x_{t+1}(i) = \frac{1}{|\mathcal{N}_t(i)|} \sum_{j \in \mathcal{N}_t(i)} x_t(j), \tag{1.1}$$

where $x_t(i)$ is the opinion of the i:th agent at a time t and $\mathcal{N}_t(i)$ is the set of *neighbours* of agent i at time t,

$$\mathcal{N}_t(i) = \{j : \|x_t(i) - x_t(j)\| \leq 1\}. \tag{1.2}$$

In words, every agent moves to the average of those that currently have an opinion that differs by at most 1 from its own. In higher dimension, each agent instead has an opinion represented by a vector in $\mathbb{R}^k$, where the distance is taken to be the 2-norm.

This update-rule does not have to be thought of as the evolution of opinions. Rather than talking about opinions we can instead think of semi-automatic robots, each having a "radar" that can see all other robots in a ball with radius 1 unit around them. They are then programmed to move to the average of all the robots they can see. The point of this is trying to get all robots within range of each other.

In the robotics interpretation, it is natural to think of two-dimensional opinions as a "floor" on which they move. However, in this case, one might introduce obstacles such as walls. It would be possible to modify the model so that the agents cannot see through walls, nor move through them. We have not done so.

The purpose of this paper is to analyse and to find new facts concerning the HK-model presented above, mainly in two dimensions since this area of research on the model has very few known results.

We have done a lot of simulation work and formulated conjectures which seem difficult to prove. The simulations were primarily done in two dimensions, mostly concerning critical areas, but also simulations regarding square lattices and some simulations regarding the critical length in one dimension. There is also some work done on a continuous-agent version of the HK-model in one dimension. Our main rigorously proven result is a generalisation of a 1-dimensional result proven by our supervisors Peter Hegarty and Edvin Wedin. We prove that if an infinite number of agents are placed in a square lattice at a distance 1 apart, such that no agent has a negative x-coordinate, then the process will be periodic in a certain sense. In order to prove this we provide sufficient conditions for the dynamics to be periodic. This can be found in Section 4. This leads to a discussion about infinite self-weight and its connection to a continuous time model.

The only feasible way of calculating the evolution is through numerical simulation. We have done this using MATLAB. Sample code, and explanations, can be found in appendix B.

1.1 An Introduction to the Hegselmann-Krause Model

The definition of the update-rule has already been shown in equation (1.1). Let us therefore begin this introduction by looking at an example of the dynamics in action. Consider 5 agents that have initial opinions 0, 1, 2, 3, 4. Then we get the following evolution of the opinions

$$\begin{pmatrix}0\\1\\2\\3\\4\end{pmatrix} \longrightarrow \begin{pmatrix}0.5\\1\\2\\3\\3.5\end{pmatrix} \longrightarrow \begin{pmatrix}0.75\\1.17\\2\\2.83\\3.25\end{pmatrix} \longrightarrow \begin{pmatrix}0.96\\1.31\\2\\2.69\\3.04\end{pmatrix} \longrightarrow \begin{pmatrix}1.13\\1.42\\2\\2.58\\2.87\end{pmatrix} \longrightarrow \begin{pmatrix}1.52\\1.52\\2\\2.48\\2.48\end{pmatrix} \longrightarrow \begin{pmatrix}2\\2\\2\\2\\2\end{pmatrix},$$

which can also be visualised as in Figure 1. The major things that happen in this evolution are:

(i) At time 4, agents 1 and 5 now both have agent 3 as a neighbour.

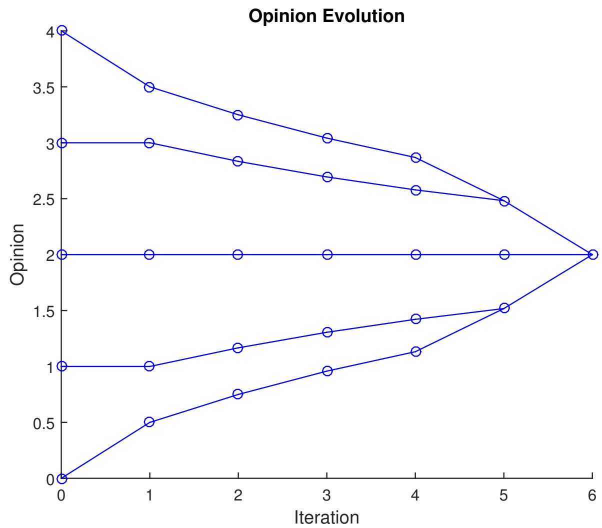

Figure 1: An example of the evolution of a configuration in the HK-model. In this case, the initial configuration was 0, 1, 2, 3, 4. The agents all stabilise at opinion 2, they have reached consensus.

(ii) At time 5, all agents are neighbours, causing them to average to the same value, 2 at $t = 6$.

Motivated by this example we can introduce a few definitions.

The vector containing all the agents' opinions as components, with agent i having opinion $x_t(i)$ at time t, is called the *opinion vector*. In our example, the initial opinion vector would be

$$\boldsymbol{x}_0 = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}. \tag{1.3}$$

In higher dimensions each agent has a vector of opinions, the matrix consisting of these vectors as rows is called the *opinion matrix*.

Two agents that have each other as neighbours are commonly referred to as *seeing* each other. The opinion vector is also commonly known as the *configuration*. There is a convention that in one dimension the opinion vector should always be written in order of increasing opinion. This is because the order is preserved. It is easy to see that this is the case, if an agent is to cross another he would have to see agents that lie beyond the range of the latter, which is impossible[1].

If several agents have the exact same opinion they are called a *cluster*[2]. In the example above, we can see that agents 1 and 2, as well as 4 and 5 formed a cluster at time 5, and all agents formed a cluster at time 6. At this point it is obvious that $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$, nothing more will happen, we call such a configuration *frozen*. Agents that will never move again are also referred to as frozen. The so-called $2r$ conjecture states that there is a tendency for clusters to lie a distance 2 apart. This is a quite vague conjecture and no good precise formulation is available.

The time it takes for a configuration to freeze is called the *freezing time*. In the previous example, the freezing time is 6. It is known that for any finite configuration of agents the freezing-time is finite. We have the following theorem, which has been proven separately by at least two groups, A. Bhattacharya et al [4] and S. Mohajer & B. Touri [5].

Theorem 1.1. *Let $f(n)$ denote the maximal number of steps needed for a configuration of n agents to freeze. Then $f(n) = \mathcal{O}(n^3)$.*

[1] There are variations of the model where this is not impossible, for instance if agents can have different ranges. See the work done by Anahita Mirtabatabaei and Francesco Bullo. [3]

[2] Some authors refer to a cluster as a group of agents that have frozen with the same opinion. We only demand that the agents have the same opinion. In one dimension there is no difference between these two conventions.

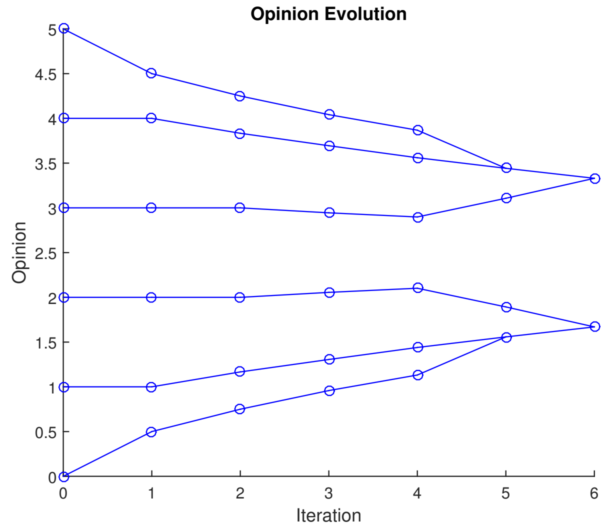

Figure 2: An example of the evolution of a configuration in the HK-model. In this case, the initial configuration was 0, 1, 2, 3, 4, 5. We do not get consensus, as we did in Figure 1.

It is thought that this bound is larger than it has to be, since no known one-dimensional sequence of configurations actually takes this long to freeze. The sequence of configurations that freeze the slowest among the ones we know today has a freezing time that is $\Omega(n^2)$ and was constructed by Peter Hegarty and Edvin Wedin [6]. These configurations, $\mathcal{D}_n$, consist of $3n+1$ agents with initial opinions

$$x_0(i) = \begin{cases} -\frac{1}{n} & \text{if } 1 \leq i \leq n, \\ i-(n+1) & \text{if } n+1 \leq i \leq 2n+1, \\ n+\frac{1}{n} & \text{if } 2n+2 \leq i \leq 3n+1. \end{cases} \tag{1.4}$$

If all agents share the same opinion when frozen the configuration is said to have reached *consensus*. This does *not* always happen, in fact it is quite rare and mostly happens when the agents are spread over a small interval. Simply by adding another agent to our configuration in (1.3), making

$$\boldsymbol{x}_0 = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, \tag{1.5}$$

we get the evolution that can be seen in Figure 2. In this case, our agents split into two camps that freeze at $\frac{2885}{1728}$ and $\frac{5755}{1728}$. Thus we do not get consensus with this configuration. Instead we say that the configuration has *fragmented*.

There is a conjecture regarding configurations on an interval on the line and whether or not they reach consensus. It states that there is a critical length $\mathcal{L}$ such that for a large number of agents consensus tends to be reached asymptotically almost surely if the length of the interval over which agents are initially spread is shorter than $\mathcal{L}$, and there tends to be fragmentation if it is longer [7]. This conjecture is explored in more detail in section 2, where we also give a precise formulation. We have also studied generalisations of this conjecture in two dimensions in section 3.

The graph consisting of nodes representing the agents and where two nodes are connected if they can see each other is called the *influence-graph* of the configuration, mostly referred to as just the graph. This contains all the information needed to construct the dynamics. In one dimension it is true that if the graph contains disconnected subgraphs, those subgraphs will remain disconnected. This is because if an agent sees no other to its left, it will never move to the left, and likewise for right. Thus if there is a gap greater than 1, it can never close.

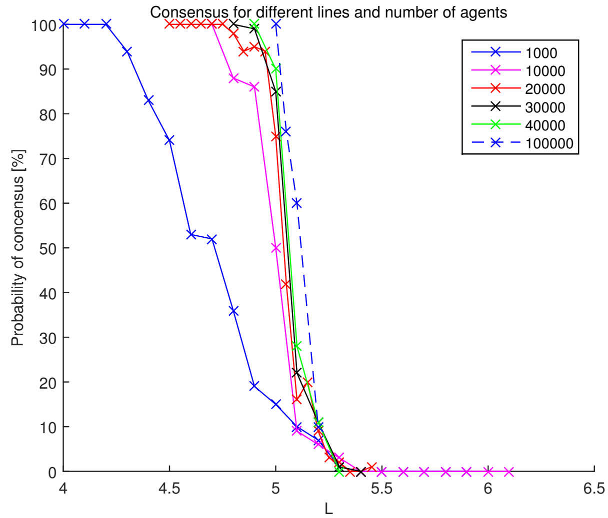

Figure 3: Percentage of simulations that reached consensus for different numbers of agents placed uniformly i.i.d on an interval. As the number of agents is increased, the distribution approaches a step-function, as predicted by Conjecture 2.1. We can also see that the critical value $\mathcal{L}$, if it exists, is probably somewhere just above 5.

By looking at the definition of the update-rule (1.1) we see that for each fixed time the evolution is a linear transformation of the opinion vector. We can therefore write

$$\boldsymbol{x}_{t+1} = \mathbf{H}_t \boldsymbol{x}_t, \tag{1.6}$$

where $\mathbf{H}_t$ is called the *transition matrix* at time t. The components of $\mathbf{H}_t$ are

$$(\mathbf{H}_t)_{ij} = \begin{cases} \frac{1}{|\mathcal{N}_t(i)|} & \text{if agents } i \text{ and } j \text{ are neighbours,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.7}$$

It is important to remark, however, that $\mathbf{H}_t$ depends on $\boldsymbol{x}_t$ so the update-rule is not in fact a linear transformation, only piecewise linear. Even so, it is often useful to work with these matrices.

2 The Critical Length of the Interval

Let us consider the HK-model in one dimension. Specifically, consider the case of a large number, n, of agents with opinions that are uniformly i.i.d on $[0,L]$ for some L. Let us ask what is the probability of reaching consensus, $P_C(L,n)$. There is the following conjecture regarding this, first precisely formulated by Edvin Wedin and Peter Hegarty. [7]

Conjecture 2.1. *There is a critical length of the interval, $\mathcal{L} \approx 5$, such that*

$$\lim_{n\to\infty} P_C(L,n) = \begin{cases} 1 & \text{if } L < \mathcal{L} \\ 0 & \text{if } L > \mathcal{L}. \end{cases} \tag{2.1}$$

The above conjecture is pretty well supported by simulations. The results of our simulations can be seen in Figure 3, which points to the critical length being approximately five. There are a lot of open questions regarding this conjecture. It is not known whether such an $\mathcal{L}$ actually exists. Indeed it is not even known that if for some L_0 the probability for consensus tends to zero as the number of agents tends to infinity, then it also tends to zero for any $L > L_0$.

One approach to proving this conjecture is shown below, and it highlights why this is harder to prove than it seems; the complexity of the distribution of the agents quickly becomes overwhelming.

2.1 A Continuous Agent Model

In order to get some understanding of what happens when the number of agents is increased, as is the case in the conjecture regarding the critical value, we consider a continuous variety of the HK-model. In this model, we introduce a density of agents, that describes how many agents have a certain opinion.

