Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

Opinion Formation: Impacts of convincing extreme individuals onto a society that typically converges to one opinion

Alexander Stein, Niklas Tidbury & Elisa Wall

Zurich
December 2017

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Alexander Stein   Niklas Tidbury   Elisa Wall

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1 Abstract

For millennia, society has consisted of many opinions and points of view. In some cases, these opinions have been oppressed, other opinions have been forced onto societies, others brainwashed. Within a democracy, these opinions are given space to spread, to change and to evolve and yet: they still converge into a general opinion. How is this possible in cases of extremism, where extreme opinions are so different compared to the majority? What effect do extreme opinions, such as that of the IS, Charles Manson and Co. have on a converging opinion of a society? We would like to examine how extreme opinions of individuals impacts such a society, and under what circumstances these opinions can have a wide-spread effect.

2 Introduction and Motivations

Basing on the papers of Holme and Newman [1] such as Laguna, Abramson and Zanette [2], we create a society on an agent-based model. These agents being single beings with an opinion in the continuous interval [0,1]. A parameter μ is defined, which is the weight a agent gives to foreign opinions. Agents of the society will interact with each other, "exchanging" their opinions and deciding on common ground. This "common ground" both agents share is defined if within a distance u on the interval of opinions. If the parameters are set accordingly, we expect that the opinion converges against an average opinion 0.5. First, we implement this model and investigate the parameters to reproduce some results of the papers.

Next, we extend the model with extreme opinions. They are defined by an opinion of 0 or 1 and have special properties which one can summarize in 2 parameters. One will be called n_{conv} and characterize the the number of convinced agents per time step. The other one $infop$ characterizes the range of influenced opinions. Within this report, we will discuss the extension and its consequences. We will focus on societies that typically converge against a common opinion and investigate the stability of the convergence when inserting the extremists.

3 Model and Implementation

3.1 Society agent in a single time step

We have set a society of N agents with opinions $x_i \in [0, 1]$, $i \in \{1, N\}$. Initially the distribution of opinions is uniform, which means that all opinions are equally alike.

We further define the parameter of a threshold of communication u. The agents interact with each other only if their difference in opinion is smaller than this threshold. In [2] this is introduced as the "bounded confidence" which corresponds to the fact that people tend to spend time with agents with similar opinion, for instance circle of friends tend to be of similar opinion. By building a threshold u into the structure, the agents only interact with similar-thinking agents, as people generally interact with the like-minded. A big u therefore corresponds to open-minded society agents which interact with people of an opinion "further away".

The second parameter of the society agents is the μ, which represents the weight on other opinions. When two agents interact with each other, they mutually adapt their opinion to some opinion in the middle of both.

The concrete implementation as in [2] looks as follows:

$$\begin{aligned} x(t+1) &= x(t) + \mu(x'(t) - x(t)) \\ x'(t+1) &= x'(t) + \mu(x(t) - x'(t)) \end{aligned} \tag{1}$$

Note that only weight adjustments $0 < \mu < 0.5$ are allowed to guarantee that the opinions after are between the two initial opinions and that the values do not exceed the range of [0,1]. Next to the technical fact, a negative value would mean that the opinions of the two agents would develop in the opposite direction of opinion and a value higher than 0.5 would mean that x' will be closer to the opinion of x than x itself after the interaction.

A characteristic property of this model is the fact that close opinions come even closer and opinions further away from each other do not influence each other, whereby "close" and "far-away" are defined by u.

Up to now we have defined under which threshold two agents meet and under which weight they exchange their opinion. Next, we define a single time step in such a way that every agent randomly chooses another agent within the society and the threshold of communication u to interact with him as described above. Then we iterate over the T time steps to get the final update of opinions.

As concluded in [2], first of all the parameters μ and T are related in such a way that the results at given u do not vary if by reducing one the other is increased. If moreover a high enough value of $u = 0.3$ or greater as a communication threshold is chosen, the society's opinion converges to a unique opinion around $x = 0.5$.

