



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

Modelling civil violence:

**An agent based approach
according to Joshua Epstein**

Alexander Grimm

Michael Koller

Zievi Ursin Solèr

Zurich

December 2010

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich diese Gruppenarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen-Hilsmittel verwenden habe, und alle Stellen, die wörtlich oder sinngemäss aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Gruppenarbeit nicht, auch nicht auszugsweise, bereits für andere Prüfung ausgefertigt wurde.

Alexander Grimm

Michael Koller

Zievi Ursin Solèr

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 Grimm

Michael Koller

Zievi Ursin Solèr

Contents

1	Abstract	6
2	Individual contributions	7
3	Introduction and Motivations	8
4	Description of the Model	10
4.1	Civil Violence Model I	10
4.2	Civil Violence Model II	11
5	Implementation	12
5.1	Outline of the Model I: ep_main_1.m	12
5.1.1	Add agents: add_agent.m	13
5.1.2	Get agents current view: get_view.m	13
5.1.3	Count agents in environment: count_agent.m . .	14
5.1.4	Show plots and measurements: show_final_plots.m	14
5.1.5	Show debug plots: show_plots.m	14
5.2	Outline of the Model II: ep_main_2.m	14
5.3	Model Variation	15
5.4	The third screen	15
5.5	The fourth screen	17
5.6	The agents vision	18
5.7	Consistency checks	19
6	Simulation Results and Discussion	20
6.1	Reproduction of the Epstein Model I	20
6.1.1	Run 1: Individual Deceptive Behaviour	20
6.1.2	Run 2: Free Assembly Catalyzes Rebellious Out- bursts	20
6.1.3	Run 3: Salami Tactics of Corruption (linear de- crease)	25
6.1.4	Run 4: Salami Tactics of Corruption (sudden drop)	25
6.2	Reproduction of the Epstein Model II	26
6.2.1	Run 6: Peaceful Coexistence	26
6.2.2	Run 7: Ethnic Cleansing	27
6.2.3	Run 8: Safe Havens	27
6.2.4	The Northwestern Trick	29

6.2.5	Parameter scan	29
6.3	Model II	38
7	Summary and Outlook	41
7.1	Comparison to the Epstein Model: Summary and Con- clusions	41
7.2	Outlook	41
7.2.1	Methods	41
7.2.2	Goals	42
8	References	43
A	MATLAB Code	44
A.1	Algorithm	44
A.2	Configuration	64
B	NetLogo Code	70

1 Abstract

The goal was to implement a civil violence model described by Epstein et al. [1],[2]. The model aims to describe the dynamics encountered in civil violence and ethnic conflict on the basis of an agent based approach and very simple rules.

The model did at first not show all the same behavioral patterns as described by Epstein even though it was carried out with exactly the same rules and parameters described. It turned out that this negative result had been previously noted by researchers at Northwestern University [7]. After the implementation of an additional rounding rule the model produced results consistent with Epstein's description.

For better visualization of the model's underlying dynamics additional screens of local agent's vision were added. We then went on and tried to reproduce the patterns described by Epstein. With the additional rounding rule similar results were observed for all given phenomena described. We then continued to carry out parameter scans and concluded with an outlook on possible improvements of the model.

Further material is also available from <http://n.ethz.ch/~usoler/public/>.

2 Individual contributions

The organization of our teams workflow required us to split up the different tasks into subtasks which were processed individually and compared frequently to assure a parallel and sensible outcome. The individual contributions are listed in the following table:

Code	(all)	Zievi Ursin Solèr
Measurements	14, 15, 16, 17, 18, 20, 21, 19, 22, 23	Alexander Grimm
Measurements	Fig. 3, 19	Michael Koller
Measurements	Fig. 2, 4, 5, 6, 7, 8, 10, 11, 12, 13	Zievi Ursin Solèr
LaTeX Report	Sec. 3, 4, 6, 7, 8	Alexander Grimm
LaTeX Report	Sec. 1, 5.5, 6, 7	Michael Koller
LaTeX Report	Sec. 5, 6, 7, 8, App. A, B	Zievi Ursin Solèr
Presentation	to be determined	Alexander Grimm
Presentation	to be determined	Michael Koller

3 Introduction and Motivations

Our goal is to implement a model which describes civil violence. With a good understanding of the dynamics of the model we might be able to understand civil violence in a better and appropriate way. The results of our investigation is going to be contrasted to the emprirical foundings given by [5, Gulden] and [4, Fearson et al.]. The fundamental architecture of the model is inspired by the model built by [1, Eppstein].¹

Studying the paper by [5, Gulden] we came to the conclusion that the Eppstein-Model could be an appropriated way in modeling the dynamics underling the civil war in Guatemala. The civial war in Guatemala took place from 1960 - 1996. The time from 1977 until 1986 was studied by [5, Gulden] because "an extraordinary data set"² was collected in this time. In the following we try to verify the model and test it to this given data.

In the following you can find all applications to our model which we want to implement.³ As stated before the basic construction is given by the Eppstein model (EM), but fist of all we want to modify the police. The police in the EM is quite simple. We want to model a more sophisticated state force. This is realized by giving them a bias like the military forces in the Guatemalian cical war actually had. Furthermore, we want to introduce a parameter model similar to the one used to model the agents of the different ethnicities. Going further, in our model there are two different ethinc groups. In order to make them diversivable we introduce different levels of grievance, state acceptance and so forth. This assumption is elementary to model a ethnic war with genocidal periods.

We want to introduce field parameters to account for a more likely scenario. With respect to the war which took place in Guatemala we want to adjust the grid of the model to account for a more realistic country shape.

Having these additional implementations built in we want to make an other parameter variation analysis to test the models sensitivity to

¹For further details see *Description of the Model*

²[5][Gulden, P. 26]

³You could consider this as add-ons to the Eppstein model

police bias and injustice. According to Wimmer et al.[3] in this approach only the special case of a rebellion is considered and attempted to be modeled.

Set up this model we want to discover limitations and significances of the EM approach in a qualitative manner. This includes finding a stable solution in which a minority group holds power (police favors minority group) and a majority which is out of power. Found a solution we want to make a statement about the robustness of such a solution by the means of the parameter analysis. Further we want to understand the effects of heterogeneity of grievance and state acceptance among different ethnic groups. Besides this, we could make a statement about the effect of a more realistic grid shape and size to the model. After all, we could try to confirm the findings of Wimmer et al.[3] who claim the share in state power is the main reason for civil violence.