Let $F_t(x)$ at a time t be defined as the ratio of the number of agents to the left of x and the total number of agents. Then we define f_t, possibly a distribution, by

$$F_t(x) = \int_{-\infty}^{x} f_t(\xi)\,\mathrm{d}\xi, \tag{2.2}$$

or equivalently,

$$f_t = \frac{\mathrm{d}F_t}{\mathrm{d}x}. \tag{2.3}$$

We can define the Hegselmann-Krause-operator, $H\!K_t(x)$, which maps an agent with opinion x at time t to its opinion at time $t+1$,

$$H\!K_t(x) = \frac{\int_{x-1}^{x+1} \xi f_t(\xi)\,\mathrm{d}\xi}{\int_{x-1}^{x+1} f_t(\xi)\,\mathrm{d}\xi}. \tag{2.4}$$

This is clearly equivalent to the finite case when f_t is a finite sum of delta-functions. The update-rule for F_t is then

$$F_{t+1}(x) = \int_{L_t(x)} f_t(\xi)\,\mathrm{d}\xi, \tag{2.5}$$

where $L_t(x)$ is the set of all agents that get mapped to the left of x by $H\!K_t$.

From (2.4) it is clear that $H\!K_t(x)$ is a non-decreasing function of x, and therefore agents in this continuous agent model can never change their order. Then, since $H\!K_t$ is non-decreasing it follows that $L_t(x) = (-\infty, \sup\{y : H\!K_t(y) < x\})$. If $H\!K_t$ is non-constant around x it is locally invertible and thus $L_t(x) = (-\infty, H\!K_t^{-1}(x)]$. We will be somewhat sloppy and use $H\!K_t^{-1}(x)$ to refer to both cases.

Using this we get the following alternative form of (2.5),

$$F_{t+1}(x) = \int_{-\infty}^{H\!K_t^{-1}(x)} f_t(\xi)\,\mathrm{d}\xi, \tag{2.6}$$

and therefore

$$f_{t+1}(x) = f_t\left(H\!K_t^{-1}(x)\right) \cdot \left(H\!K_t^{-1}(x)\right)' = \frac{f_t\left(H\!K_t^{-1}(x)\right)}{H\!K_t'(H\!K_t^{-1}(x))}. \tag{2.7}$$

We need to define what consensus means in this model. There is one obvious way to define this.

Definition 2.1. *Given an initial distribution f_0 we say that it will reach consensus if there exists an opinion o such that*

$$\lim_{t\to\infty} F_t(x) = \theta(x - o), \tag{2.8}$$

where θ is the Heaviside step function and the limit is taken in the L_2 norm.

Remark 2.1. *If the configuration does reach consensus in the above sense, it is clear that it must do so in finite time since all agents must then eventually be within distance 1 from one another.*

We developed this continuous model because the following conjecture seems likely.

Conjecture 2.2. *Let n agents have opinions that are i.i.d random variables with distribution f_0. Let $P(n)$ be the probability that they reach consensus.*

Then under suitable conditions on f_0, if the continuous agent model with f_0 as initial distribution reaches consensus then $P(n) \to 1$ as $n \to \infty$ and if it fragments under the continuous model then $P(n) \to 0$ as $n \to \infty$.

Remark 2.2. *It is obvious that this conjecture can be generalised to higher dimensions.*

Remark 2.3. *If this conjecture is true then it implies a 0-1 law, that the probability either tends to 0 or 1 as the number of agents tends to infinity, regardless of the initial distribution.*

2.1.1 Attempt at Proving the Conjecture Regarding the Critical Length

Let us now consider the case where, for $x \in [0,L]$,

$$f_0(x) = \frac{1}{L} \tag{2.9}$$

that is to say, the opinions are distributed uniformly on $[0,L]$.

In order to calculate f_1 we need to determine $H\!K_0^{-1}$, which means we need to determine $H\!K_0$. We have for $1 \le x \le L-1$

$$H\!K_0(x) = \int_{x-1}^{x+1} \xi \,\mathrm{d}\xi \Big/ \int_{x-1}^{x+1} \mathrm{d}\xi = x. \tag{2.10}$$

For $0 \le x \le 1$ we have

$$H\!K_0(x) = \int_{0}^{x+1} \xi \,\mathrm{d}\xi \Big/ \int_{0}^{x+1} \mathrm{d}\xi \tag{2.11}$$

$$= \left[\frac{\xi^2}{2}\right]_0^{x+1} /(x+1) \tag{2.12}$$

$$= \frac{x+1}{2}. \tag{2.13}$$

Since f_0 clearly is even around $L/2$ we do not need to calculate the special case on the far side of $L/2$.

Thus we have

$$H\!K_0(x) = \begin{cases} \frac{x+1}{2} & \text{if } 0 \le x \le 1 \\ x & \text{if } 1 < x \le L/2. \end{cases} \tag{2.14}$$

This allows us to calculate

$$H\!K_0^{-1}(y) = \begin{cases} 2y-1 & \text{if } \frac{1}{2} \le y \le 1 \\ y & \text{if } 1 < y \le L/2. \end{cases} \tag{2.15}$$

By plugging this into our update-rule (2.7) we get

$$f_1(x) = \begin{cases} 0 & \text{if } 0 \le x < \frac{1}{2} \\ \frac{2}{L} & \text{if } \frac{1}{2} < x < 1 \\ \frac{1}{L} & \text{if } 1 < x \le L/2 \\ \text{and evenly symmetrical around } L/2. \end{cases} \tag{2.16}$$

If we repeat this process for our new density we find that

$$f_2(x) = \begin{cases} 0 & \text{if } 0 \le x < \frac{11}{12} \\ \frac{2}{L}\left(1 + \frac{x}{\sqrt{x^2-1/2}}\right) & \text{if } \frac{11}{12} < x < \frac{9}{8} \\ \frac{1}{L}\left(1 + \frac{x}{\sqrt{x^2-1/2}}\right) & \text{if } \frac{9}{8} < x < \frac{27}{20} \\ \frac{1}{L}\left(1 + \frac{1-x}{\sqrt{(x-1)^2+8}}\right) & \text{if } \frac{27}{20} < x < 2 \\ \frac{1}{L} & \text{if } 2 < x \le L/2 \\ \text{and evenly symmetrical around } L/2 \end{cases} \tag{2.17}$$

At this point $H\!K_2$ has become very complex. For example, in the interval $\frac{47}{20} \le x < 3$ we have

$$H\!K_2(x) = \frac{4\ln 8 - \frac{9}{8} - 4\ln\left(2(x-2) + 2\sqrt{(x-2)^2+8}\right) + \frac{x}{2}\sqrt{(x-2)^2+8} + 2x}{\sqrt{(x-2)^2+8} - 1}. \tag{2.18}$$

This function, while invertible, is not invertible in elementary functions. Therefore calculating the densities beyond this point must be done numerically. Doing it numerically doesn't allow us to decide if the conjecture is true.

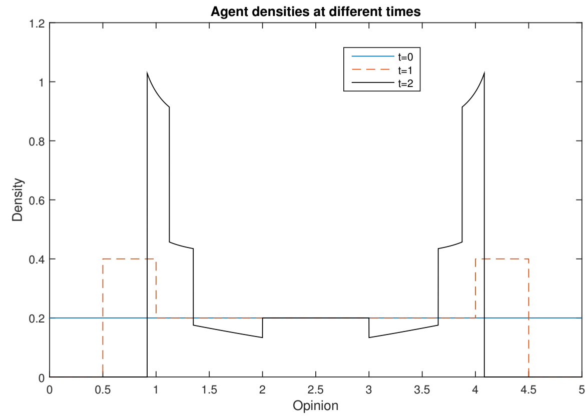

Figure 4: Densities at different times when starting with a uniform distribution on [0,5]. The complexity of the function increases rapidly, and the number of discontinuities in the density increases exponentially.

3 Critical Areas

We have looked at the one-dimensional case, and concluded that there probably exists a so-called critical length. We will now study the same behaviour in two dimensions.

In two dimensions besides the density of agents and the area (analogous to the length of an interval in one dimension), the shape is also likely to influence the probability of consensus.

For example, any number of agents distributed on a disc with area $\frac{\pi}{4}$ will always reach consensus in one time step since all agents can see each other. This is not the case for a widely stretched rectangle with the same area.

Before proceeding we need to define precisely what the concept of a critical volume in $\mathbb{R}^d$ (i.e critical area in $\mathbb{R}^2$) really is.

Definition 3.1. *Consider a fixed domain $D \subset \mathbb{R}^d$ with volume 1. Let $P_C(a,n)$, $a \in \mathbb{R}^+$, be the probability that n agents with opinions that are uniformly i.i.d random variables on aD reach consensus. If there exists an a_c such that for $a < a_c$, $P_C(a,n) \to 1$ as $n \to \infty$ while for $a > a_c$, $P_C(a,n) \to 0$ as $n \to \infty$, then the volume of $a_c D$, a_c^d is called the critical volume of all domains with the shape of D and a_c is called the critical value.*

Remark 3.1. *It is not proven that there exists a critical length of an interval in one dimension, even though it seems likely. We do not have a proof that the same phenomenon exists in two dimensions either, in fact we suspect that there might be domains that don't have finite critical areas.*

Remark 3.2. *In this section, unless otherwise noted, $d = 2$.*

From the definition, two questions arises: For a fixed d, does there exist an a_c for all D and is it independent of D? If not, which shapes are optimal in the sense that they have the largest critical area in some family of shapes?

We start by trying to answer the first question with a theoretical discussion backed by results from simulations on discs and equilateral triangles. The second question is examined in two ways. First we use the result from the first discussion and investigate equilateral polygons and their probability of consensus. Secondly we expand the question and investigate how different width/height ratios of equiareal rectangles affect the probability of consensus.

3.1 Critical Area of the Disc

By running the HK-model on agents uniformly i.i.d on a disc we have seen that the agents tend to group up in circles inside of each other. One can see this tendency in Figure 5. The circles observed in the disc-simulations should tend to perfect circles as the number of agents tends to infinity, which should reach consensus.

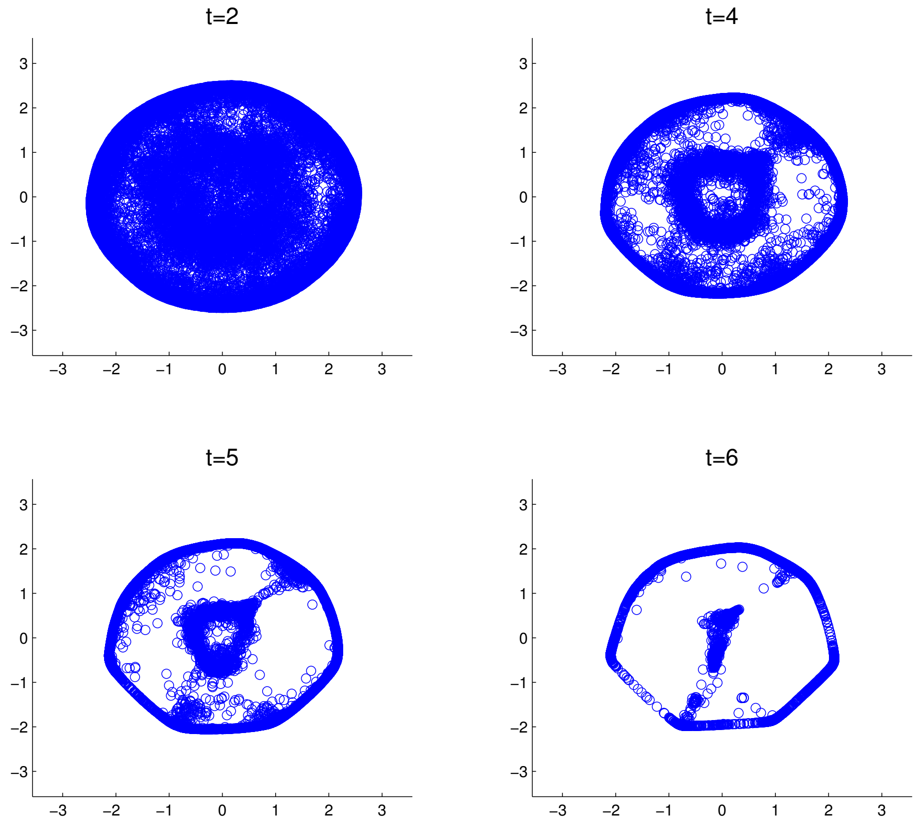

Figure 5: An evolution of 25600 uniformly i.i.d agents on a disc with area 40. At $t = 5$ we see that two "circles" have formed inside one another.

More precisely, if we accept the generalisation of Conjecture 2.2 in two dimensions then the probability of consensus should tend to 1 for any area of the disc as the number of agents tends to infinity. This is because it is obvious that a uniform distribution of agents on the disc will reach consensus in the continuous agent model. Thus we state the following.

Conjecture 3.1. *If D is a disc of area 1 then, in the notation of Definition 3.1, $P_C(a,n) \to 1$, for all $a > 0$, as $n \to \infty$.*

The data from our simulations support Conjecture 3.1. Looking at Figure 6, we see that increasing the number of agents exponentially, doubling each time, seems to move the curve roughly the same distance to the right every time. We expect this behaviour to continue if the number of agents is increased further.

The number of simulations in the underlying data can be found in the appendix. Worth mentioning here however, is that the curve for 51200 agents is based only on between 20 and 50 simulations for each area, and is not to be thought of as an unquestionable result.

3.2 Critical Area of The Equilateral Triangle

From the simulations on the disc we saw that it doesn't appear to have a critical area. We will now seek to motivate that there are shapes that in fact do have critical areas.