3.2 Extremists

Basing on the setting of society of interacting agents with the three parameters u, μ and T of the last chapter we add extreme individuals of non-changing opinion on each of the outskirts of the opinion range at 0 and 1. They represent charismatic individuals of good rhetoric, able to convince many people's opinion, such as social

media stars or politicians with extreme opinions.
This is implemented with the following parameters for each of the two sides: At each time step the n extreme opinion individuals can persuade p other agents by a probability of success of κ each, but again only within a threshold of communication, this time called $infop$.
We seemingly add four new parameters. However, in fact there are some correlations. First, we have an effective number of persuaded people in each time step of $n_{eff} = n{\cdot}p$ that can be united to one parameter. Secondly, this again can be united to $n_{conv} = \kappa \cdot n_{eff}$, under the constraint that we have large numbers such that statistical averages take place.
In the end we have only two additional parameters. n_{conv}, which defines the amount of convinced people and $infop$ which defines the range of influence and remained as before.

4 A world without extremists

For deeper understanding and for having references to later cases we started the simulations with a society without extremists. Therefore, we coded situations as given in [2] in order to reproduce their results.

4.1 Speed of convergence

We reproduce the fact that μ acts as a parameter of convergence. In case of constant u, a small μ has to be compensated with a large number of time steps T. The resulting stable state, defined by T running against infinity, will be the same.

The effect is shown in figure 1. Because the weight on foreign opinions μ was increased from 0.03 to 0.3, the number of time steps T could be decreased from 200 to 30 with the same steady state of a common opinion around 0.5.

4.2 Cluster building

According to [2], such a society of N agents with random uniform distributed opinions x_i in [0,1] will converge to a common opinion around 0.5 if the communicating interval u of agents communicating with each other is big enough, more precise if $u > 0.3$. This was repeated and shown in figure 2.

In case $u = 0$, there is no opinion interaction between the agents and the uniform distribution of the opinions gives us the percentage $\frac{|0.45-0.50|}{1} = 10\%$. For values of u between 0.18 and 0.22 we see that almost no opinions are between 0.45 and 0.55. This corresponds to the fact that the opinions converge against two opinions that are

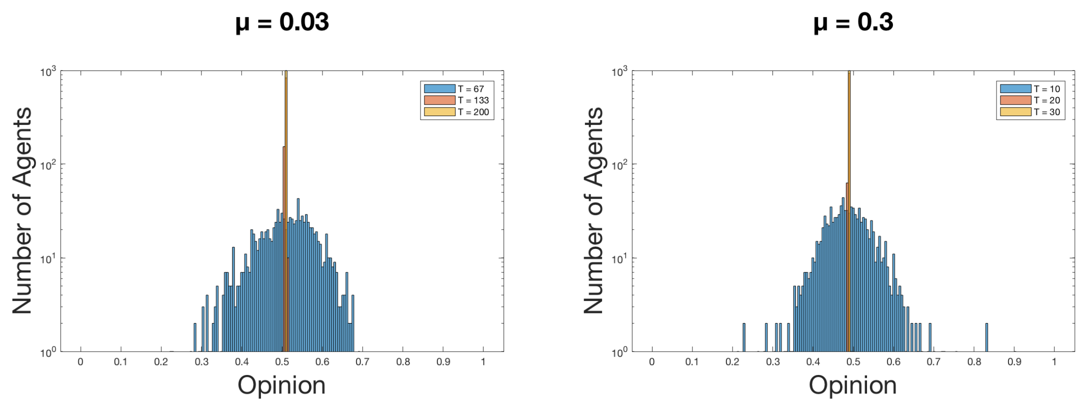

Figure 1: Opinion distribution for different times in a society **without extremists** for a given μ of 0.03 or 0.3. We fixed the number of society agents $N = 1000$ and the threshold $u = 0.3$