4 Description of the Model

The underlying model is the model by Joshua M. Epstein [1, Epstein]. It is an agent based computational model which aims to describe the outbursts of rebellions. The model has two variants. In the first one a central authority suppresses a decentralized rebellion and in the second variant the central authority suppresses comunal violence. In the second version the agents belong to a certain ethinc group. If their grievance is high enough they get active in the sense that they become a potential killer. They aiming for killing the agents of the other athinc group. The model wants to describe civil violence although there is no political or social order implemented.

4.1 Civil Violence Model I

There are two types of agents. Firstly, the agents ⁴ who are the inhabitants of the state. Secondly, the cops who represent the state authority. The agents can stay quitesence or they can become activ. The cops arrest rebels if they appear within their vision.

Agents specification

The fundamental agents rule is the following:

If $G - N > T$ be active; otherwise be quite. ⁵

With, G: Grievance; N: Net risk; T: Threshold (non negative). The Grievance is been calculated by:

$$G = H(1 - L) \tag{1}$$

With L: legitimacy. The Net risk is been calculated by:

$$N = R * P * J^\alpha \tag{2}$$

With R: Level of risk aversion, J the jail term (α controls the injailments time) and the agents estimated arrest probability P:

$$P = 1 - exp\left(-k \left(\frac{C}{A}\right)_v\right) \tag{3}$$

⁴Furhter these are called just agents

⁵[1]Epstein p. 7244

Where V is the agents vision and gives the number of fields which a certain agent can oversee. Because the vision is just limited to a few fields the information which one agent can gain is limited and local. The cop over agents ratio $(\frac{C}{A})_V$ is an important quantity and will become relevant later. K is a constant which is set to $k = \ln(10)$. The habits of the agents are local only. Since each agents calculates its own utility before becoming active, there is no intrinsic value of getting active. An agent gets active on local information only and in no sense having a global social change in mind.

Quantity	Description	Specification	Interval
H	hardship	exogenous, heterogenous	(0,1)
L	legitimacy	exogenous, homogenous	(0,1)
R	risk aversion	exogenous, heterogenous	(0,1)

Cops specification

The cops are even simpler structured than the agents. They have also a vision, which is quantified by v^* . The fundamental cop rule is the following: Inspect all sites within v^* and arrest a random active agent.⁶ Once an agent is arrested it stays in jail for a time which lies in the interval between $(0, J_{Max})$.

4.2 Civil Violence Model II

In this version of the Model Epstein wants to discover a concurrence situation between two different ethnic groups. The two ethnic groups are the famous blues and the keen greens. If an agents gets active means that it kills an agent of the other group. The legitimacy of the Model I is interpreted " to mean each groups assessment of the other s right to exist."⁷ Since there is some death, there must be some birth. Therefore, Epstein introduced a population dynamic in the sense that an agent clones offspring to a randomly chosen field with a certain probability p . The maximal age for one agent is max_{age} . The cops arrest whenever one red agent occurs within their vision.

⁶[1]Epstein p. 7244

⁷[1, Epstein], P. 7248

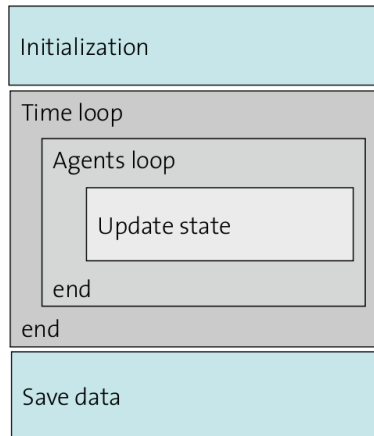


Figure 1: Simulations program structure.

5 Implementation

The implementation of the models was done in MATLAB according to the guidelines given by the course. The original version was done by Epstein in Ascape ⁸ and another one is available in NetLogo ⁹. This is where the ambiguities start, since even if obeying all advices given in the paper, there exists still some degrees of freedom in the models implementations. The code used here is given in app. A. But when comparing to the NetLogo implementation e.g. there are several differences like any cop moves to the patch of the jailed agent or the fact that prisoners stay on the grid.

The work was split up into several tasks. One for each Epstein model and another one for further considerations.

For each configuration or used parameter set a own init file was written, to have an easy way to switch between different setup.

5.1 Outline of the Model I: `ep_main_1.m`

First task was to create Epstein's Model I and verify it's function. The main program structure resembles the default structure for physical simulations as given in fig. 1. The important part is given in *Update*

⁸<http://ascape.sourceforge.net/>

⁹<http://ccl.northwestern.edu/netlogo/models/Rebellion> from Center for Connected Learning (CCL), confer also app. B and [7]

state:

1. movement (rule M) to random site in vision
2. agents: activation (rule A) and ageing
3. cops: arrest random active agent (rule C)
4. prison: release and age agents
5. measurement: data acquisition for later extraction

and was implemented as described by Epstein. As already mentioned the advices were neither complete nor unique. Since the code of Epstein was not available to us we first decided to base our version on that given from last years project [9]. But that code did not fulfill our needs since it was not that compact and we felt it should be able to be done with less code and thus probability more elegant ¹⁰ which would also mean less error-prone. The implementation was done in one single file where possible and roughly optimized for speed by profiling. This is the reason why some code was in-lined again, after moving into a function. The sub function used by this simulation are described in the following.

The iteration over all agents (agent loop) was done over a random permutation of the agent list. That means every agent is updated exactly once but in random order. This assures correct ageing and minimizes effects due to type-writer update scheme.

5.1.1 Add agents: `add_agent.m`

Lot of this code was borrowed from last year's project. However since we used another data structure (with less memory overhead) a lot of the code had to be re-written from scratch. This function can at any point of the simulation add a given number of different agents types and cops to the field (if there is still a free site there). It may be not the fastest implementation but is also rarely called.

5.1.2 Get agents current view: `get_view.m`

Is one of the most important part of the code since it extracts any agents local neighbourhood and the agents within it's sight. First it

¹⁰but may be we don't reached this goal ;)

has to conform with the boundary conditions. For this purpose a function `get_ind.m` was introduced, which returned the corrected indices by help of the modulus function. This code was in-lined again. The next step is to apply the vision mask to the neighbourhood and then get a list of all other present agents.

Further attention has to be payed to the agents vision, confer sec. 5.6 also.

5.1.3 Count agents in environment: `count_agent.m`

A short and simple function (thanks to MATLABs struct indicing facilities) to accumulate all active agents and cops within a given view or sight.