Comparing Figure 6 to Figure 7 we see that the percentage of configurations reaching consensus as a function of the area seems to approach a step function as the number of agents are increased exponentially on the triangle, while on the circle the curves move to the right. This means that the equilateral triangle probably has a critical area, around 18.

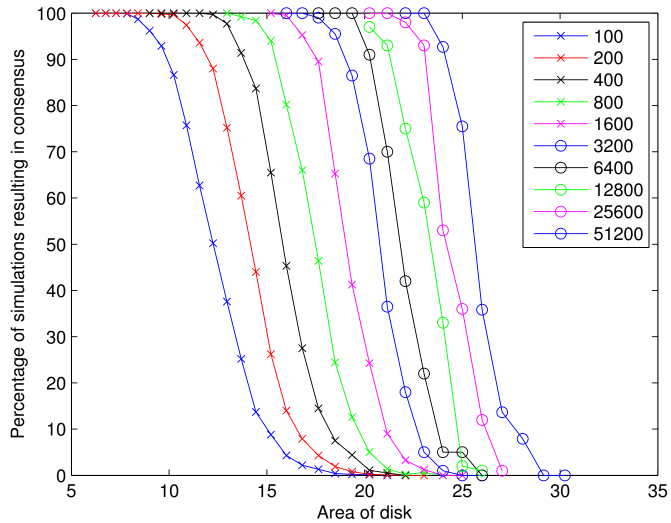

Figure 6: Percentage of simulations resulting in consensus as a function of the area of the disc on which the agents were initially distributed. The curve moves to the right as the number of agents is increased, rather than tending to a step-function as in Figures 3 and 7. Thus there doesn't appear to be a critical area.

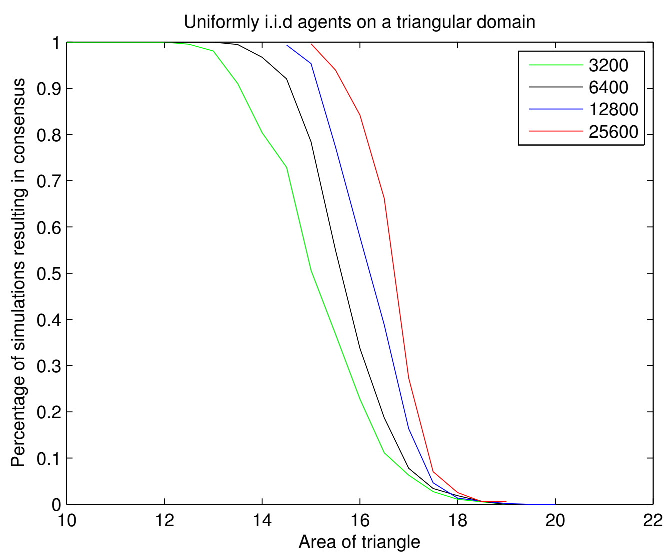

Figure 7: Percentage of simulations resulting in consensus as a function of the area of the triangle on which the agents were initially distributed. It seems like it's approaching a step-function, just like in Figure 3. Therefore it seems like there is a critical area for the triangle and that it is around 18.

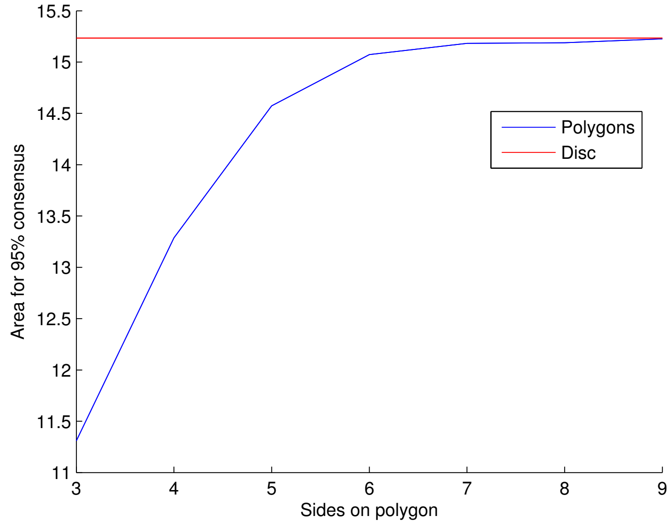

Figure 8: Approximate area for which 800 agents reach consensus 95% of the time as a function of the number of sides on the polygon. It seems like its approaching the area for the disc as the number of sides increases.

Thus we have seen one shape that probably doesn't have a critical area, the disc, and one that probably does, the triangle. Now we seek to answer the second question, what shapes are optimal? We will begin by looking at the influence that corners and edges have in equilateral polygons and then we will turn to asymmetry by looking at rectangles.

3.3 Behaviour of Equilateral Polygons

Concerning distributions of agents on equilateral polygons, a reasonable conjecture to be made is the following:

Conjecture 3.2. *The critical area of an equilateral polygon with $N \geq 3$ sides exists, increases monotonically with N and tends to infinity as N tends to infinity.*

The reasoning behind this conjecture is simply that a polygon with N sides will approach the appearance of a disc as $N \to \infty$. Another aspect is that having corners will probably lower the probability of consensus, since clusters tend to form near them.

When studying finitely many uniformly i.i.d agents, we have the odd chance of agents starting in configurations which will not result in consensus. Looking at the area for which all of the simulations result in agents reaching consensus is therefore not a useful way to examine how different factors influence the probability of consensus. A way to avoid this problem, while still having results that can support our conjecture, is to instead look at the area for which a certain percentage of simulations result in consensus for a fixed number of agents. We see that this is an eligible approach since we can see in Figures 6 and 7 that the curves always have a sharp descent and look similar near the critical area, if it exists. The "if it" exists is important, as it appears impossible to tell from a fixed number of agents whether a shape has a critical area. The triangle probably does have a critical area however, so we have reason to assume that any polygon will as well.

Figure 8 supports conjecture 3.2 as it's easier to reach consensus for larger areas when the number of sides is increased. The figure shows that the area for which 95% of our simulations result in consensus seem to grow as a function of the number of sides of the domain, with the limit being the "95%-consensus area" of the disc for 800 agents.

We conclude that corners and sharp edges probably lower the probability of consensus, and that Conjecture 3.2 is probably true.

3.4 Behaviour of Agents Distributed on Equiareal Rectangles.

What happens if we shift the relationship between the side lengths of a rectangle with a given area, if the width-height ratio is increased while keeping the area constant?

What happens for the percentage of configurations reaching consensus as the ratio is increased? Can we say anything about the supposed critical area for a given ratio? For example, does the critical area decrease monotonically as the ratio is increased? Likewise, does there exist something like a critical ratio for a given area?

Definition 3.2. *$r_c(A)$ is a critical ratio for a given area A of the rectangle if*

$$\lim_{n\to\infty} P_C(r,A,n) = \begin{cases} 1 & \text{if } r < r_c \\ 0 & \text{if } r > r_c. \end{cases} \tag{3.1}$$

where $P_C(r,A,n)$ is the probability that n agents reach consensus on a rectangle with area A and $\frac{width}{heigth}$-ratio, r.

Fixing a shape and increasing the number agents in the simulations (as we have done in Figure 6) when investigating several ratios would be incredibly time consuming. This made us settle on investigating four different areas for a given density (400 agents/unit area) and let the ratio change for all of them.

Before presenting the results we want to introduce what we call "semi-stable configurations". These are examples of configurations with higher freezing times than normal. In order to present this we first need to define what a *stable* configuration is. We will use the definition presented by V.D. Blondel, J.M. Hendrickx, and J.N. Tsitsiklis [8] where we consider a weighted HK-model where each agent i has an associated weight $w_i \in \mathbb{R}^+$, and where the update rule is as follows:

$$x_{t+1}(i) = \frac{\sum\limits_{j\in N_t(i)} w_j x_t(j)}{\sum\limits_{j\in N_t(i)} w_j}. \tag{3.2}$$

Definition 3.3. *Let $\tilde{x}$ be a frozen configuration. Add a perturbing agent with opinion x_0 and weight m to the configuration, update according to (3.2) until a new frozen configuration $\tilde{x}'$ is obtained, and then remove the perturbing agent. $\tilde{x}$ is called a stable configuration if*

$$\sup_{x_0} \sum_i w_i |\tilde{x}_i - \tilde{x}'_i| \to 0, \text{ as } m \to 0. \tag{3.3}$$

Remark 3.3. *Note that (3.2) is equivalent to (1.1) if we let $w_i = 1\ \forall i$. Further, note that this makes the definition of stable configurations consistent with (1.1).*

In words, a configuration is stable if and only if, by adding an agent of arbitrarily small weight, we cannot induce major changes in it.

The "semi-stable configurations" are the following configurations, which are examples of unstable configurations where such a perturbing agent has been added.[3] The name is slightly confusing. If we have clusters of n_0 agents at x_0, n_1 agents at x_1 and n_2 agents at x_2, where $x_0,x_1,x_2 \in \mathbb{R}^k, k \in \mathbb{N}$, we will refer to the configuration as semi-stable if

i $n_0 \approx n_2$

ii $n_1 << n_0,n_2$

iii $\|x_1 - x_0\| \leq 1$

iv $\|x_1 - x_2\| \leq 1$

v $\|x_0 - x_2\| > 1$.

As the semi-stable configuration updates (with notation as above), x_0 and x_2 will move a distance proportional to $\frac{n_1}{n_0}$ and $\frac{n_1}{n_2}$ respectively, towards x_1.

We have investigated areas and ratios for configurations where the final configuration will have either two clusters at a distance approximately 2, or one cluster. This often leads to semi-stable configurations when we are close to criticality.

[3]Blondel, Hendrickx and Tsitsiklis went the other way around, and used these configurations as motivation for introducing the definition of a stable configuration.

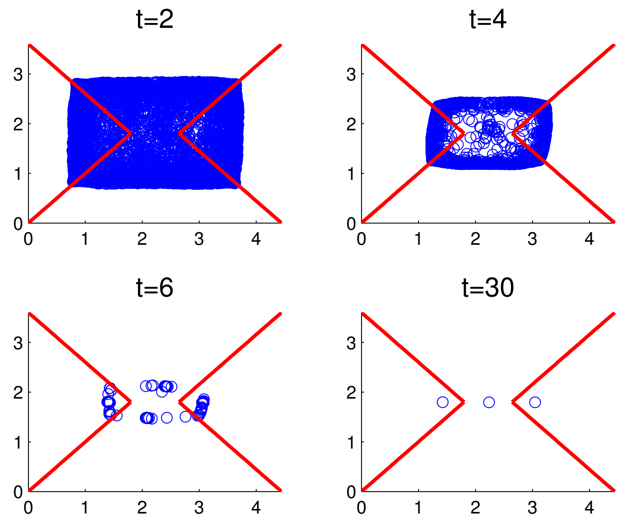

Figure 9: Evolution of 6400 uniformly random i.i.d agents on a rectangle with area 16 and width/height ratio 1.11. Corner clusters approximately move along red lines. Red lines meet at what we refer to as the "short-end points". At $t = 30$ the configuration is semi-stable, and will eventually reach consensus.

3.4.1 Results and discussion

When looking at the simulations we have observed two different behaviours of the rectangles: the "square" behaviour and the "rectangle" behaviour. For both behaviours clusters will form at the corners. What induces the changes in the behaviours is the location of the points towards which the clusters tend to move, before they see a cluster from another corner. It's obvious that the corners at the short ends ultimately tend towards the same point due to symmetry, so the essential difference in the behaviours depends on how far away these short-end points (as in Figure 9) that attract the clusters are from each other.

The "rectangle" case appears if these points are far away from each other. The corners will collapse into two clusters, one at each short end side, long before they can see another cluster from the far side.

In the "square" case, however, this behaviour will not appear. The configuration will be more dependent on how the agents are initially distributed and it's hard to tell which corners will see which corners first.

Further, in the "rectangle" behaviour there are semi-stable configurations and non-semi-stable configurations. Semi-stable configurations appear when the two short-end-side-clusters are at a distance ≤ 1 from the middle, where a small cluster most often occurs. As said above, these configurations slowly move towards the middle which yields a greater freezing time, but consensus will eventually be reached, at least most of the time.

The non-semi-stable configurations appear when the distance from the middle to at least one of the short-end clusters is greater than one, hence, this cluster won't tend to the middle. As a result these configurations have a smaller freezing time and will not reach consensus.

Looking at Figure 11 there is one tendency that is particularly interesting. The consensus percentage of $A = 16$ and $A = 15$ appears to be not strictly decreasing as a function of the ratio, which is surprising. We believe this is a result of a behavioural transition from the "square" behaviour to the more "rectangle" behaviour around ratio 1.2. This theory is supported by Figure 13 where we see a significant increase of the average freezing time for these ratios. The tendency is also true for density 800. If this is true as the number of agents tends to infinity, then, as a consequence, there would be no critical ratio as we define it, for these areas. Also, if the probability of consensus tends to some number between 0 and 1 for some ratio around 1.2 as the number of agents tends to infinity, this would indicate that there is no critical area for this ratio and contradict the supposed $0 - 1$ law discussed in Remark 2.3.