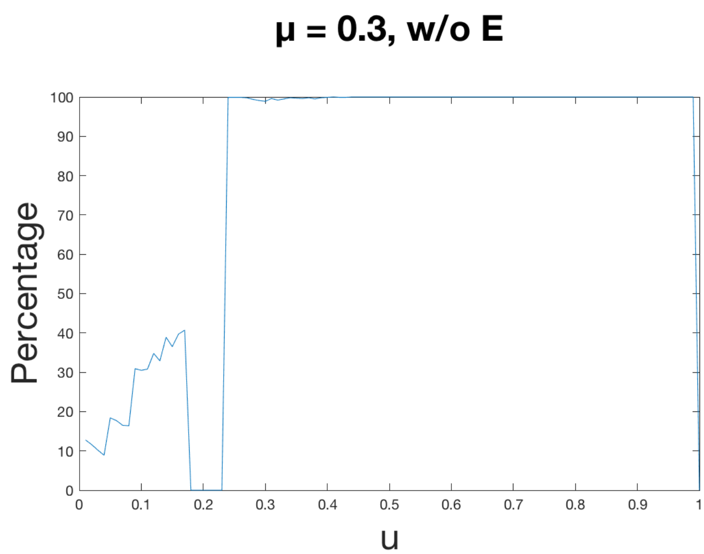

Figure 2: Percentage of agents in central cluster between 0.45 and 0.55 when varying u

around 0.25 and 0.75, shown in [2]. If $u > 0.24$ we see almost 100% in this cluster. This is the main result of this chapter and is defined as the convergence against a common opinion around 0.5.

5 A world with extremists

The society model described above is being extended to a society with extreme opinion individuals who can influence the agents. We carry on to vary different parameters in this society model.

5.1 The "new" parameter μ

In the case of a society without extremists, μ was a measure of convergence. However, it did not have an influence on the final result. If a small μ was chosen, the number of time steps T could be increased accordingly to get the same result of convergence (See Chapter 4.1).
While varying the values of μ in a society with extremists, μ should not influence only the speed of convergence, but also the spread of opinions. If a small μ is chosen, thus the natural convergence is slow, this should lead to more agents with extreme opinions, compared to a high μ.
This would be explained as follows: If the natural convergence is slow, more time steps are needed to get to the stationary state. But the extremists influence is independent of μ. Therefore, if the natural influence is small, the extremists should have more time to grasp agents with opinions in the interval accessible to the the agents who could pull them to the average opinion cluster at $x = 0.5$.
In figure 3 we show the opinion distribution o two societies with different μ value in a logarithmic histogram. We expect to see more extremists on the left plot since it corresponds to a smaller μ, but see a comparable plot for the two scenarios. Hence we could not produce the expected result of the new μ.

5.2 The interval of influence

Next, the interval of influence $infop$ is varied. Concretely, the range of people influenced by the extremists with opinion $x = 0$ is in [0, infop] and by the extremists with opinion $x = 1$ is in [1-infop, 1]. We vary infop from 0 to 0.5.

The results can be seen in figure 4. We see that the convergence with this set of parameters is amazingly stable until we come close to $infop = 0.5$. Further, the number of convinced agents, defined by n_{conv} and controlled by varying p, has a small influence for small $infop$, but at exactly 0.5 a higher p leads to more extremists.

If a low number of extremists ($p \leq 20$) takes influence on our society, it is important for them to reach almost every opinion even in the middle. Otherwise we always see a strong cluster around the opinion 0.5. Note that for $infop = 0.5$ we reach all opinions which will lead to a situation where all agents have extreme opinions for T running against infinity.

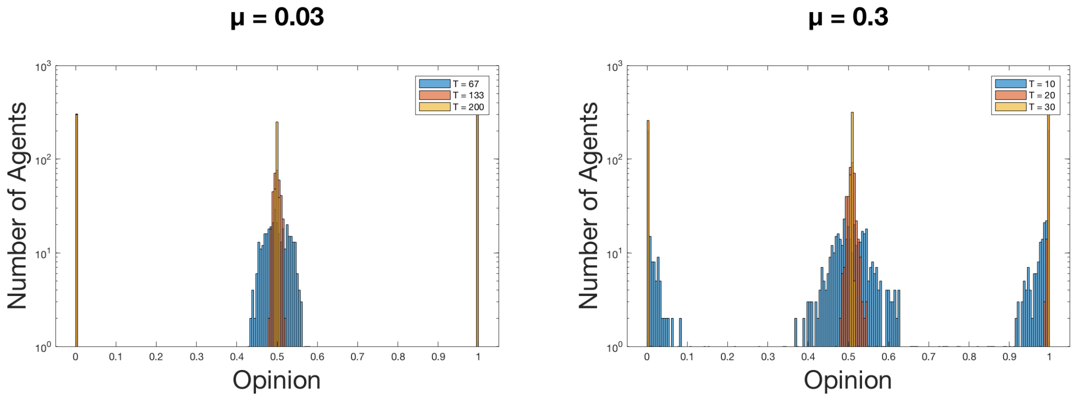

Figure 3: Opinion distribution for different times in a society **with extremists** for a given μ of 0.03 or 0.3. We fixed the number of society agents $N = 1000$ and the threshold $u = 0.3$ for $n_{conv} = 10$