5.1.4 Show plots and measurements: `show_final_plots.m`

This function was written because of the need to switch off any graphical display during extensive calculations. Thus it extracts all important data from measurements done during calculation. This extraction is also time consuming because the data are measured in a way that is fast during calculation but with the drawback of a slow extraction and evaluation. After extraction it displays plots and textual output of the data as well as images if wished to further process into a movie, e.g. In contrast to the original paper and because of a lack in information to evaluate the models correctly we have introduced another screen displaying the sites actual $(C/A)_V$ value, look also at sec. 5.4.

5.1.5 Show debug plots: `show_plots.m`

This code is in big part similar to the previous one, but it is used for debugging when you want to look at the runs in real-time to see 'what is going on'. The two part were separated not to disturb each others (with respect to the programming and execution of the code).

5.2 Outline of the Model II: `ep_main_2.m`

Epstein's Model II is slightly different that Model I. Mainly it introduces different ethnicities and population dynamics. Thus agents gain some new abilities. They can birth, die and kill now. This means for

the implementation essentially that the number of agents is not constant any more and this may have a serious impact on the simulation speed, especially after a few generations. The *Update state* for this algorithm is given here (the 2nd step has changed only):

1. movement (rule M) to random site in vision
2. agents: ageing and dying, birth (with cloning probability), killing (rule A)
3. cops: arrest random active agent (rule C)
4. prison: release and age agents
5. measurement: data acquisition for later extraction

The code implementation was done similar to the first code, but as already mentioned the code was only split up into pieces were it was absolutely necessary. This has some drawbacks, but the big advantage that the code keeps simpler and Models II code was not influenced by Model I (bugs, init conditions, ...). However the few sub-functions already used in Model I were finally also used in Model II.

5.3 Model Variation

Not yet implemented but should have been derived from previous code for the Model II with additional features look at sec. 3 and 7.

5.4 The third screen

In contrast to the original paper and because of a lack in information to evaluate the models correctly we have introduced another screen displaying the sites actual $(C/A)_V$ value, for an example look at fig. 2 (the screen in the lower right corner). We experienced that such a display is much more meaningful then the private grievance screen at least to understand activity outbursts (not for tension measurement e.g.). Some attention has to be payed to the scaling, since the grievance colormap is inverted to make the cops black, so forget about the sign there.

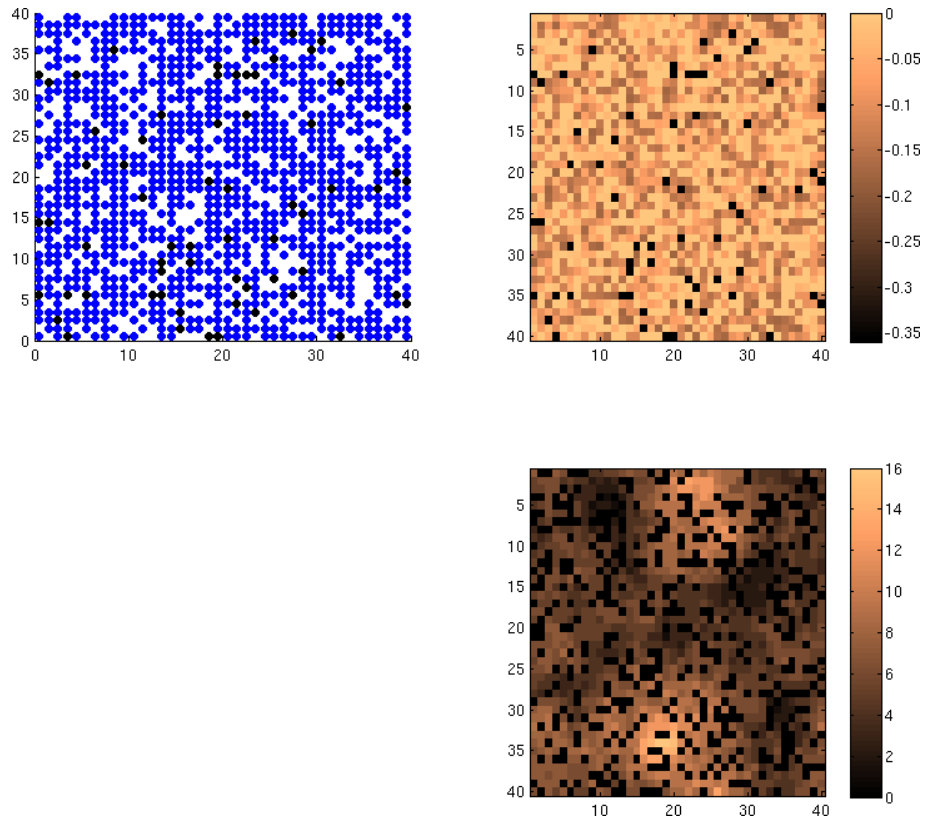


Figure 2: The third screen. Shows the agents local $(C/A)_V$ value.

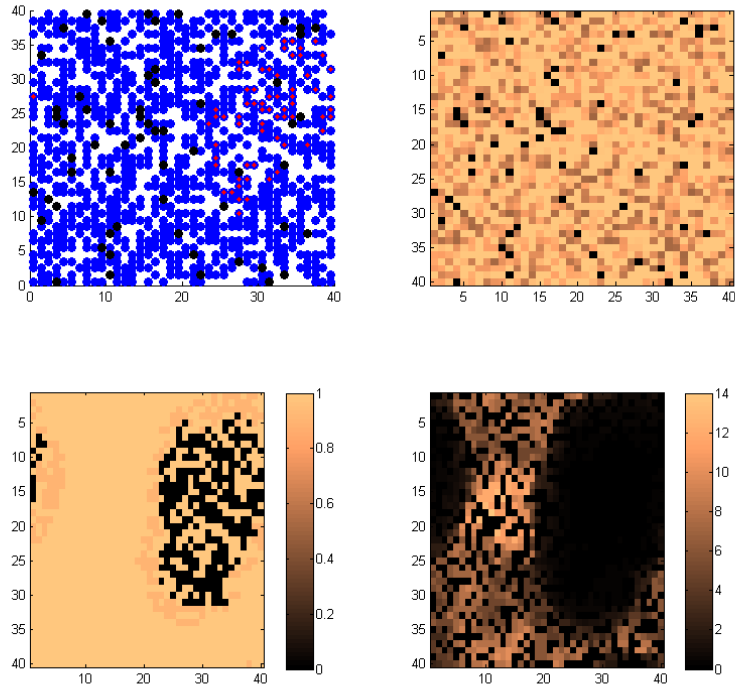


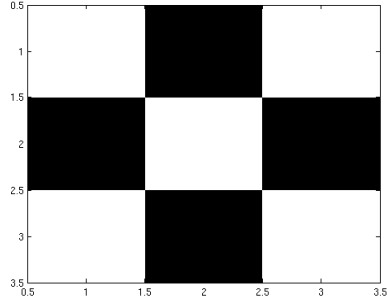
Figure 3: The fourth screen plots the agents local arrest probability value.