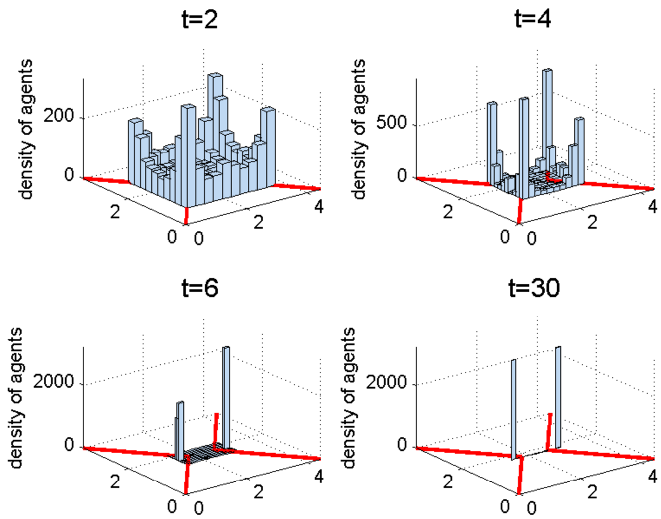

Figure 10: 3d-histogram for the evolution depicted in Figure 9. At $t = 30$ there is a cluster of 4 agents in between the two larger clusters, not visible in the figure.

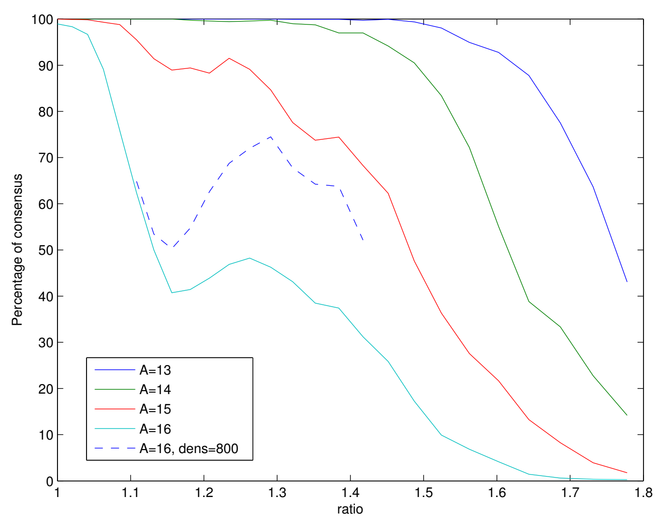

Figure 11: Percentage of simulations reaching consensus for uniformly i.i.d agents with a density of 400 agents/unit area (solid lines) and 800 agents/unit area (dashed line) on equiareal rectangles as a function of the $\frac{width}{height}$-ratio of the rectangle. Looking at ratios between 1.15 and 1.25 the percentage appears to be increasing. Note that the minimum before the percentage rises also increases as the number of agents is increased. This indicates that either the probability tends to 1 and the 0-1 law is true, or, more interestingly, the probability of reaching consensus could be tending to something else that is neither 0 nor 1.

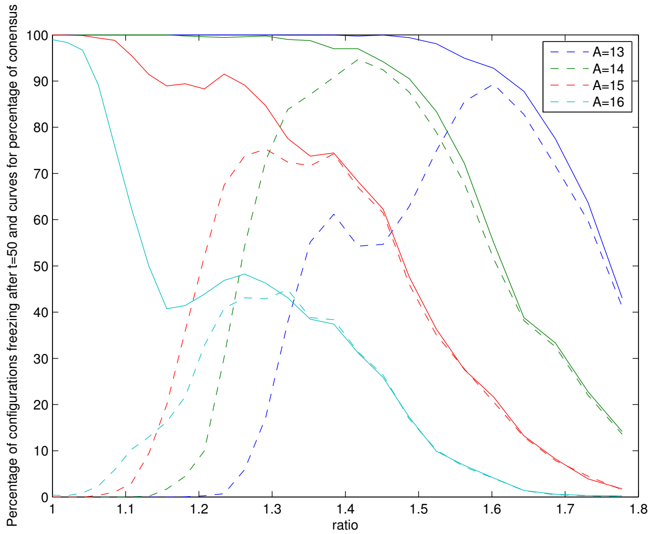

Figure 12: Figure 11 (solid lines) where the percentage of simulations with a freezing time greater than 50 has been superimposed (dashed lines). At higher ratios the curves overlap very closely.

Figure 12 depicts the consensus percentage as well as the percentage of configurations resulting in a freezing-time greater than 50. From this it seems reasonable to conclude that if consensus is reached for configurations with ”rectangle” behaviour, then the configuration became semi-stable. The reversed case is seemingly also true, that semi-stable occurrences almost always reach consensus. This is not at all certain for higher ratios.

Since the percentage of configurations reaching consensus is an increasing function of the ratio in the interval (1.15,1.25) both for a density 400 agents/unit area and 800 agents/unit area, it would be interesting to do simulations including an even larger number of agents in this interval in order to get a hold on what's going on. We have at least 1300 simulations for all ratios and areas. For ratios in the interval (1.15,1.25) we have at least 2300 simulations, and are thereby confident in our results.

We also believe that semi-stable occurrences are much more frequent around r_c, if it exists, compared to its surrounding. As a result we suggest that, given an area that has a critical ratio, for a large enough number of agents, if we approach the critical ratio from below, then the average freezing time would increase and reach maximum just around r_c.

Further it would be relevant to investigate how the critical area depends on the ratio, r, as $r \to \infty$. As $r \to \infty$, then $A_c \to 0$, and the width of the critical rectangle ought to tend to the critical length of the interval.

3.5 Conclusions

We've found that there probably is a critical length of the interval in $\mathbb{R}^1$. As we move from one dimension to two we can have shapes that appear to have no critical area, for example the disc, and some that do, for example the equilateral triangle. Corners and sharp edges are bad for reaching consensus and going from a square to a rectangle leads to simulations with surprising behaviour, where increasing the asymmetry of the rectangle first tends to decrease the probability of consensus, then increase it, and then reduce it again.

4 Periodicity for equidistant lattices

We will now stop our discussion on critical lengths and areas and instead talk about periodic evolutions in configurations where the agents are placed at fixed distances from their neighbours. We will begin by stating a result in one dimension, and then try to generalise it to two dimensions.

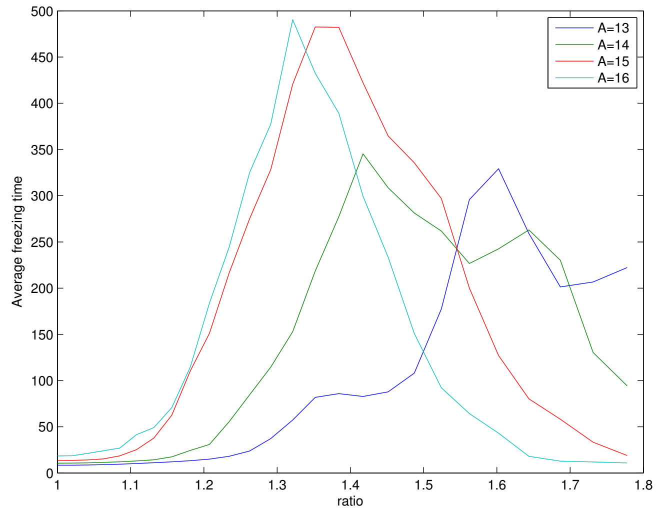

Figure 13: Average freezing time for uniformly i.i.d agents with a density of 400 agents/unit area on equiareal rectangles as a function of the $\frac{width}{height}$-ratio of the rectangle. Maxima of the curves are thought to be around the last ratio where the semi-stable configurations can occur as the number of agents tend to infinity.

Consider a chain of n agents placed a distance 1 apart on the line where $n = 6k + l$, $0 \leq l \leq 5$. That is to say, the initial configuration is $\boldsymbol{x} = [0, 1, 2, \cdots, n-1]$. The behaviour of these configurations has been studied by Edvin Wedin and Peter Hegarty [9].

They proved that the evolution is in some sense periodic, namely:

i After every fifth time step, a group of three agents disconnect from both ends of the chain, and at the subsequent time step they form a cluster.

ii the final configuration will consist of $2k$ clusters of size 3, and one cluster of size l with opinion $\frac{n+1}{2}$.

iii the freezing time is $5k + \epsilon(l)$, where

$$\epsilon(l) = \begin{cases} l - 1 & \text{if } l = 2,3, \\ l & \text{if } l = 1, \end{cases} \tag{4.1}$$

In Figure 14 one can clearly see what is meant with the behaviour being periodic.

However, this result begs the question, how common is this periodic behaviour? Simulations of agents placed on the line with other inter-agent distances tend to also behave very regularly. It is not easy to prove that the behaviour will be regular. Indeed, it is not even easy to formulate precisely what we mean with it being regular. We will now look at this phenomenon in two dimensions.

4.1 Finite Square Lattices

It is not immediately obvious how to generalise the configuration with equidistant agents on an interval into two dimensions, but we have chosen to study squares. An example of such a configuration is in Figure 15, where we see the evolution of such a configuration of length 7 where each agent initially lies a distance 1 from its closest neighbours.

It is not obvious from this figure if there is any periodicity involved. Instead we will look at a larger configuration, the one in Figure 17, with length 40. There is not much obvious regularity in a square of this size.

There is one thing we can say with some certainty, concerning the behaviour of the main diagonals. Every third time step, two agents on the diagonal will move far from their initial positions

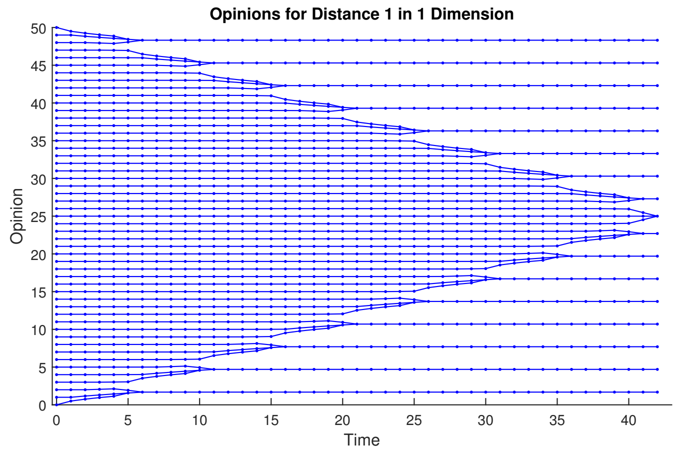

Figure 14: The evolution of a 1-dimensional chain where each agent starts at a distance 1 from its nearest neighbours. One can see that the evolution is periodic.

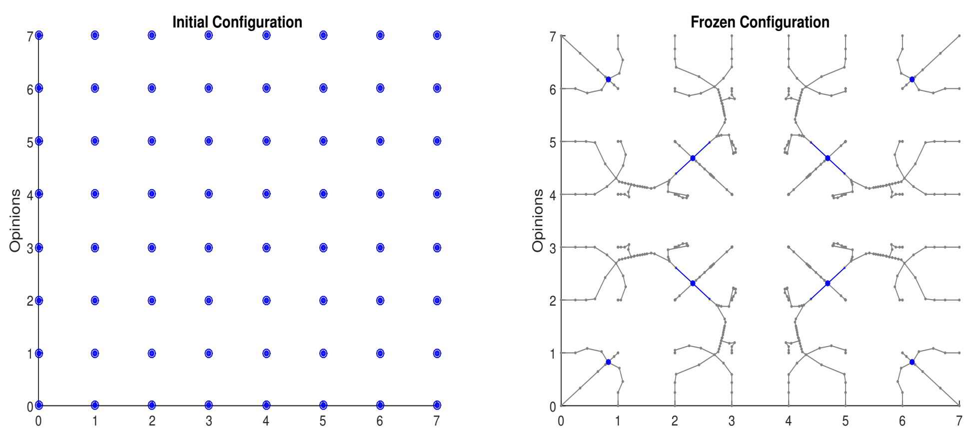

Figure 15: The initial configuration of a square to the left and the evolution to the right. In the figure to the right the blue crosses are the final opinions while gray lines and circles represent the path of the agents and the old opinions, thus one can see the path the agents took. The evolution itself is not trivial.

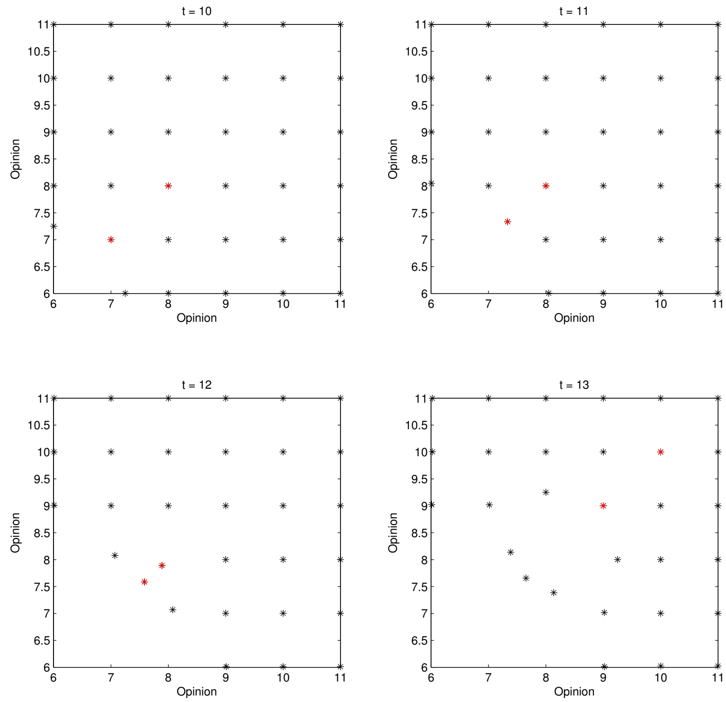

Figure 16: The evolution of a part of the square lattice along the diagonal from the 10:th to the 13:th time step. After the 13:th time step, the two red agents in image 4 are in almost the same situation as the two red agents in the first image. Therefore the evolution should repeat in the same way.

and start forming a cluster with nearby agents, as can be seen in Figure 17 if looking closely. The exception to this is the agents that lie in the corner of the initial configuration, it takes four time steps for them to be perturbed. This behaviour can be seen in more detail in Figure 16. Thus we state the following conjecture.