5.3 The power of convincing

We fix $\mu = 0.1$ and $u = 0.32$ to be sure that in the case of a society without extremists we would see a common convergence of opinions to 0.5. The extremists are inserted with $infop = 0.3$. We examine the development of the number of extreme opinions. More precise we measure the percentage of the opinions in the intervals $x \in [0, 0.1] \cap [0.9, 1]$.
For simplicity we fix $n = 1$ and $\kappa = 0.2$ and vary the number of influence-able people $p \in [0, P]$. One can calculate the final parameter for the amount of influenced agents simply with $n_{conv} = n \cdot \kappa \cdot p$.

Figure 5 shows the result of the described measurement. For $p < 50$ we see an the obvious relation that for stronger influence of the extremists also the number of extremists increase. For $50 < p < 250$ we see a jump and the extremists manage to convince the whole society. An interesting case is given again for $p > 250$. The percentage of the extreme opinion goes down again. This corresponds to the fact that they destroy "bridges". They are so strong that they collect all agents in their range defined by $infop$ within a few time steps. It stays a rest in the middle that converges again to 0.5.

The interesting numbers are $p = 50$ and $p = 250$. They define the range in which the extremists are able to use μ to their advantage and are strong enough to collect all agents even if they cannot reach all agents directly (see chapter 4.2).

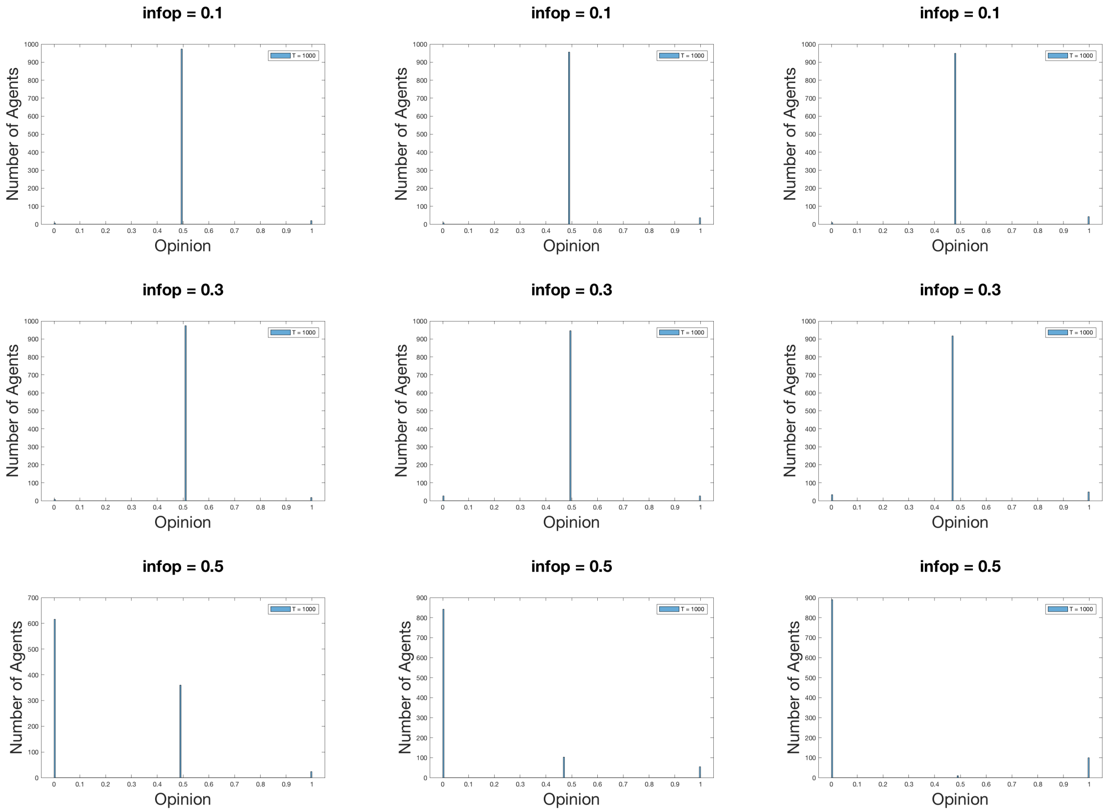

Figure 4: Opinion spread for different $infop$ for p = 5, 10 and 20 (from left to right). We fixed $N = 1000$, $\mu = 0.3$ and $u = 0.32$.