5.5 The fourth screen

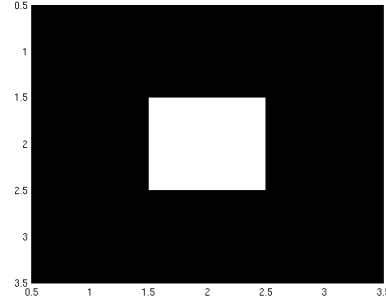
After we discovered that rounding the rounding of the $(C/A)_V$ value is needed to produce punctuated equilibria we introduced a fourth screen which plots the agent's arrest probability according to:

$$P = 1 - \exp\left(-k \cdot \text{round}\left(\frac{C}{A}\right)_V\right) \quad (4)$$

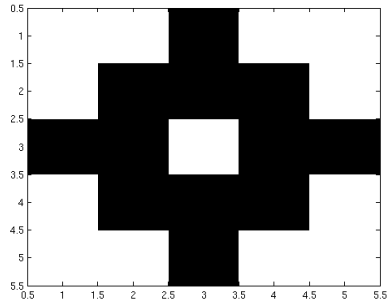
Figure 3 shows an example of the display of the fourth screen in the bottom left corner. The insights of this adapted visualization have been very limited. One thing which is apparent is that the distribution of values for P is not as smooth as the the distribution of $(C/A)_V$ but rather either one or zero.



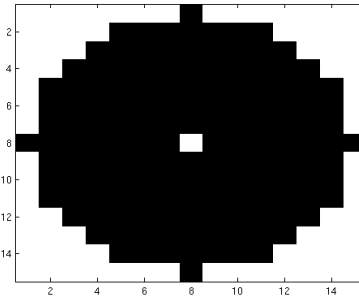
(a) Vision for $v = 1.0$. Von-Neumann neighbourhood 1st order.



(b) Vision for $v = 1.7$. Moore neighbourhood of 1st order.



(c) Vision for $v = 2.0$. Von-Neumann neighbourhood 2nd order.



(d) Vision for $v = 7.0$.

Figure 4: Several vision patterns generated by using all sites that lie within the circle of radius v . The agent itself is always excluded from its sight.

5.6 The agents vision

The vision mask is generated in the main code from a real valued vision range. This has the benefit to be able to generate different vision patterns or masks at ease. Some examples are given in fig. 4. Since this is a very simple (natural) approach it is not such sophisticated as the one used by our predecessors [9]. Also to note is the fact that going from $v = 1.7$ up to $v = 7.0$ means a big increase of calculation power, since for each agents sight an element-wise matrix multiplication has to be done and this size goes $\propto v^2$.

5.7 Consistency checks

The get a real clue and a final result how well this model reproduces and fits together with Epstein's consistency checks had to be carried out. On one hand all (or most) of Epstein's results have tried to be reproduced and on the other hand we compared to the NetLogo implementation, also given. Some examples of this can be seen in the next section (6) and app. B.

6 Simulation Results and Discussion

In the original paper were different parameter sets specified as RUN 1 - 8, were the first 5 are related to Model I. To verify the implementation of our code (since it had to be newly written) we first checked the outcome of those parameter sets.

6.1 Reproduction of the Epstein Model I

6.1.1 Run 1: Individual Deceptive Behaviour

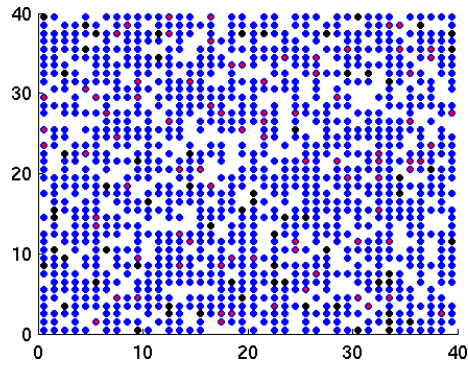
As described and obvious from the agent rule, agents should toggle from active to inactive whilst a cop is close. To investigate this effect, and get a real undistorted look onto such a situation, the best thing is to disable the agents movement and allow it for cops only. The second thing to consider is forget about imprisoning because then the agent has a real chance to get inactive, else it will just be caught by the cop. Look at fig. 5. for an example. Epstein published a movie of this and some other situations investigated in his book [2]. So we did the same and took a movie of such an example¹¹.

6.1.2 Run 2: Free Assembly Catalyzes Rebellious Outbursts

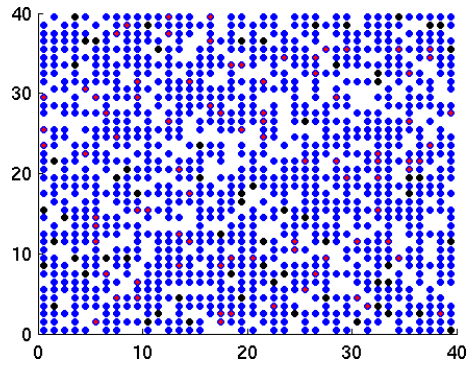
Next thing to check was the occurrence of local outburst or stated in other word the punctuated equilibrium of this metastable system (we search the flipping point). This task turned out to be more tricky than the first and in fact is was only possible to reproduce such a situation within the given parameters with an additional cheat only. As already recognized by [8] and implemented in NetLogo a *rounding* of the cop-to-agent ratio *has to be done!* In this context we have also found that *round* matches better with the parameters given by Epstein than *floor*. Even though there is a tendency for a higher outbreak rate than mentioned by Epstein. For the example shown in fig. 6 and 7 was also a movie created.

It seems that the parameters given here especially the legitimacy is very close to the tipping value of the system, since more or less every value smaller leads to a lot activity.