Conjecture 4.1. *For a square lattice where each agent is placed a distance 1 apart from its nearest neighbours, the freezing time is at least* $\frac{3L+2}{4}$*, where* L *is the width of the lattice.*

Remark 4.1. *This lower bound for the freezing time for the square lattice is at least*$\frac{3}{4}L$*, while for the one-dimensional case the freezing time has been proven to be close to* $\frac{5}{6}L$ *[9]. This means that the freezing time might be smaller for the square of width* L *than for the line of length* L*.*

Saying much more than this is hard, the problem is that these square lattices with inter-agent distances equal to 1 are difficult to simulate correctly. We are so close to the discontinuity in the update-rule that small errors can lead to vastly different configurations.

We have reason to suspect that our simulations are unreliable for larger lattices. Consider a rectangular lattice with inter-agent distances equal to 1, made up out of three columns with N rows each. For n large enough, agents in the middle rows of the lattice will be static vertically so the distance to the closest neighbour above and below an agent will remain exactly 1 for all time steps $t < \frac{n}{2}$. However, the distance ε to the closest neighbour in the same row will eventually be very small. As a result the distance to the closest agent above, in an adjacent column, will be $\sqrt{1^2 + \varepsilon^2}$. After enough time, ε^2 will be on the order of the machine-epsilon, thus, when calculating this distance *MATLAB* will interpret it as exactly 1, even though it should be greater than 1. This means that these rounding errors can change the graph if the agents lie close to distance 1, causing large errors.

We know that this rounding error has a significant effect on the square lattice. As we can see in Figure 17 the sides consist of three rows that have disconnected from the rest of the square. In

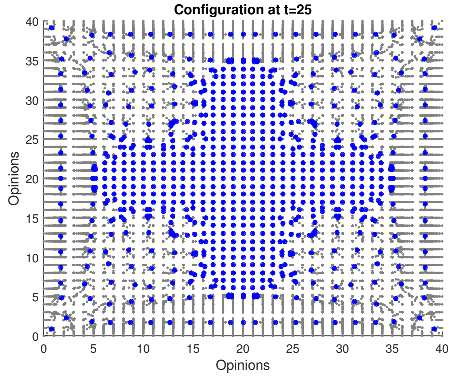

Figure 17: The evolution for a square lattice with initial distance 1. At $t = 25$ (left image) we can no longer see any resemblance to a square, and there is no obvious regularity in the final configuration either (right image).

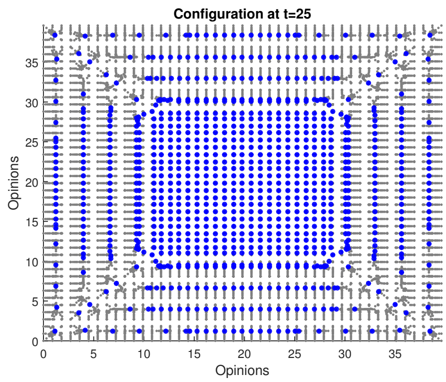

Figure 18: The evolution for a square lattice with initial distance 0.9. At $t = 25$ (left image) the parts in the middle are still shaped like a square, in stark contrast to the configuration in Figure 17. In the final configuration (right image) we can see that the evolution is highly regular. By tracing the grey lines we can see that the evolution along each row or column happened in essentially the same way.

these rows the agents will converge exponentially to the same opinion in one direction. From here on the evolution will be wrong.

We can instead look at lattices where the inter-agent distance is smaller than 1. These lattices are less prone to errors. They also tend to be more regular. In Figure 18 we see an example, where the agents are placed a distance 0.9 apart in a square lattice. There is clearly repetition going on in the evolution almost everywhere.

Analysing the entire square at once is too big a project, so we will study a smaller problem: the edge of an infinitely large square.

4.2 Periodicity for the Half-Plane

Consider an infinite number of agents placed in a half-plane, $\mathbb{Z}_{\geq 0} \times \mathbb{Z}$. The initial distances between neighbouring agents are 1.

It is obvious due to symmetry that displacement will only occur horizontally. Further, while there is no vertical displacement, all displacement in the horizontal direction will be identical for each row. Thus, every agent will have two others, one above and one below with the same horizontal opinion as itself at distance exactly one.

Because the agents always remain at a distance 1 vertically, an agent can only see another in a different row if they have the exact same opinion horizontally. And since clustering happens only when two or more agents have the same set of neighbours, clusters will never form. Therefore, every agent will at all times see exactly two others with the same horizontal opinion as itself. As a further corollary to this, the freezing-time is no longer finite, since no two agents will ever see the same average.

The update rule is therefore equivalent to a one-dimensional model, where, in every step, each agent's own opinion is weighted by an additional two. This update-rule can be written as

$$x_{t+1}(i) = \frac{1}{2 + |\mathcal{N}_t(i)|}\Big[2x_t(i) + \sum_{j\in\mathcal{N}_t(i)} x_t(j)\Big] \tag{4.2}$$

We will go on to prove that for this model the behaviour of the configuration $[0, 1, 2, \cdots]$ is eventually periodic. In order to get there, we will need to develop some theory regarding the evolution of the distances between consecutive agents.

4.2.1 Distances Between Agents

What we want to do here is to prove Corollary 4.1, which tells us when the evolution is periodic. In order to prove it we will need some properties regarding the dynamics in what we call distance-space. In distance-space, the configuration is described by a vector that contains the distances between consecutive agents. It is a useful description of the configuration in one-dimensional models such as this one.

In particular we will need to prove that the transition matrix $\mathbf{B}_t$ in distance-space is non-negative, just like the $\mathbf{H}_t$-matrix in opinion-space.

The vector of distances between consecutive agents for a configuration $\boldsymbol{x}_t$ can be written as

$$\boldsymbol{y}_t = \begin{pmatrix} x_t(2) - x_t(1) \\ x_t(3) - x_t(2) \\ \vdots \end{pmatrix} = \Delta \boldsymbol{x}_t \tag{4.3}$$

where

$$\Delta = \begin{pmatrix} -1 & 1 & 0 & \cdots \\ 0 & -1 & 1 & \cdots \\ 0 & 0 & -1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{4.4}$$

Δ is not an invertible matrix. However, if we apply an additional condition that $x(1) = 0$ there exists a right[4] inverse matrix,

$$\Delta_0^{-1} = \begin{pmatrix} 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ 1 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{4.5}$$

In the more general case where we demand $x(1)$ be any real number the right inverse is

$$\boldsymbol{x}_t = \Delta^{-1}(\boldsymbol{y}_t) = x_t(1) \cdot \mathbf{1}_\infty + \Delta_0^{-1}\boldsymbol{y}_t, \tag{4.6}$$

where $\mathbf{1}_\infty$ is a vector with all components 1.

Using these two matrices we can calculate the operator that evolves a distance-vector. Since the dynamics are translationally invariant we can wlog assume that $x_t(1) = 0$ to get

$$\boldsymbol{x}_{t+1} = \mathbf{H}_t \boldsymbol{x}_t \tag{4.7}$$

$$\Rightarrow \Delta\boldsymbol{x}_{t+1} = \Delta\mathbf{H}_t \boldsymbol{x}_t \tag{4.8}$$

$$\Rightarrow \boldsymbol{y}_{t+1} = \Delta\mathbf{H}_t\Delta_0^{-1} \boldsymbol{y}_t \tag{4.9}$$

$$= \mathbf{B}_t \boldsymbol{y}_t, \tag{4.10}$$

where we define

$$\mathbf{B}_t = \Delta\mathbf{H}_t\Delta_0^{-1}. \tag{4.11}$$

[4] We have $\Delta\Delta_0^{-1} = I$ but not $\Delta_0^{-1}\Delta = I$. This last product instead equals S_{-1}. The reason for this is that with finitely many, N, agents Δ is an $(N-1)\times N$ matrix, so it is not "truly" square.

This is good, now we know $\mathbf{B}_t$ if we know $\mathbf{H}_t$. What we want is $\mathbf{B}_t$ based only on the graph of the configuration. We can rewrite

$$\Delta = \mathbf{S}_1 - \mathbf{I}, \tag{4.12}$$

where $\mathbf{S}_1$ is a shift matrix as defined in appendix A, in order to get

$$(\Delta \mathbf{H}_t)_{ij} = (\mathbf{S}_1 \mathbf{H}_t - \mathbf{H}_t)_{ij} \tag{4.13}$$
$$= (\mathbf{H}_t)_{(i+1)j} - (\mathbf{H}_t)_{ij}. \tag{4.14}$$

We can also write

$$\Delta_0^{-1} = \sum_{k=1}^{\infty} \mathbf{S}_{-k} \tag{4.15}$$

to get

$$(\Delta \mathbf{H}_t \Delta_0^{-1})_{ij} = \sum_{k=1}^{\infty} \left((\mathbf{H}_t)_{(i+1)j} - (\mathbf{H}_t)_{ij} \right) \mathbf{S}_{-k} \tag{4.16}$$
$$= \sum_{k=1}^{\infty} \left((\mathbf{H}_t)_{(i+1)(j+k)} - (\mathbf{H}_t)_{i(j+k)} \right) \tag{4.17}$$
$$= \sum_{k=1}^{\infty} (\mathbf{H}_t)_{(i+1)(j+k)} - \sum_{k=1}^{\infty} (\mathbf{H}_t)_{i(j+k)} \tag{4.18}$$

From the definition of $\mathbf{H}_t$ we can interpret the first of these sums as the proportion of neighbours of $i+1$ to the right of agent j, and the second sum as the proportion of neighbours of i to the right of agent j. Therefore we can rewrite the sums as

$$(\mathbf{B}_t)_{ij} = \frac{n_{i+1,j}}{N_{i+1}} - \frac{n_{i,j}}{N_i} = \frac{n_{i+1,j}}{n_{i+1,j} + \tilde{n}_{i+1,j}} - \frac{n_{i,j}}{n_{i,j} + \tilde{n}_{i,j}}, \tag{4.19}$$

where $n_{x,y}$ is the number of agents agent x sees to the right of agent y, not including agent y, N_x is the total number of agents agent x sees and $\tilde{n}_{x,y}$ is the number of agents agent x sees to the left of agent y, including agent y.

Now we claim that the first of these terms is at least as great as the other.

Proposition 4.1. $\mathbf{B}_t$ *only has non-negative entries.*

Proof. It is obvious that for any i,j, we have $\tilde{n}_{i+1,j} \leq \tilde{n}_{i,j}$ and $n_{i+1,j} \geq n_{i,j}$. This together with (4.19) gives us

$$(B_t)_{ij} = \frac{n_{i+1,j}}{n_{i+1,j} + \tilde{n}_{i+1,j}} - \frac{n_{i,j}}{n_{i,j} + \tilde{n}_{i,j}} \tag{4.20}$$
$$= \frac{n_{i+1,j}\tilde{n}_{i,j} - n_{i,j}\tilde{n}_{i+1,j}}{(n_{i+1,j} + \tilde{n}_{i+1,j})(n_{i,j} + \tilde{n}_{i,j})} \geq 0 \tag{4.21}$$

□

Remark 4.2. *It follows directly that agents will never change order. Since* $\mathbf{B}_t$ *only has positive entries, we can never get negative entries in* $\boldsymbol{y}_t$ *if we start with only positive ones.*

Remark 4.3. $\mathbf{B}_t$ *will have only positive entries in any variation of the HK-model of the form*

$$x_{t+1}(i) = \frac{w x_t(i) + \sum_{j \in \mathcal{N}_t(i)} x_t(j)}{|\mathcal{N}_t(i)| + w} \tag{4.22}$$

as long as $w \geq 0$*. We will come back to models of this form in Section 4.3.*

We are now ready to state and prove the following theorem.

Theorem 4.1. *If we have three initial configurations* $\boldsymbol{a}_0$*,* $\boldsymbol{x}_0$ *and* $\boldsymbol{b}_0$ *such that*

$$\Delta \boldsymbol{a}_0 \leq \Delta \boldsymbol{x}_0 \leq \Delta \boldsymbol{b}_0, \tag{4.23}$$

and the transition matrices (and therefore graphs) for $\boldsymbol{a}_t$ *and* $\boldsymbol{b}_t$ *are equal at all time steps up to some time* τ*, then* $\boldsymbol{x}_t$ *also has the same matrices (and graphs) as* $\boldsymbol{a}_t$ *and* $\boldsymbol{b}_t$ *up to time* τ*.*

Remark 4.4. *The ordering we're using is $\boldsymbol{a} < \boldsymbol{b}$ if and only if for all i, $a(i) < b(i)$.*

Proof. Suppose we know that at some time t, $\Delta \boldsymbol{a}_t \leq \Delta \boldsymbol{x}_t \leq \Delta \boldsymbol{b}_t$. It is then true that any agents that are not neighbours in configuration $\boldsymbol{a}_t$ are also not neighbours in $\boldsymbol{x}_t$. By the same token, any agents that are neighbours in $\boldsymbol{b}_t$ are also neighbours in $\boldsymbol{x}_t$. Since a_t and b_t are assumed to have the same graphs, this means that x_t also has the same graph.