6 Summary and Discussion

The challenge of the simulation was to keep track of which parameters were to be varied in the society with extremists compared to the society wihtout. We managed to reproduce the effects explained in the paper. The fact that Laguna found converging opinions for $u > 0.3$ and we found converging opinions already for $u > 0.24$ is unexplained. However, we assumed that we implemented the code like Laguna.

It would have been interesting to see the change of the parameter μ in the second society compared to the first. Unfortunately this effect could not be shown. It was further interesting to see that a variation of the interval of influence $infop$ did not make a big difference on the result up to the point where $infop$ comes close to the value 0.5. In chapter 5.3, we also found a interval for strength of the extremists, for which they collect all society agents. If the number of influencable agents was

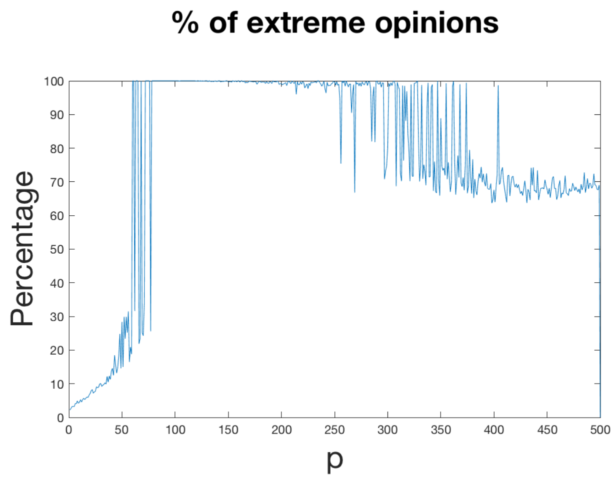

Figure 5: Fraction of extreme opinions in $[0, 0.1] \cap [0.9, 1]$ for convinced agents $\propto p$

increased, first the proportion of extreme opinions, then jumped to 100% and decreased again for a very high number of influencable agents due to bridge effects.

7 Outlook

Up to now we considered only symmetric random distributions and symmetric distribution of extremists. We could have also looked at asymmetric distributions and look at these results. For instance, if we consider $x = 0$ as political left and $x = 1$ as political right within our model, we can give the politicians different strength or range of influence.
One could further look at a Gaussian instead of a uniform distribution of opinions, which would strengthen the average opinion. It would be a sensible alternative opinion distribution where 0.5 is the main stream opinion and other opinions are spread around the main stream distribution. Sigma would be measure for 'how far' the other opinions are spreaded from the main stream opinion.
It would further be interesting to vary N to see other bridge effects as described in chapter 5.3 or the number of time steps T needed to reach the steady state for different settings of parameters.

References

[1] Peter Holme and M. E. J. Newman. *Nonequilibrium phase transition in the coevolution of networks and opinions*. arXiv:physics/0603023v3, 9 March 2006.