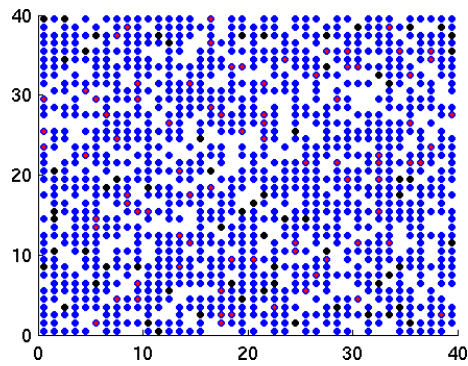
¹¹http://n.ethz.ch/~usoler/download/usoler_civil-viol10/



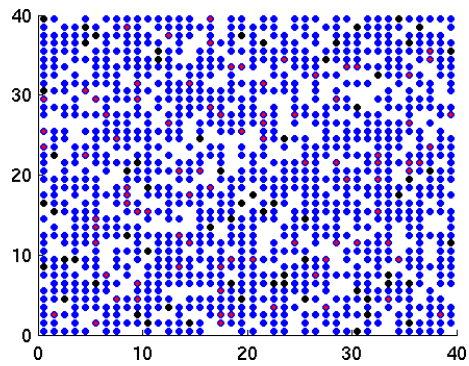
(a) Look at the cop on (15,19) moving to north-east.



(b) Cop on (16,20) and the two agents next to the cop turned inactive whilst one other turned active once the cop moved away.

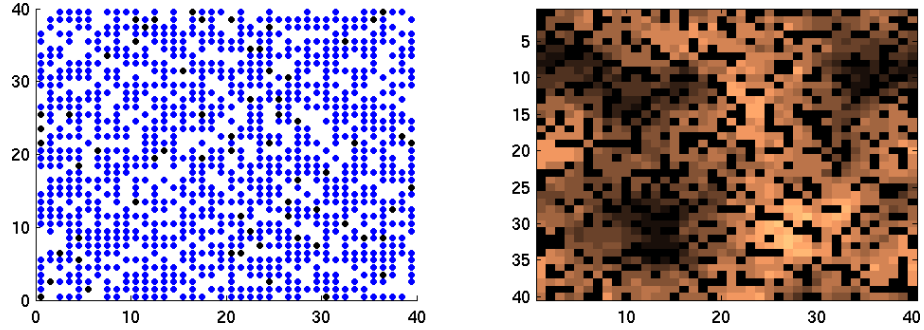


(c) Cop on (17,21) one of the two agents goes active again.

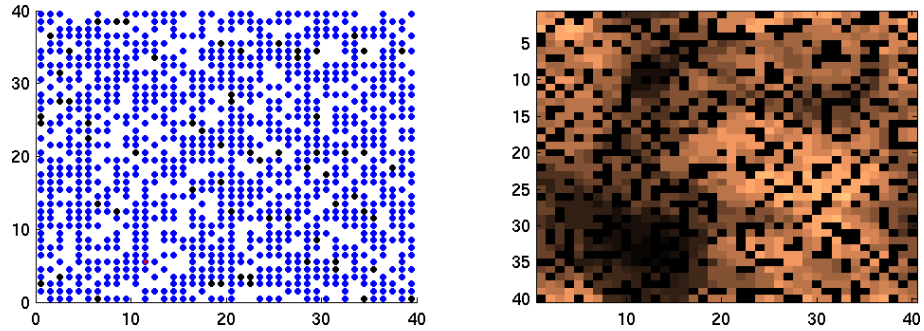


(d) Finally cop on (18,21) and all agents are active again.

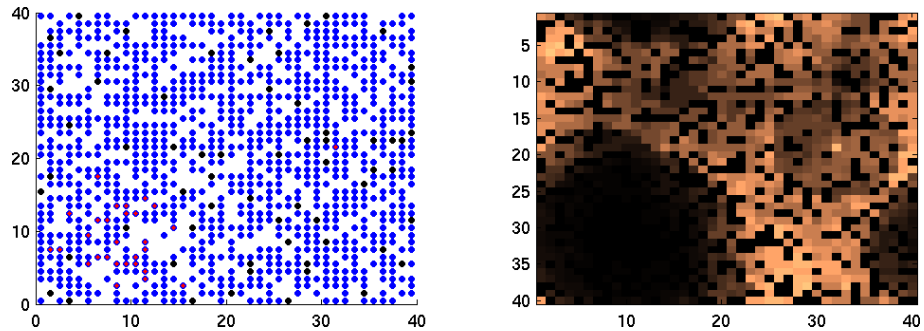
Figure 5: Following the cops path starting on (15,19) one can see 3 different agents changing state, to hide from the cop.



(a) $t = 57$: no activities yet.

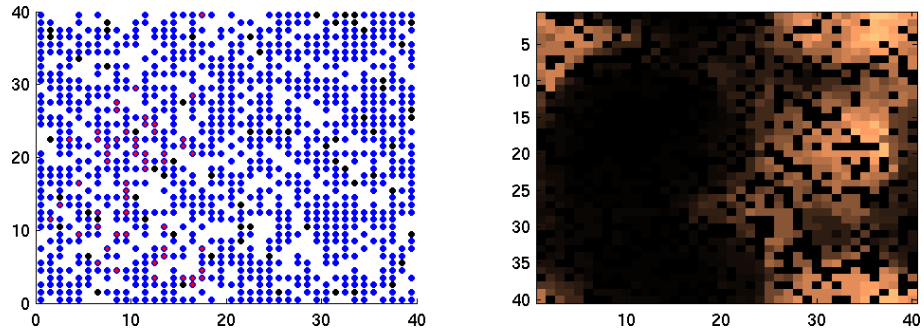


(b) $t = 58$: the seed; someone revolts.

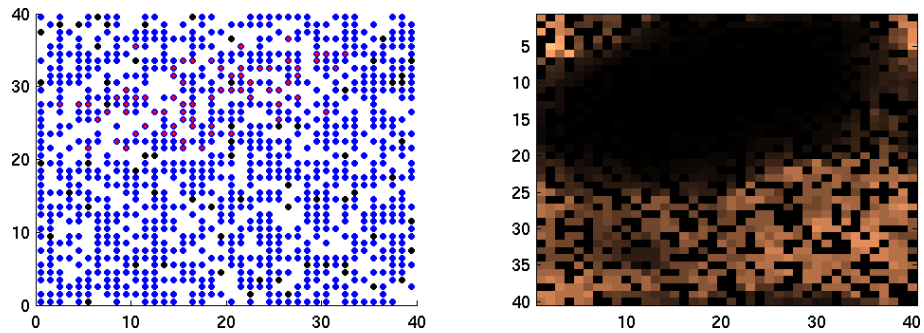


(c) $t = 60$: obviously the time was right, there are others joining.