Since by proposition 4.1 $\mathbf{B}_t$ only has positive entries the inequalities are preserved in the next time-step. By induction in t up to τ the proof is finished. □

Corollary 4.1.1. *Suppose we have three configurations $\boldsymbol{a}_0, \boldsymbol{x}_0, \boldsymbol{b}_0$ as above but also know that there exists a k such that the first k elements each of $\boldsymbol{a}_0$,$\boldsymbol{b}_0$ become disconnected from the rest of the agents at time τ. Suppose further that we have for all m*

$$\Delta \boldsymbol{a}_0 \leq \mathbf{T}^m \Delta \boldsymbol{x}_0 \leq \Delta \boldsymbol{b}_0, \tag{4.24}$$

where we define

$$\mathbf{T} = \mathbf{S}_k \mathbf{B}(\boldsymbol{a}_{\tau-1}) \mathbf{B}(\boldsymbol{a}_{\tau-2}) \cdots \mathbf{B}(\boldsymbol{a}_1) \mathbf{B}(\boldsymbol{a}_0) \tag{4.25}$$

and $\mathbf{B}(\boldsymbol{x})$ is the transition matrix in distance-space for the configuration $\boldsymbol{x}$.

Then we know, writing $t = l\tau + t'$, with $0 \leq t' < \tau$, that

$$\mathbf{B}(\mathbf{S}_{lk} \boldsymbol{x}_t) = \mathbf{B}(\boldsymbol{x}_{t'}) \tag{4.26}$$

Proof. Follows directly from Theorem 4.1 by induction. □

In essence Theorem 4.1 states that if we have two configurations that behave the same way, and a configuration that lies between them in distance space, then the middle configuration will also behave in the same way. The corollary states that if the inner part of this middle configuration looks similar enough to the original configuration (and the outer part has disconnected) then this inner part will behave the same way as the original configuration, that is, the matrices describing the transitions will be the same, and the evolution is periodic.

4.2.2 Proof of Periodicity

Theorem 4.2. *The evolution of the weighted version* (4.2) *of the HK-update rule with initial distance vector $\boldsymbol{z_0} = \mathbf{1}_\infty$ is periodic from $t = 10$ and onwards, with period $\tau = 21$. At $t = 10 + l\tau, l \in \mathbb{N}$, the four outermost agents will be disconnected from all other agents.*

What we need to prove periodicity is upper and lower bounds on the distance vector over time. These bounds need to be tight enough for Corollary 4.1 to be applicable.

We start by calculating the transition matrices and the upper bound. Thereafter we prove a proposition that gives us a lower bound. These combined are enough to finally prove this theorem.

Proposition 4.2. *At $t = 31$, the distance vector of the still connected agents is $\boldsymbol{y_0} = \mathbf{T}(\mathbf{S_3 B_1^1 B_0^9 1_\infty})$, with $\mathbf{T}$, $\mathbf{B_0}$ and $\mathbf{B_1}$ as below.*

Proof. The transition matrix $\mathbf{B}_0$ for the original configuration is

$$\mathbf{B_0} = \begin{bmatrix} \frac{11}{20} & \frac{1}{5} & 0 & \cdots & & & \\ \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & 0 & \cdots & & \\ 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & 0 & \cdots & \\ 0 & 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & \cdots & \\ \vdots & & & & \ddots & & \end{bmatrix}.$$

Direct calculation shows that this is the same for each of the first 9 time steps. Thus, $\boldsymbol{z_9} = \mathbf{B}_0^9 \boldsymbol{z_0}$.

At $t = 9$, agent 0 can see agent 2. For this time-step, we instead get the transition matrix

$$\mathbf{B_1} = \begin{bmatrix} \frac{2}{5} & 0 & 0 & \cdots & & & \\ \frac{1}{30} & \frac{7}{15} & \frac{1}{6} & 0 & \cdots & & \\ \frac{1}{6} & \frac{1}{3} & \frac{19}{30} & \frac{1}{5} & 0 & \cdots & \\ 0 & 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & \cdots & \\ \vdots & & & & \ddots & & \end{bmatrix}$$

After one more time step, agents 0, 1 and 2 are disconnected from the rest and we disregard these to only consider the distance-vector from agent 3 and beyond. This is done by multiplying $\boldsymbol{z}_{10}$ by $\mathbf{S_3}$ and we have $\hat{\boldsymbol{z}}_{10} = \mathbf{S_3}\boldsymbol{z}_{10}$.

The following 21 time steps describe the evolution that will reoccur every period. So we are now deriving the total transition matrix $\mathbf{T}$.

The transition matrix for the remaining agents during each of the next 9 time-steps is again $\mathbf{B_0}$. At $t = 19$, agents 3 and 5 see each other for the first time. This yields the same transition matrix as for $t = 10$, $\mathbf{B_1}$, which is valid during the next 11 time steps. At $t = 30$, agents 3 and 4 will both see agent 6 and we have a new transition matrix,

$$\mathbf{B_2} = \begin{bmatrix} \frac{1}{3} & 0 & 0 & \dots & & & & \\ 0 & \frac{1}{3} & 0 & 0 & \dots & & & \\ \frac{1}{42} & \frac{1}{21} & \frac{17}{42} & \frac{1}{7} & 0 & \dots & & \\ \frac{1}{7} & \frac{2}{7} & \frac{3}{7} & \frac{23}{35} & \frac{1}{5} & 0 & \dots & \\ 0 & 0 & 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} & 0 & \dots \\ \vdots & & & & & & \ddots & \end{bmatrix},$$

valid during one time-step.

In the subsequent time step, at time $t = 31$, agent 6 can no longer see agent 7, thus agents 3 to 6 are disconnected from the rest of the agents. To disconnect agents 3 to 6 we multiply $\boldsymbol{z}_{31}$ by $\mathbf{S_4}$ and we have

$$\hat{\boldsymbol{z}}_{31} = \mathbf{S_4}\mathbf{B_2}\mathbf{B_1}^{11}\mathbf{B_0}^{9}\hat{\boldsymbol{z}}_{10}. \tag{4.27}$$

In our proof we will let $\boldsymbol{y_0} = \hat{\boldsymbol{z}}_{31}$ be the initial distance vector. We will also use the total transition matrix defined as

$$\mathbf{T} = \mathbf{S_4}\mathbf{B_2}\mathbf{B_1}^{11}\mathbf{B_0}^{9}. \tag{4.28}$$

□

Remark 4.5. *The curious reader finds the initial components of the vector $\boldsymbol{y_0}$, the row sums of $\mathbf{T}$ and the repeating sequence of all non-zero elements at each row in $\mathbf{T}$, starting at row 18, in appendix C. Noteworthy is that all row sums are less than or equal to one.*

Proposition 4.3. *For $\boldsymbol{y_0}$ as above, and $\boldsymbol{y}_m = \mathbf{T}^m\boldsymbol{y}_0$, for all $m \in \mathbb{N}$, we have the following inequalities:*

$$0 < y_0(i) - y_m(i) < c(i), i \leq 40 \tag{4.29}$$

$$0 < y_0(i) - y_m(i) < a\gamma^i, i > 40, \tag{4.30}$$

where $\gamma = 0.6$, $a = \frac{1}{1000}$, and $\boldsymbol{c}$ is the solution to the following minimisation problem:

$$\begin{aligned} \underset{c(i)}{\text{minimise}} \quad & z = \sum_{i=1}^{200} c(i) \\ \text{s.t.} \quad & (\tilde{\mathbf{T}} - \tilde{\mathbf{I}})\boldsymbol{c} \leq \boldsymbol{y}_1 - \boldsymbol{y}_0 \\ & c(i) > 0, \forall i. \end{aligned} \tag{4.31}$$

Here

i) $\tilde{\mathbf{T}}$ *is the first* 200 *rows of* $\mathbf{T}$ *including all non-zero columns:* $\tilde{\mathbf{T}} \in \mathbb{R}^{200\times(200+25)}$.

ii) $\tilde{\mathbf{I}} \in \mathbb{R}^{200\times(200+25)}$*, where the first* 200×200 *rows and columns are the identity matrix, and the last* 25 *are zero-columns.*

Proof. Concerning the left-hand inequalities, assume that $\mathbf{T}\boldsymbol{y_m} \leq \boldsymbol{y_m}$. We know that $\mathbf{T}$ is a non-negative matrix, since it by Proposition 4.1 and 4.2 is a product of non-negative matrices. Thus we have $\mathbf{T}\boldsymbol{y_{m+1}} = \mathbf{T}(\mathbf{T}\boldsymbol{y_m}) \leq \mathbf{T}\boldsymbol{y_m} = \boldsymbol{y_{m+1}}$. Lastly we have that $\boldsymbol{y_1} = \mathbf{T}\boldsymbol{y_0} \leq \boldsymbol{y_0}$. For the small indices we can verify this by direct multiplication. For the larger indices we can use that $y(i) = 1$

for all $i >> 0$ and that the row sums of $\mathbf{T}$ are at most 1. By induction, $\boldsymbol{y_{m+1}} \leq \boldsymbol{y_m}$, and therefore $\boldsymbol{y_{m+1}} \leq \boldsymbol{y_0}$.

Regarding the right-hand inequalities, suppose that

$$\boldsymbol{y_0} - \boldsymbol{y_m} < \boldsymbol{\tilde{c}} \tag{4.32}$$

for some m. We have

$$\begin{aligned} \boldsymbol{y_0} - \boldsymbol{y_{m+1}} &= \boldsymbol{y_0} - \mathbf{T}\boldsymbol{y_m} && (4.33)\\ &= \boldsymbol{y_0} - \mathbf{T}\boldsymbol{y_0} + \mathbf{T}(\boldsymbol{y_0} - \boldsymbol{y_m}) && (4.34)\\ &= \boldsymbol{y_0} - \boldsymbol{y_1} + \mathbf{T}(\boldsymbol{y_0} - \boldsymbol{y_m}). && (4.35) \end{aligned}$$

Then, inequality (4.32) implies that $\boldsymbol{y_0} - \boldsymbol{y_{m+1}} < \boldsymbol{y_0} - \boldsymbol{y_1} + \mathbf{T}\boldsymbol{\tilde{c}}$. If, for the same vector $\boldsymbol{c}$ we have

$$\boldsymbol{y_0} - \boldsymbol{y_1} + \mathbf{T}\boldsymbol{\tilde{c}} \leq \boldsymbol{\tilde{c}}, \tag{4.36}$$

then, by induction, (4.32) holds for all m if it holds for the base case $\boldsymbol{y_0} - \boldsymbol{y_1} < \boldsymbol{\tilde{c}}$. One such $\boldsymbol{\tilde{c}}$, fulfilling (4.36) while being "fairly tight" is the solution to the linear minimisation problem

$$\begin{aligned} \underset{\tilde{c}(i)}{\text{minimise}} \quad & z = \sum_{i=1}^{\infty} \tilde{c}(i) \\ \text{s.t.} \quad & (\mathbf{T} - \mathbf{I})\boldsymbol{\tilde{c}} \leq \boldsymbol{y_1} - \boldsymbol{y_0} \\ & \tilde{c}(i) > 0, \forall i. \end{aligned} \tag{4.37}$$

To avoid solving an infinite system of equations it is sufficient to find any $\boldsymbol{\tilde{c}}$ from the solution space of (4.37). Finding such a $\boldsymbol{\tilde{c}}$ can be done by solving the partial problem (4.31) and combining parts of the yielded vector with an infinite vector $\boldsymbol{\gamma}$ with elements defined as $\gamma(i) = a\gamma^{40+i}, i = 1,2,\ldots$, that is

$$\boldsymbol{\tilde{c}} = [c(1),c(2),\ldots,c(39),c(40),\gamma(1),\gamma(2),\ldots]. \tag{4.38}$$

We need to motivate that $\boldsymbol{\tilde{c}}$ is in the solution space of (4.36), namely that

$$y_0(i) - y_1(i) + \boldsymbol{T_i}\boldsymbol{\tilde{c}} \leq \tilde{c}(i), \ \forall i. \tag{4.39}$$

We divide the problem into three cases:

1. i such that $\boldsymbol{T_i}\boldsymbol{\tilde{c}}$ and $\tilde{c}(i)$ only includes elements from (4.31), $i \leq 15$.
2. i such that $\boldsymbol{T_i}\boldsymbol{\tilde{c}}$ and $\tilde{c}(i)$ includes values from both the solution of (4.31) and $\boldsymbol{\gamma}$, $i \in [16,57]$.
3. i such that $\boldsymbol{T_i}\boldsymbol{\tilde{c}}$ and $\tilde{c}(i)$ only includes elements from $\boldsymbol{\gamma}$, $i \geq 58$.

The first case already satisfies (4.39). The second case is confirmed to satisfy (4.39) by calculations in *MATLAB*. When checking the last case we observe that $y_0(i) - y_1(i) = 0, i \geq 27$, so (4.39) is reduced to $\boldsymbol{T_i}\boldsymbol{\tilde{c}} < \tilde{c}(i)$. We subtract $\tilde{c}(i)$ from both sides and expand to get

$$\boldsymbol{T_i}\boldsymbol{\tilde{c}} - \tilde{c}(i) = \sum_{j=i-17}^{i+25} T(i,j)\tilde{c}(j) - \tilde{c}(i) = \sum_{j=i-17}^{i+25} T(i,j)a\gamma^j - a\gamma^i. \tag{4.40}$$

The sum is finite due to the fact that most of the elements in $\mathbf{T}$ are zero. We realise that

$$\sum_{j=i-17}^{i+25} T(i,j)a\gamma^j - a\gamma^i = a\Big(\sum_{j=-17}^{25} T(i,i+j)\gamma^{i+j} - \gamma^i\Big) = a\gamma^i \underbrace{\Big(\sum_{j=-17}^{25} T(i,i+j)\gamma^j - 1\Big)}_{\approx -0.614,\ \forall i}. \tag{4.41}$$

Hence all three cases satisfy (4.39) and $\boldsymbol{\tilde{c}}$ is in the solution space of (4.37). Checking that $\boldsymbol{y_0} - \boldsymbol{y_1} < \boldsymbol{\tilde{c}}$ holds and that the transition matrix describes the same behaviour for a displaced vector $\boldsymbol{y_0} - \boldsymbol{\tilde{c}}$ completes the induction step. □

Proof of Theorem 4.2. Proposition 4.3 gives us an upper and lower bound that satisfy the requirements of Corollary 4.1.1. Thus, we have periodicity. □

The half-plane lattice updates according to (4.2) and therefore the evolution of the half-plane is also periodic.