[2] M. F. Laguna, Guillermo Abramson, and Damian H. Zanette. *Minorities in a Model for Opinion Formation*. Wiley Periodicals, Inc., Vol. 9, No.4, 5 January 2004

```
%% Project on Opinion Formation for the course
%% "Modelling and Simulating of Social Systems with MATLAB"
%% author: The Opinionators (Elisa Wall, Alexander Stein, Niklas Tidbury)

%% number of time steps
T = 500;

%% number of iterations
Tg = 50;

%% number of society agents
N = 1000;

%% Properties of the SocietyAgents
% The threshold u defines when two agents speak/interact with each other
u = 0.32;


% Mu defines the change of opinion when two agents speak with each other
% mu has to be between 0 and 1 to ensure that all opinions are
% opinions are between 0 and 1.
mu = 0.3;


%% Properties of the extremists
% number of extremists
n0 = 1;
n1 = 1;
% number of agents one extremist can reach
p0 = 500;
p1 = 500;
% An extremist convinces an agent with probability kappa
kappa0 = 0.2;
kappa1 = 0.2;
% an extremist has a range of people he reaches
% The extremist with opinion 0 can reach all agents with opinion in
% [0,infop0], repectively extremists with opinion 1 to [infop1, 1]
infop0 = 0.3;
infop1 = 1-infop0;


%% run the program


%% Figure 1
gen_plot("hist", false, 3, run_simulation("without", create(N), Tg, 30, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1), "μ = 0.3, w/o E", "Opinion", "Number of Agents", 30, N, false);
gen_plot("hist", false, 3, run_simulation("without", create(N), Tg, 200, N, u, 0.03, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1), "μ = 0.03, w/o E", "Opinion", "Number of Agents", 200, N, false);

%% Figure 2
gen_plot_interval("line", "% of opinion between 0.45 and 0.55, w/o E", "μ", "Percentage", false, "without", "mu", create(N), Tg, 15, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1);

%% Figure 3
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 30, N, u, mu, n0, 50, kappa0, n1, 50, kappa1, infop0, infop1), "μ = 0.3, w/ E, p = 50", "Opinion", "Number of Agents", 30, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 200, N, u, 0.03, n0, 50, kappa0, n1, 50, kappa1, infop0, infop1), "μ = 0.03, w/ E, p = 50", "Opinion", "Number of Agents", 200, N, false);

%% Figure 4
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 5000, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1), "μ = 0.3, w/ E, p = 500", "Opinion",
```

```
"Number of Agents", 5000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 5000, N, u, 0.03, n0,↙
p0, kappa0, n1, p1, kappa1, infop0, infop1), "μ = 0.03, w/ E, p = 500", "Opinion",↙
"Number of Agents", 5000, N, false);

%% Figure 5
gen_plot_interval("line", "% of opinion between 0.45 and 0.55, w/ E", "μ",↙
"Percentage", false, "with", "mu", create(N), Tg, 15, N, u, mu, n0, 50, kappa0, n1,↙
50, kappa1, infop0, infop1);

%% Figure 6
% infop = 0.1
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

% infop = 0.3
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

% infop = 0.5
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

%% Figure 7
gen_plot_interval("line", "% of extreme opinions", "p", "Percentage", true, "with",↙
"p", create(N), Tg, 10000, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1);




%% Functions


%% Function for generating plots
% plot_type = plot type (hist or line)
% slider_bool = when number_of_plots is 1, the option to add a slider is
% given
% number_of_plots = number of data sets in plot
% data = data from simulation (call run_simulation())
% plot_name = name window of plot
% T = entire time (must be same T as passed to run_simulation())
% N = society size (must be same N as passed to run_simulation())
% save = activate / deactivate saving plot as png in folder "exports"
function [] = gen_plot(plot_type, slider_bool, number_of_plots, data, plot_name,↙
x_axis, y_axis, T, N, save)
    figure('name', plot_name);
    disp("Running...");
```

```
    if plot_type == "hist"
        edges = linspace(0,1,200);
        if number_of_plots > 1
            for i = 1:number_of_plots
                histdata = data(round(i*T/number_of_plots),:);
                histogram(histdata, edges, 'DisplayName', ['T = ', num2str(round↙
(i*T/number_of_plots))]);
                hold on;
            end
        else
            slmin = 1;
            slmax = T;
            histogram(data(T,:), edges, 'DisplayName', ['T = ', num2str(T)]);
            if slider_bool
                hsl = uicontrol('Style','slider','Min',slmin,'Max',slmax,...
                    'SliderStep',[1 1]./(slmax-slmin),'Value',1,...
                    'Position',[50 10 500 10],'FontSize', 25);
                set(hsl,'Callback',@(hObject,eventdata) histogram(data(round(get↙