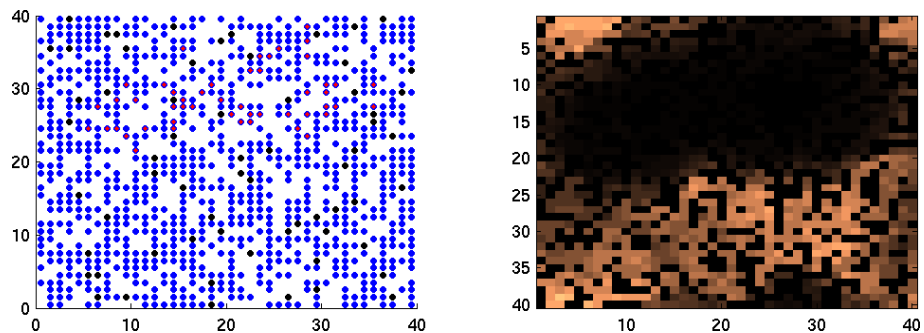
Figure 6: Example of an local outburst propagating through the system. The right screen is the $(C/A)_V$ ratio. Second part is fig. 7.



(a) $t = 62$: the outburst propagates through the system.



(b) $t = 64$: and finally reaches its climax.



(c) $t = 66$: and two time steps after this one (at $t = 68$) the outburst has completely vanished again.

Figure 7: Second part of an local outburst example, first part is fig. 6.

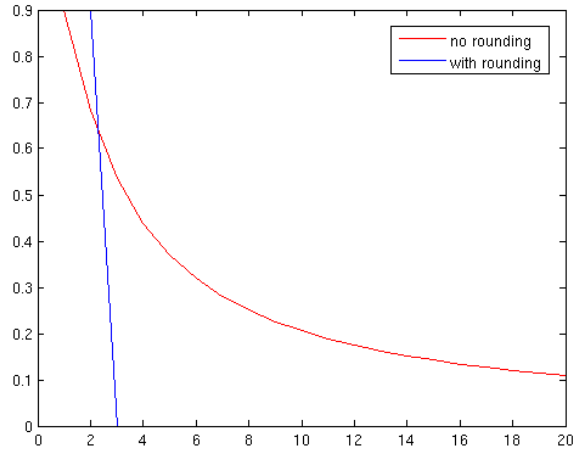


Figure 8: Agents arrest probability P with and w/o rounding.

Important to point out is that *without* using the the rounding trick (either to floor or round) it was not possible to generated outbreaks according to the papers specifications. Epstein defines outburst of 50 actives or more are taken into account only and he has a lot of outbreaks wit up to 200 actives. This can be reproduced with rounding - without only about 20 to 50 actives were observed at once. Further investigation of the given relations, look at fig. 8, shows that in fact this exponential function together with round directly can be replaced by a simple condition. Since as you can see the value simply switches as soon as there are more active agents than cops. In other words this introduces another tipping point, threshold or step function and furthermore can be replaced by a simple condition, as noted elsewhere [8].

If $C < (A - 1)$ then $P = 0$; otherwise $P = 0.9$.

(the minus one is needed since every agent counts itself also, as noted by Epstein). This improved algorithm will also be faster than always evaluating an exponential function. The fourth screen also shows this behaviour as only zeros and values close to one are observed. This poses the first questions on whether or not the model is able to provide valuable insight into real life behavioural patterns in the context of civil violence. The fourth screen also reveals that there are

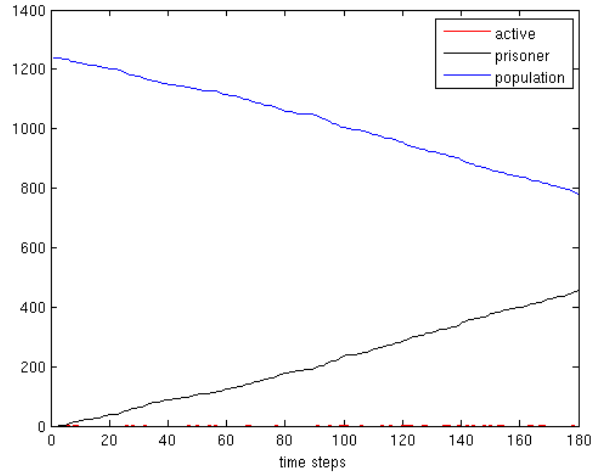


Figure 9: Salami tactics linear legitimacy reduction (from 0.9 to 0.0), leads to total ~ 450 prisoners.

very often agents present with $P = 0$. If their grievance is high enough they will go active and act as the famous spark cited by Eppstein et al. in the catalysis of an outbreak of civil violence [1]. To summarize: The punctuated equilibria could only be reproduced by modifying the computation of the arrest probability described by Eppstein et al. to the one explained above.

6.1.3 Run 3: Salami Tactics of Corruption (linear decrease)

This concept is interesting and was investigated also, first the linear decrease. Our results are shown in fig. 9. This value matches with the one given in the paper which is also ~ 400 but the legitimacy reduction here was done a little bit faster, within 180 steps instead of 260 like in the paper.

...

6.1.4 Run 4: Salami Tactics of Corruption (sudden drop)

The sudden drop case is an ambiguous point in the paper again. Since in the text is written that a drop from 0.9 to 0.7 is performed but it is also stated that this equals a drop of 0.3. Furthermore in the according

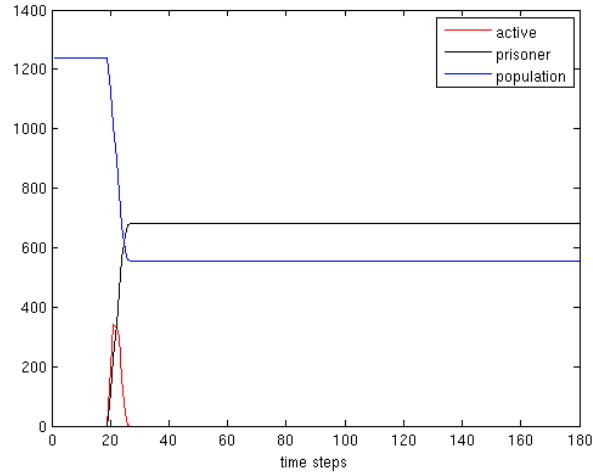


Figure 10: Salami tactics sudden legitimacy drop (from 0.9 to 0.4), leads to total ~ 650 prisoners.

figure a drop to 0.65 is plotted. Thus the real params are not obvious and in the table the runs 3 and 4 are listed together. The example given in fig. 10 was done for a drop from 0.9 to 0.4 (which is still > 0.0) and shows the effect very nicely.

...

6.2 Reproduction of the Epstein Model II

This model, even more complexity included was simpler to reproduce. For example the *round* was not needed. Further investigation even showed a smaller impact of this change onto Model II. An important change however is the fact that only agents that have gone active and killed another one are further treated as active. Of all those measurements also a movie has been taken. No parameter scan were performed, since we focused on the first model.