4.3 Discussion and Continuous Time Model

In (4.2) we stated the update rule for a HK-model where each agent's own opinion is weighted by an additional two. We will now look at different generalisations of this, and derive a related HK-model with continuous time.

Consider a configuration $\mathbb{Z}_{\geq 0} \times \mathbb{Z}^d$. Then we obtain an update rule according to

$$x_{t+1}(i) = \frac{1}{2d + |\mathcal{N}_t(i)|} \left[2d x_t(i) + \sum_{j \in \mathcal{N}_t(i)} x_t(j) \right]. \tag{4.42}$$

It is clear that this formula would work for any value of d, not only integers. Thus, we can consider the rule above for any $w = 2d \in \mathbb{R}$.

$$x_{t+1}(i) = \frac{1}{w + |\mathcal{N}_t(i)|} \left[w x_t(i) + \sum_{j \in \mathcal{N}_t(i)} x_t(j) \right], \tag{4.43}$$

where we will assume $w \geq 0$.

In this model, each agent will move toward the average opinion of the agents it sees, just like in the standard model. The main difference is that it will now do so much more slowly as w increases. It is therefore reasonable to make the model take smaller time steps, we will use time steps of $\mathrm{d}t = \frac{1}{w}$. Thus we arrive at what we will call the HK_w model.

$$x_{t+1/w}(i) = \frac{1}{w + |\mathcal{N}_t(i)|} \left[w x_t(i) + \sum_{j \in \mathcal{N}_t(i)} x_t(j) \right]. \tag{4.44}$$

Now, consider the expression

$$\frac{x_{t+1/w}(i) - x_t(i)}{1/w} = w \left[\frac{1}{w + |\mathcal{N}_t(i)|} \left(w x_t(i) + \sum_{j \in \mathcal{N}_t(i)} x_t(j) - (w + |\mathcal{N}_t(i)|) x_t(i) \right) \right] \tag{4.45}$$

$$= \frac{w}{w + |\mathcal{N}_t(i)|} \sum_{j \in \mathcal{N}_t(i)} (x_t(j) - x_t(i)). \tag{4.46}$$

If we treat this as a continuous function of time we might be tempted to take the limit and arrive at the following set of coupled differential equations, where we now move the vector index to a subscript and time as a parameter,

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \sum_{j \in \mathcal{N}_i(t)} (x_j(t) - x_i(t)). \tag{4.47}$$

One must tread carefully, however, for it is not quite clear how we should interpret this limit, since $\boldsymbol{x}$ is not usually interpreted as an actual function of a continuous time, and the right hand side is not continuous for all times!

Instead, we can consider the integral equations

$$x_i(t) = x_i(0) + \int_0^t \sum_{j \in \mathcal{N}_i(\tau)} (x_j(\tau) - x_i(\tau)) \, \mathrm{d}\tau, \tag{4.48}$$

a solution of which is a so-called Carathéodory solution to (4.47). These equations have been studied by Blondel et al [10], and they have shown that a unique solution exists for almost all initial configurations $\boldsymbol{x}_0$.

A lot of things are still unclear. For instance, what, if anything, can we say about the behaviour of a configuration in the HK_w model given the behaviour of the configuration in the continuous time model?

5 In Conclusion

We have introduced the concept of critical volume and investigated the concept in two dimensions, as well as provided support for a conjecture in one dimension. We found that the disc probably has no critical area, and that there are major differences in the behaviour of the evolution between squares and rectangles as well as other polygons.

We have developed theory for distance-space and proved some basic theorems. We also proved that agents spread on an integer half-plane will eventually have a periodic evolution.

Finally, we showed that by imagining the HK-model in an infinite number of dimensions we get something that probably behaves like a HK-model in continuous time.

A The S_k Matrices

In this thesis, particularily section 4, we make repeated use of matrices denoted $\mathbf{S}_k$. They are defined as follows (with i,$j \geq 0$);

$$(\mathbf{S}_k)_{ij} = \delta_{i+k,j} = \begin{cases} 1 & \text{if } i+k=j \\ 0 & \text{otherwise.} \end{cases} \tag{A.1}$$

As an example,

$$\mathbf{S}_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}. \tag{A.2}$$

If k is positive they strip the first k elements of a vector, and if k is negative they prepend zeroes in the k first places. For $k = 0$ we get the identity matrix.

When acting upon matrices the behaviour still depends on the sign of k. If k is positive and we multiply with S_k from the right the matrix is shifted k columns to the right, with zeroes in the "new" components. If we multiply from the left the matrix instead gets shifted up, removing the first k rows. If k is negative we get the opposite effect; when multiplying from the right we shift the matrix to the left, removing the first k columns. When multiplying from the left it shifts down, adding k rows containing only 0.

B Matlab Files

Below are the files referenced in the introduction and used throughout for most numerical calculations. They can be used to simulate and visualise a lot of the simpler problems in the HK-model. No license, do with them as you wish.

B.1 HK_example.m

The following file is a full-fledged example, that creates a random configuration, calculates the evolution and plots the result. This file requires `HK_solver.m`, `HK_plot.m` and `HK_opts.m`.

```
% A sample script for using the functions HK_solver, HK_plot and HK_opts.
% Uses a configuration of an N-cube of length L with uniform random
% opinions.

L = 5;
agents = 100;
dimensions = 3;
% Generates initial agents.
Initial_Ops = L*rand(agents,dimensions);

% Default options for the initial configuration. Since we don't change any
% options this is technically unneccesary. We don't need to supply it to
% HK_solver or HK_plot in this case.
opts = HK_opts(Initial_Ops);
% Solves for all times.
results = HK_solver(Initial_Ops, opts);

% Plot the results.
HK_plot(results, opts)

% Plots the results, and prints the clustersize next to each cluster when
% it's done.
% HK_plot(results, opts, results.allweights{end});

% Instead prints when the cluster froze
%HK_plot(results, opts, results.indiv_freeze_times)
% Prints nothing
```

B.2 HK_solver.m

This is the file that finds the opinions at all times for a given initial configuration. Let us first look at the file in its entirety and then look at some of the more complex parts.

```matlab
function [ results ] = HK_solver(Initial_Ops, Opts)
  % Finds the opinion vectors for all times up to the freezing time or
  % Opts.tmax, whichever comes first, starting at Initial_Ops.
  % Initial_Ops should be an N x d matrix, where N is the number of
  % agents and d is the number of dimensions. Opts is an optionl argument
  % containing options, it is a struct, as a default use Opts=HK_opts();

  if ischar(Initial_Ops)
    % Loads data from the given file instead of starting anew. Most
    % commonly this would be in order to restore a simulation.
    load(Initial_Ops);
  else
    % Initial data
    if nargin == 1
      Opts = HK_opts;
    end
    [clusters,dimensions] = size(Initial_Ops);
    results.Ops{1} = Initial_Ops;
    t = 0;

    results.pre_cluster_indices = {};
    results.post_cluster_indices = {};

    results.indiv_freeze_times = zeros(clusters,1);
    weights = ones(clusters,1);
    epsilon = Opts.epsilon; % Extracts epsilon for the parfor loop

    % Timers
    print_iteration = tic;
    save_progress_timer = tic;
  end

  while t < Opts.tmax
    % Clusterises the opinion matrix/vector
    [results.Ops{t+1}, ia, ic] = unique(results.Ops{t+1},'rows');
    % New number of agents/clusters
    [clusters, ~] = size(results.Ops{t+1});
    % Save indices so that the unclustered opinion vector can be
    % restored
    results.pre_cluster_indices{t+1}  = ic;
    results.post_cluster_indices{t+1} = ia;

    % Calculate new weights.
    preweights = weights;
    weights = zeros(clusters,1);
    for c = 1:clusters
      % Sum of all weights that clustered to cluster c
      weights(c) = sum(preweights(ic == c));
    end
    results.allweights{t+1} = weights;

    few_agents = clusters < Opts.large_num_agents;

    Old_Ops = results.Ops{t+1};
    [New_Ops] = HK_iteration(Old_Ops);

    % Extracts the freeze times of the still existing clusters
    results.indiv_freeze_times = results.indiv_freeze_times(ia);

    % If cluster didn't move, do nothing. If it did, add current time
    % and subtract old freezing time.
    moved = sum((New_Ops-Old_Ops).^2, 2) > Opts.epsilon;
    current_times = t*ones(clusters,1);
    results.indiv_freeze_times = results.indiv_freeze_times ...
                               + moved.*( current_times ...
```

```
                                              - results.indiv_freeze_times );


    % The matrices aren't exactly equal even if we are frozen due to
    % floating point arithmetic, and I want control over this rather
    % than Matlab
    if (length(New_Ops) == length(Old_Ops)) ...
    && (norm(New_Ops-Old_Ops) < Opts.maxnorm)
      break;
    else % If we didn't freeze, store the result and go on.
      results.Ops{(t+1)+1} = New_Ops;
      t = t + 1;
    end

    if Opts.verbose && toc(print_iteration) > Opts.print_progress_delay
      fprintf('Iteration %d, remaining clusters, %d.\n', ...
              t, clusters);
      print_iteration = tic; % resets timer
    end
    if toc(save_progress_timer) > Opts.save_progress_delay
      if Opts.verbose, fprintf('Saving progress..'), end;
      save(Opts.save_progress_path);
      if Opts.verbose, fprintf('! Done!\n'), end;
      save_progress_timer = tic; %resets timer
    end
  end

  function [ newops ] = HK_iteration( Ops )
    newops = zeros(clusters,dimensions);
    if few_agents
      for i = 1:clusters
        Own_Ops = repmat(Ops(i,:),clusters,1);
        dists = sum((Own_Ops-Ops).^2,2);
        N = dists <= 1 + epsilon;
        newops(i,:) = (weights.*N)'*Ops/(N'*weights);
      end
    else
      parfor i = 1:clusters;
        Own_Ops = repmat(Ops(i,:),clusters,1);
        dists = sum((Own_Ops-Ops).^2,2);
        N = dists <= 1 + epsilon;
        newops(i,:) = (weights.*N)'*Ops/(N'*weights);
      end
    end
  end
end
```

The solver will begin setting up options if it hasn't received any, or load in old data if it is given a file to load from. It then enters the main loop, which will continue until the configuration freezes or $t > t_{\max}$, where $t_{\max}$ is given by the options, and is usually infinite.

The first step in the main loop is "clusterising" the opinion matrix. This means taking all agents that have the exact same opinions and storing them as a single agent, but with a different weight. This is done by the following lines.

```
% Clusterizes the opinion matrix/vector
[results.Ops{t+1}, ia, ic] = unique(results.Ops{t+1},'rows');
% New number of agents/clusters
[clusters, ~] = size(results.Ops{t+1});
% Save indices so that the unclustered opinion vector can be
% restored
results.pre_cluster_indices{t+1}  = ic;
results.post_cluster_indices{t+1} = ia;

% Calculate new weights.
preweights = weights;
weights = zeros(clusters,1);
for c = 1:clusters
  % Sum of all weights that clustered to cluster c
  weights(c) = sum(preweights(ic == c));
```

```
end
results.allweights{t+1} = weights;
```

The first block does the actual clusterisation, and stores the results. The second block calculates the new weights. The vector `ic` contains the indices in the new matrix corresponding to indices in the old matrix. That is, if `[C,ia,ic]=unique(A)`, then `A=C(ic,:)`. Thus, what the second block does is simply adding the weights of the clusters that now formed.

The purpose of doing this is to lower the amount of agents that have to be simulated. This is crucial in being able to simulate large dense configurations.