(hObject,'Value')),:), edges, 'DisplayName', ['T = ', num2str(round(get↙
(hObject,'Value')))]));
            end
        end
        legend('show');
    elseif plot_type == "line"
        tot_perc = zeros(T);
        for i = 1:T
           tot_perc(i) = countPercentage(0,0.1,data(i,:),N);
           if tot_perc(i) == 100
               disp(["100% at T: ", num2str(i)]);
           end
        end
        plot(tot_perc);
    end
    title({' ', plot_name, ' '}, 'FontSize', 25);
    xlabel(x_axis, 'FontSize', 25);
    ylabel(y_axis, 'FontSize', 25);
    % because data is lost for large Ts and log scaling
    if T < 400
        set(gca,'yscale','log');
    end
    % save to file with unique name
    if save
       format shortg;
       c = clock;
       fix(c);
       filename = sprintf("exports_second/gen_plot_%d%d%d%d%d%d.png",c(1),c(2),c(3),c↙
(4),c(5),c(6));
       saveas(gcf,filename);
    end
    disp("Finished!");
end


%% Function for generating plots over interval of a certain var (can be customized)
% plot_type= plot type (hist or line)
% plot_name = name window of plot
% param = string name of variable to generate and plot over
% other vars same as usual
function [] = gen_plot_interval(plot_type, plot_name, x_axis, y_axis, save, simtype,↙
param, op, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1)
    FIG = figure('name', plot_name);
    disp("Running...");
    if param == "p"
        if plot_type == "line"
            perc_total = zeros(p0, 1);
            for j = 1:p0
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, u, mu, n0, j, kappa0, n1,↙
```

```
j, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0, 0.1, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at p0: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end
    elseif param == "u"
        if plot_type == "line"
            perc_total = zeros(50, 1);
            for j = 1:100
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, j/100, mu, n0, p0, kappa0,↙
n1, p1, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0.45, 0.55, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at p0: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end

    elseif param == "mu"
        if plot_type == "line"
            perc_total = zeros(50,1);
            for j = 1:50
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, u, j/100, n0, p0, kappa0,↙
n1, p1, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0.45, 0.55, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at mu: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end
    end
    title({' ', plot_name, ' '}, 'FontSize', 25);
    xlabel(x_axis, 'FontSize', 25);
    ylabel(y_axis, 'FontSize', 25);
    if param == "u"
        % set x axis labelling for u
        xticks([0 10 20 30 40 50 60 70 80 90 100]);
        xticklabels({'0','0.1','0.2','0.3','0.4','0.5','0.6', '0.7', '0.8', '0.9',↙
'1'});
    elseif param == "mu"
        % set x axis labelling for mu
        xticks([0 10 20 30 40 50]);
        xticklabels({'0','0.1','0.2','0.3','0.4','0.5'});
    end
    disp("Finished!");
    % save file with unique name
    if save
       format shortg;
       c = clock;
       fix(c);
       filename = sprintf("exports_second/gen_plot_intervall_%d%d%d%d%d%d.png",c(1),c↙
(2),c(3),c(4),c(5),c(6));
       saveas(gcf,filename);
    end
end


%% Function for calculating average over several simulated societies and plotting them↙
```

```
accordingly
% param = with / without extremists
% number_of_plots: number of data sets in plot
% data = data from simulation (call run_simulation())
% plot_name = name window of plot
% plot_type = type of plot
% x_axis / y_axis = axis labelling
% other vars = as usual
% save = activate / deactivate saving plot as png in folder "exports"
function [] = gen_av_plot(param, number_of_plots, plot_type, plot_name, x_axis,↙
y_axis, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1, save)
    av_matrix = zeros(T, N, Tg);
    for j = 1:Tg
        av_matrix(:,:,j) = run_simulation(param, create(N), Tg, T, N, u, mu, n0, p0,↙
kappa0, n1, p1, kappa1, infop0, infop1);
    end
    size(av_matrix)
    av = mean(av_matrix,3);
    gen_plot(plot_type, number_of_plots, av, plot_name, x_axis, y_axis, T, N, save);
end


%% Running a round of simulation
% returns an T x N matrix of data from the simulation, so we have the data
% from each given time step for analysis
function [data] = run_simulation(param, op, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1,↙
kappa1, infop0, infop1)
    if param == "without"
        data = without(op, T, N, u, mu);
    elseif param == "with"
        data = with(op, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1);