6.2.1 Run 6: Peaceful Coexistence

The peaceful coexistence was not hard to implement, it is just a question of an appropriate high legitimacy value. The result can be seen in fig. 11.

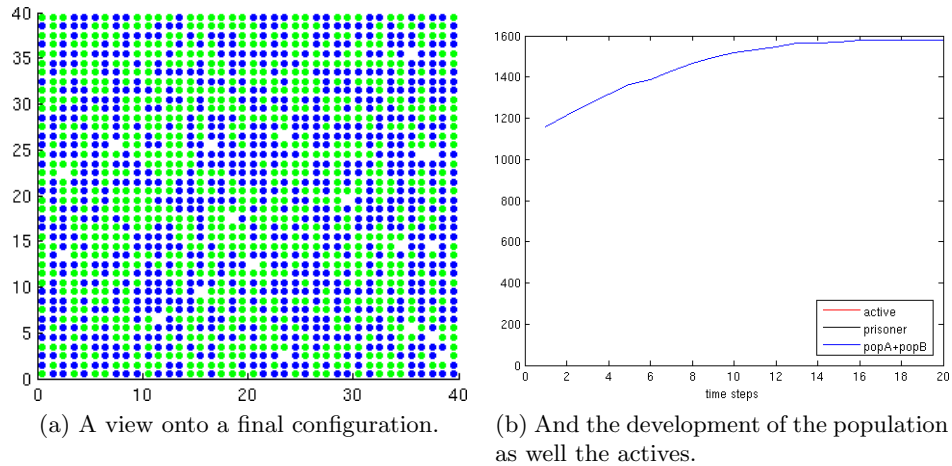


Figure 11: An example of peaceful coexistence without cops.

It seems that the legitimacy given in this case is again very close to the tipping value of the system, since more or less every value smaller leads to a lot activity.

...

6.2.2 Run 7: Ethnic Cleansing

Ethnic Cleansing seems to be among one of the basic need, at least following this model. Since this case was also implemented at ease, like can be seen from fig. 12.

...

6.2.3 Run 8: Safe Havens

In action to reproduce the safe havens, we want to doubt that a cloning rate of 0.05 was used, also when considering the movies given in [2]. Trying to emerge safe havens with this rate failed. On the other hand if increasing the cloning rate up to ~ 0.30 it works. The final state an the measurements are showed in fig. 13.

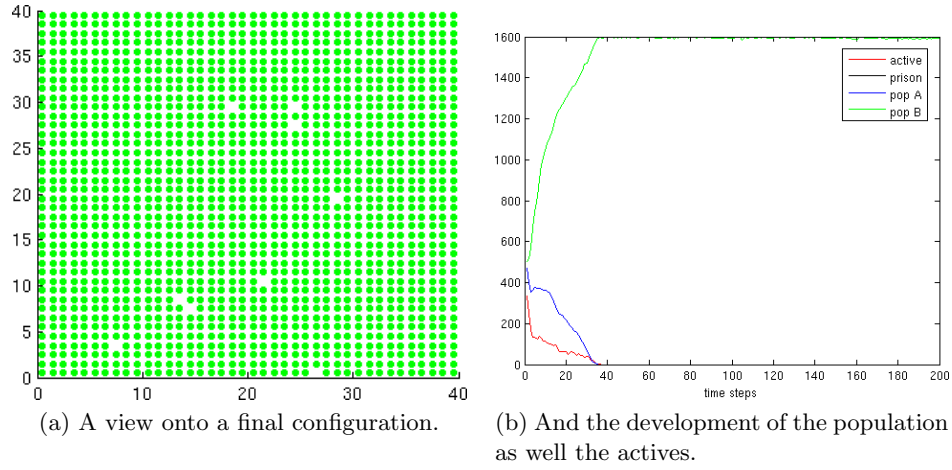


Figure 12: An example of any ethnic cleansing between two groups.

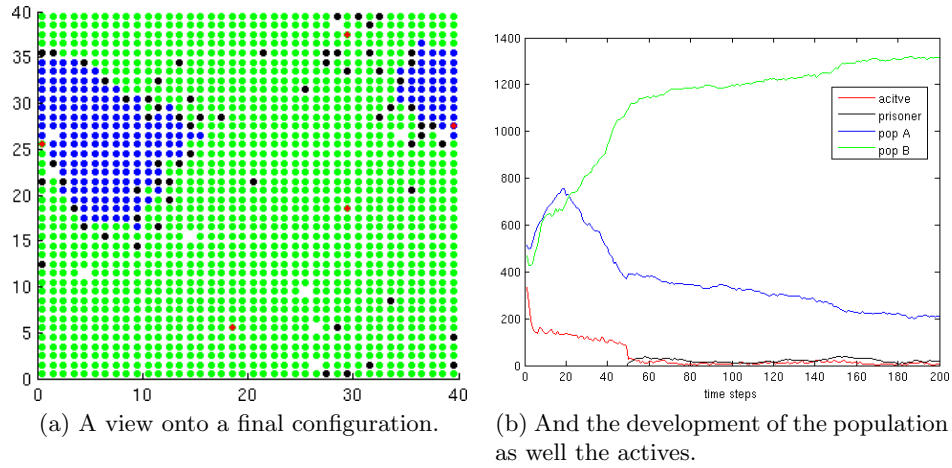


Figure 13: An example of more or less peaceful coexistence with cops.

6.2.4 The Northwestern Trick

In the beginning we had serious problems in reproducing the results which Epstein has produced with his model ¹². Although, the dynamics were quite the same there were no outbursts like Epstein has discovered. Thanks to Giovanni Luca Ciampaglia we came upon the hint given by <http://ccl.northwestern.edu/netlogo/models/Rebellion>. They reproduced the Epstein model in NetLogo and had in the beginning the same problems we had. Finally they changed the agent rule into the following sense:

$$P = 1 - \exp\left(-k \cdot \text{floor}\left(\frac{C}{A}\right)_v\right) \quad (5)$$

Whereas, the floor function rounds the cop-to-agent ratio to the next lower integer. We tried both the floor and the round function (which rounds to the next closest integer). Implemented this the dynamics became much more appropriate and closer to the Epstein results. Well, in the beginning it was just a hint taken from the internet. But at a second glance we tried to look a bit closer.