Updating the opinion matrix itself is done by the loop

```
for i = 1:clusters
  Own_Ops = repmat(Ops(i,:),clusters,1);
  dists = sum((Own_Ops-Ops).^2,2);
  N = dists <= 1 + epsilon;
  newops(i,:) = (weights.*N)'*Ops/(N'*weights);
end
```

The first row creates a matrix where all the rows are the opinion of agent `i`. Thus, the vector `dists` contains the distance from agent `i` to all other agents, and `N` is a vector with `N(j)` equal to 1 if `i` and `j` are neighbours. The last row is thus essentially

$$x_{t+1}(i) = \frac{\begin{pmatrix} w(1)N(1) & w(2)N(2) & \cdots & w(c)N(c) \end{pmatrix} \begin{pmatrix} x_t(1) \\ x_t(2) \\ \vdots \\ x_t(c) \end{pmatrix}}{\begin{pmatrix} N(1) & N(2) & \cdots & N(c) \end{pmatrix} \begin{pmatrix} w(1) \\ w(2) \\ \vdots \\ w(c) \end{pmatrix}}. \tag{B.1}$$

B.3 HK_plot.m

This file plots the results from `HK_solver.m` in a reasonable manner.

```
function [ ] = HK_plot( res,  opts, final_data)
  [clusters,dimensions] = size(res.Ops{1});

  if nargin < 2 % Gets default options if none are provided
    opts = HK_opts(res.Ops{1});
  end % Sets final_data if it is not provided.
  if nargin < 3
    final_data = [];
  end

  % If final_data is not empty we fix it up and say that it should be
  % plotted.
  if ~isempty(final_data)
    draw_str_data = true;
    if isnumeric(final_data)
        final_data = cellstr(num2str(final_data));
    end
  else
    draw_str_data = false;
  end

  % Sets stopping time.
  if ~isfinite(opts.tmax)
    tmax = length(res.Ops)-1;
  else
    tmax = opts.tmax;
  end

```

```
% Used for axis limits. The minima and maxima of the starting
% configuration are never passed at any time so they serve as good
% boundaries.
mins = min(res.Ops{1});
maxs = max(res.Ops{1});

fig = figure; hold on;
if dimensions == 1
  xlabel('Time'), ylabel('Opinion');
elseif dimensions == 3
  view(3);
elseif dimensions > 3
  error('Cant plot in more than 3 dimensions');
end

circs = [];
lines = [];

t = 0;
while true
  [clusters,~] = size(res.Ops{t+1});
  switch dimensions
    case 1
      circs = plot(t, res.Ops{t+1}, ...
                   opts.opinion_marker, ...
                   'Color', opts.active_color);
      if t > 0 && opts.draw_move_lines
        inds = res.pre_cluster_indices{t+1};
        lines = plot([t-1 t], ...
                     [res.Ops{t} res.Ops{t+1}(inds)], ...
                     opts.linestyle, 'Color', opts.active_color);
      end
      axis([-0.3 length(res.Ops) mins(1) maxs(1)]);
    case 2
      if t > 0 && opts.draw_move_lines
        inds = res.pre_cluster_indices{t+1};
        lines = plot([res.Ops{t}(:,1)'; res.Ops{t+1}(inds,1)'],...
                     [res.Ops{t}(:,2)'; res.Ops{t+1}(inds,2)'],...
                     opts.linestyle, 'Color', opts.active_color);
      end
      circs = plot(res.Ops{t+1}(:,1)', res.Ops{t+1}(:,2)',...
                   opts.opinion_marker, ...
                   'Color', opts.active_color);
      axis([mins(1) maxs(1) mins(2) maxs(2)]);
    case 3
      if t > 0 && opts.draw_move_lines
        inds = res.pre_cluster_indices{t+1};
        lines = plot3([res.Ops{t}(:,1)'; res.Ops{t+1}(inds,1)'],...
                      [res.Ops{t}(:,2)'; res.Ops{t+1}(inds,2)'],...
                      [res.Ops{t}(:,3)'; res.Ops{t+1}(inds,3)'],...
                      opts.linestyle, 'Color', opts.active_color);
      end
      circs = plot3(res.Ops{t+1}(:,1)', res.Ops{t+1}(:,2)',...
                    res.Ops{t+1}(:,3)', opts.opinion_marker,...
                    'Color', opts.active_color);
      axis([mins(1) maxs(1) mins(2) maxs(2) mins(3) maxs(3)]);
  end
  title(sprintf('t=%d,clusters=%d', t, clusters));
  drawnow

  if opts.pause_time > 0
    pause(opts.pause_time);
  end

  if t >= tmax || ~ishandle(fig)
    break;
  end
  if dimensions > 1
    if opts.delete_old
      delete(lines);
      delete(circs);
```

```
      else
        set(lines, 'Color', opts.old_color);
        set(circs, 'Color', opts.old_color);
      end
    end
    t = t + 1;
  end

  if ishandle(fig)
    switch dimensions
      % we don't do anything special for 1d
      case 2
        if ~opts.delete_old
          plot(res.Ops{t+1}(:,1)', res.Ops{t+1}(:,2)', ...
               opts.final_cluster_marker, 'Color', opts.active_color,...
              'MarkerSize', 15, 'LineWidth', 2);
        end
        if opts.draw_circles;
          viscircles(res.Ops{t+1},ones(length(res.Ops{end}),1),...
                     'LineWidth', 1);
        end
        if draw_str_data
          text(res.Ops{t+1}(:,1)',res.Ops{t+1}(:,2)', final_data,...
               'VerticalAlignment','bottom', 'FontSize', 18);
        end
      case 3
        plot3(res.Ops{t+1}(:,1)',res.Ops{t+1}(:,2)',res.Ops{t+1}(:,3)',...
              opts.final_cluster_marker, 'Color', opts.active_color, ...
              'MarkerSize', 15, 'LineWidth', 2);
        if draw_str_data
          text(res.Ops{t+1}(:,1)',res.Ops{t+1}(:,2)',res.Ops{t+1}(:,3)',...
              final_data, 'VerticalAlignment','bottom', 'FontSize', 18);
        end
    end
    title(sprintf('t=%d, clusters=%d',t,clusters));
  end
end
```

B.4 HK_opts.m

Provides options for the file `HK_plot.m` and

```
function [ opts ] = HK_opts( IOps )
% Generates default options for an initial configuration IOps. IOps is an
% optional argument. If it is not supplied a ''worst case'' configuration
% will be assumed (read:large configuration).

    % HK_multi_dim.m opts
    % If the solver should print things.
    opts.verbose = true;

    % Maximum norm difference for ''different'' opinion matrices.
    opts.maxnorm = 1e-6;

    % Term added to ensure that neighbours are actually counted. This
    % results in a slight overcounting of the number of neighbours to an
    % agent, but having it as 0 leads to larger errors as agents are then
    % undercounted.
    opts.epsilon = 1e-13;

    % What constitutes a LARGE number of agents. If there are more agents
    % than this the solver will calculate the opinions in parallell.
    opts.large_num_agents = 10000;

    % Minimum time between each successive progress message. The solver can
    % only state its progress in between time steps though.
    opts.print_progress_delay = 60;

    % Minimum time between each save. The solver can only save its progress
```

```
    % in between time steps though, so the time is only approximate.
    opts.save_progress_delay = 3600;

    % The path the solver saves progress to.
    opts.save_progress_path = 'current_progress';

    % HK_plot opts
    % Color of agents in the current time step.
    opts.active_color = [0 0 1];

    % Color of agents in previous time steps, if delete_old is false.
    opts.old_color    = [0.5 0.5 0.5];

    % Symbol to represent agents
    opts.opinion_marker = 'o';

    % Style used for lines connecting successive opinions, if
    % draw_move_lines is true.
    opts.linestyle = '-';

    % Symbols used to clearly mark where the configurations ended up, only
    % used if delete_old is false.
    opts.final_cluster_marker = 'x';

    % What constitutes a small number of agents. This controls what
    % defaults are reasonable when plotting
    opts.small_num_agents = 500;

    % If no initial configuration is given we have to guess what the number
    % of agents is, we guess large, since small configurations work with
    % the large number defaults, but the small number defaults don't work
    % with large numbers of agents (plotting 10000 lines is hard work,
    % and mostly looks cluttered.
    if nargin == 1 %
        [agents,~] = size(IOps);
    else
        agents = inf;
    end
    opts.tmax = inf;

    % Bool saving wether or not the number of agents is small.
    opts.is_small_num_agents = agents < opts.small_num_agents;

    % Only pause if the number of agents is small, otherwise the time it
    % takes to plot is sufficient.
    opts.pause_time = opts.is_small_num_agents*0.3;

    % Draws lines between the old opinion and the new for each agent. We
    % only do this for small number of agents.
    opts.draw_move_lines = opts.is_small_num_agents;

    % Deletes old points. We delete for large numbers of agents.
    opts.delete_old = ~opts.is_small_num_agents;

    % This one is mostly for debugging. Draws circles over the final
    % configurations.
    opts.draw_circles = false;
end
```

C Vectors And Row Sums From Theorem 4.2

$$\boldsymbol{y_0} = \begin{bmatrix} 0.939434455792740 \\ 0.974026205690910 \\ 0.988867155446126 \\ 0.995281299625763 \\ 0.998133487411561 \\ 0.999347260629519 \\ 0.999803484327996 \\ 0.999949216324104 \\ 0.999988599972631 \\ 0.999997733206551 \\ 0.999999592532790 \\ 0.999999932745440 \\ 0.999999989730794 \\ 0.999999998552695 \\ 0.999999999813982 \\ 0.999999999978612 \\ 0.999999999997848 \\ 0.999999999999815 \\ 0.999999999999987 \\ 0.999999999999999 \\ \vdots \end{bmatrix} \text{row sums of } \mathbf{T} = \begin{bmatrix} 0.943289394780053 \\ 0.976626016149058 \\ 0.990378548497346 \\ 0.996051088497944 \\ 0.998480647346650 \\ 0.999486654774494 \\ 0.999853388903845 \\ 0.999965130541467 \\ 0.999993112164404 \\ 0.999998868765432 \\ 0.999999845841670 \\ 0.999999982761419 \\ 0.999999998455863 \\ 0.999999999893532 \\ 0.999999999994697 \\ 0.999999999999830 \\ 0.999999999999997 \\ 1.000000000000000 \\ \vdots \end{bmatrix}$$

$$\text{non-zero elements of } 18th \text{ row in } \mathbf{T}= \begin{bmatrix} 0.000000000000002 \\ 0.000000000000132 \\ 0.000000000004008 \\ 0.000000000077951 \\ 0.000000001092417 \\ 0.000000011750672 \\ 0.000000100924016 \\ 0.000000710907940 \\ 0.000004186424759 \\ 0.000020905815945 \\ 0.000089494781581 \\ 0.000331201792577 \\ 0.001066657232324 \\ 0.003005153910970 \\ 0.007437386488071 \\ 0.016222305311175 \\ 0.031265519829195 \\ 0.053352061225862 \\ 0.080729347237190 \\ 0.108440297672697 \\ 0.129407305986437 \\ 0.137254703068160 \\ 0.129407305986437 \\ 0.108440297672697 \\ 0.080729347237190 \\ 0.053352061225862 \\ 0.031265519829195 \\ 0.016222305311175 \\ 0.007437386488071 \\ 0.003005153910970 \\ 0.001066657232324 \\ 0.000331201792577 \\ 0.000089494781581 \\ 0.000020905815945 \\ 0.000004186424759 \\ 0.000000710907940 \\ 0.000000100924016 \\ 0.000000011750672 \\ 0.000000001092417 \\ 0.000000000077951 \\ 0.000000000004008 \\ 0.000000000000132 \\ 0.000000000000002 \end{bmatrix}$$

$$\tilde{\boldsymbol{c}} = \begin{bmatrix} 0.003587993320581 \\ 0.002786941973362 \\ 0.001946652172613 \\ 0.001241650117371 \\ 0.000729718584874 \\ 0.000397283599498 \\ 0.000201274026043 \\ 0.000095399986897 \\ 0.000042599281979 \\ 0.000018074390882 \\ 0.000007358600904 \\ 0.000002905015619 \\ 0.000001123483585 \\ 0.000000429432534 \\ 0.000000163290714 \\ 0.000000062005826 \\ 0.000000023550730 \\ 0.000000008949611 \\ 0.000000003402465 \\ 0.000000001294409 \\ 0.000000000493342 \\ 0.000000000188995 \\ 0.000000000073380 \\ 0.000000000029459 \\ 0.000000000012770 \\ 0.000000000006426 \\ 0.000000000004013 \\ 0.000000000003093 \\ 0.000000000002740 \\ 0.000000000002602 \\ 0.000000000002546 \\ 0.000000000002520 \\ 0.000000000002506 \\ 0.000000000002495 \\ 0.000000000002487 \\ 0.000000000002478 \\ 0.000000000002470 \\ 0.000000000002462 \\ 0.000000000002453 \\ 0.000000000002444 \\ 0.000000000000802 = (\gamma(1)) \\ a\gamma^{42} \\ a\gamma^{43} \\ \vdots \end{bmatrix}$$

References

[1] U. Krause. Soziale dynamiken mit vielen interakteuren. eine problemskizze. *in: Modellierung und Simulation von Dynamiken mit vielen interagierenden Akteuren, U. Krause and M. Stöckler eds., Universität Bremen*, 1997.

[2] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[3] A. Mirtabatabaei and F. Bullo. Opinion dynamics in heterogeneous networks: Convergence, conjectures and theorems. *SIAM Journal on Control and Optimization*, 50(5):2763–2785, 2012.

[4] B. Chazelle Bernard A. Bhattacharyya, M. Braverman and H.L. Nguyen. On the convergence of the Hegselmann-Krause system. *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, pages 61–66, 2013.

[5] S. Mohajer and B. Touri. On convergence rate of scalar Hegselmann-Krause dynamics. *http://arxiv.org/abs/1211.4189*, 2012.

[6] E. Wedin and P. Hegarty. A quadratic lower bound for the convergence rate in the one-dimensional Hegselmann–Krause bounded confidence dynamics. *Discrete & Computational Geometry*, 53(2):478–486, 2015.

[7] E. Wedin and P. Hegarty. The Hegselmann-Krause dynamics for continuous agents and a regular opinion function do not always lead to consensus. *To appear in IEEE Transactions on Automatic Control. Available online at http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7027168*, 2014.

[8] J.M. Hendrickx V.D. Blondel and J.N. Tsitsiklis. On Krause's multi-agent consensus model with state-dependent connectivity. *IEEE transactions on Automatic Control*, 54(11):2586–2597, 2009.

[9] P. Hegarty and E. Wedin. The Hegselmann-Krause dynamics for equally spaced agents. *http://arxiv.org/abs/1406.0819*, 2014.

[10] J.M. Hendrickx V.D. Blondel and J.N. Tsitsiklis. Continuous-time average-preserving opinion dynamics with opinion-dependent communications. *SIAM Journal on Control and Optimization*, 48(8):5214–5240, 2010.