    end
end


%% Creating the society
% Input: number of extremists
% Output: (1xN) opinion matrix of an arbitrary random distribution
%       (gaussian or uniform distribution)
function [op] = create(N)
    %% Creating the society
    % A society of N SocietyAgents with opinions op in [0,1] that are randomly
    %       distributed

    % uniform distribution
    op = rand(1,N);
end


%% The program without extremists as given in the paper
% Input: T, N, u, mu
% Output: updated opinion
function [simulation] = without(op, T, N, u, mu)
    %%  A world without extrimists
    %   The influence of a single agent in a singlte timestep t is defined in the
    %   function SocietyAgent. We raise up the time steps to T. In every
    %   time step t, every agent has the chance to speak with another.

    simulation = zeros(T, N);
    simulation(1,:) = op;
    for t = 1:T+1
        for i = 1:N
            [op0, op1, k] = SingleAgent(op(i), op, u, N, mu);
            op(i) = op0;
            op(k) = op1;
        end
        simulation(t+1, :) = op;
    end
```

```
end


%% The program with extremists
% Input: T, N, mu, n0, n1, p0, p1, kappa0, kappa1, infop0, infop1
% Output: histograms
function [simulation] = with(op, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0,↙
infop1)
    %% Effective number of influenced people by the extremists
    %   All agents have the same behavior, so we can sum up the influence of all
    %   agents in the number of people that get influenced

    simulation = zeros(T, N);
    neff0 = p0 * n0;
    neff1 = p1 * n1;

    % add start state to data matrix
    simulation(1,:) = op;

    %% A world with extremists
    for t = 1:T+1
        % For timestep t; the SocietyAgents play their game
        for i = 1:N
            [op0, op1, k] = SingleAgent(op(i), op, u, N, mu);
            op(i) = op0;
            op(k) = op1;
        end
        % For timestep t; the extremists with opinion 0 play their game
        for i = 1:neff0
            r = rand;
            k = randi(N);
            counter = 0;
            % extremists with opinion 0 only reach agents with similar opinion
            while op(k) > infop0 && counter < N
                k = randi(N);
                counter = counter + 1;
            end
            if r < kappa0
                op(k) = 0;
            end
        end
        % For timestep t; the extremists with opinion 1 play their game
        for i = 1:neff1
            r = rand;
            k = randi(N);
            % extremists with opinion 1 only reach agents with similar opinion
            while op(k) < infop1 && counter < N
                k = randi(N);
                counter = counter + 1;
            end
            if r < kappa1
                op(k) = 1;
            end
        end

        % add current state to data matrix
        simulation(t+1, :) = op;
    end
end


%% Defining the influence of a single SocietyAgent during one timestep t
% Input: op0 = opinion of a single agent, op = opinion of the society,
%       mu, u and N as described above
% Output: new opinion of op0 (opnew0),
%       new opinion of a randomly chosen op1 (opnew1) that interacted with
%       op0 and the position of op1 (pos)
function [opnew0, opnew1, pos] = SingleAgent(op0, op, u, N, mu)
```

```
    % op0 meets a randomly chosen agent in the society op, called op1
    pos = randi(N);
    op1 = op(pos);
    if abs(op1 - op0) < u
        % weighted difference of opinions, weigh is given by mu
        opnew0 = op0 + mu*(op1-op0);
        opnew1 = op1 + mu*(op0-op1);
    else
        opnew0 = op0;
        opnew1 = op1;
    end
end


%% Calculate percentage of opinions in certain interval (extremist, 0-0.1, 0.9-1)
function [perc] = countPercentage(lower, upper, op, N)
  counter = 0;
  for i = 1:N
      % add to counter if opinion in interval
       if (op(i) >= lower && op(i) <= upper) || (op(i) >= 1-upper && op(i) <= 1-lower)
           counter = counter + 1;
       end
  end
  % calculate percentage
  perc = counter/N*100;
end

%% Create movie over T from histogram data
function [] = createMovie(data, title, T)
    edges = linspace(0,1,200);
    vidObj = VideoWriter(title,'MPEG-4');
    vidObj.FrameRate = 20;
    open(vidObj);
    % iterate through data along T
    for t = 1:T
        histogram(data(t,:), edges, 'DisplayName', ['T = ', num2str(T)]);
        title({' ', plot_name, ' '}, 'FontSize', 25);
        xlabel(x_axis, 'FontSize', 25);
        ylabel(y_axis, 'FontSize', 25);
        pause(0.005);
        % draw plot and frame
        drawnow;
        F = getframe(FIG);
        writeVideo(vidObj,F);
    end
    close(vidObj);
end
```