If you use the floor function the arrest probability calculated by each agent becomes to $\left(\frac{C}{A}\right)_v \leq 1$. The agent rule as a whole becomes a step function. The meaning of this is if there is $A > C$ the arrest probability would be 0. On the other hand if there were $A < C$ the arrest probability would be equal to 1. The entire agent rule looks then like (if the agents become active):

$$\left(H * (1 - L)\right) - R * \left(1 - \exp\left(-k \cdot \text{floor}\left(\frac{C}{A}\right)_v\right)\right) \geq T \quad (6)$$

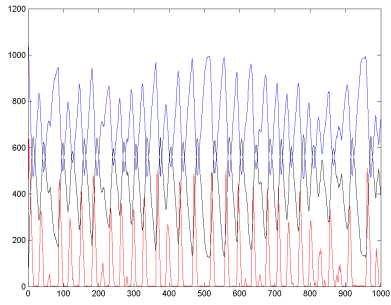
6.2.5 Parameter scan

In this section we do the parameter scans. The first scan varies the Legitimacy, *ceteris paribus*.

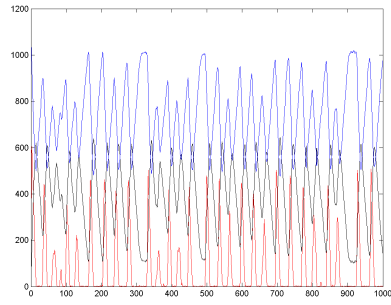
1. Legitimacy scan

In the following you can see a snippet of the figures of the Legitimacy scans.

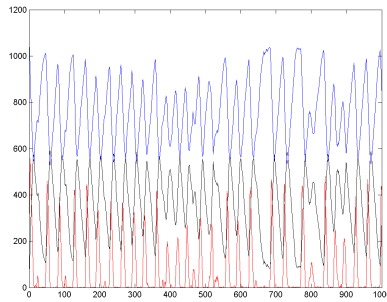
¹²e.g. punctuated equilibria



(a) Legitimacy scan for $L = 0.60$



(b) Legitimacy scan for $L = 0.65$



(c) Legitimacy scan for $L = 0.70$

Figure 14: Selected results from the legitimacy parameter scan.

They point out very clearly the behavior called *punctuated equilibria*¹³ which is supposed to be “one of the hallmarks of complex systems“[1, Epstein]. More or less long lasting periods of relative peace were disrupted by short and by the same time extremely high peaks the so-called *outbursts*. Epstein set the threshold for an outburst to at least 50 actives.

We follow Epstein in the sense that we are interested in the *Waiting time distribution* between outbursts. The following statistics had been made with respect to the active agents only although we have gathered at huge amount of data. In the following we analyse the behavior of the outbursts which you can see in the figure 14 where we have plotted the active agents and the imprisoned agents with respect to time. You can observe the clearly oscillating behavior. In the following we will quantify this oscillating behavior, following Epstein.

In order to calculate the waiting time distribution¹⁴ we wrote a short function which counts the time between the certain outbursts and the duration of an outburst. In figure 15 you can see a time serie plot for $L = 0.80$ which shows a typical oscillating behavior. The red colored sequences are the one of interest¹⁵. For this example the mean duration of an outburst is $d_{mean} = 5.0741$ steps with a Standard deviation of $\sigma_d = 2.0177$. The mean time between two outburst (the waiting time), the waiting time is $w_{mean} = 30.5357$ steps with a Standard deviation of $\sigma_w = 16.2856$. The peaks of outburst are very steep and alter in their altitude on a range of 5 magnitudes. Besides their different height the duration of an outburst is very similar of each peak. Comparing to the waiting time the standard deviation of the outburst time is relatively smaller. The durations of outbursts have in most cases relatively equal. The durations of the time between two following outbursts differs much more compared to the outburst time. This can be seen from the following figures: It is quite obvious from fig. 16 that the outburst time is much more peaked than the waiting time. These results are in line with Epsteins investigations. The waiting times are almost equal, we find the maximum of waiting time at 24.88. Like Epstein suggests¹⁶ a Weibull or a Lognormal distribution. In fig. 17 we zoomed in the

¹³Further reading at [1, Epstein], Page 7245 ff.

¹⁴“Frequency to Bins”, [1, Epstein P.7246]

¹⁵if at least 50 actives are involved

¹⁶P.7245

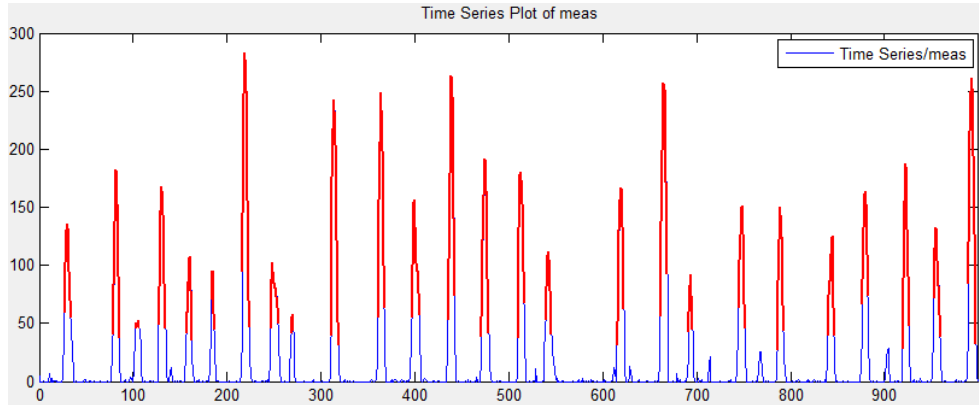
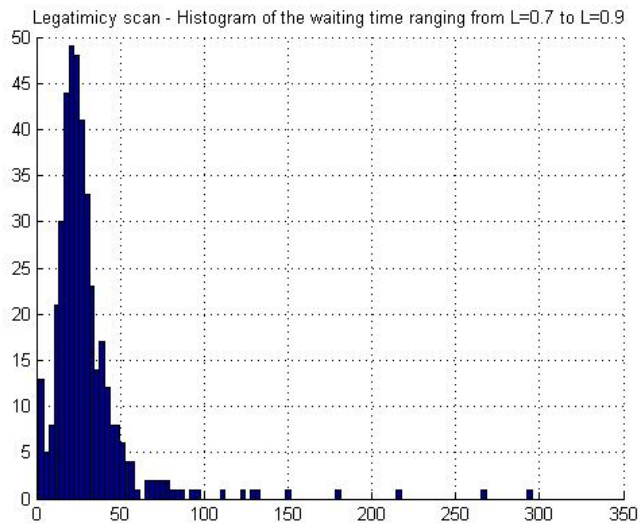


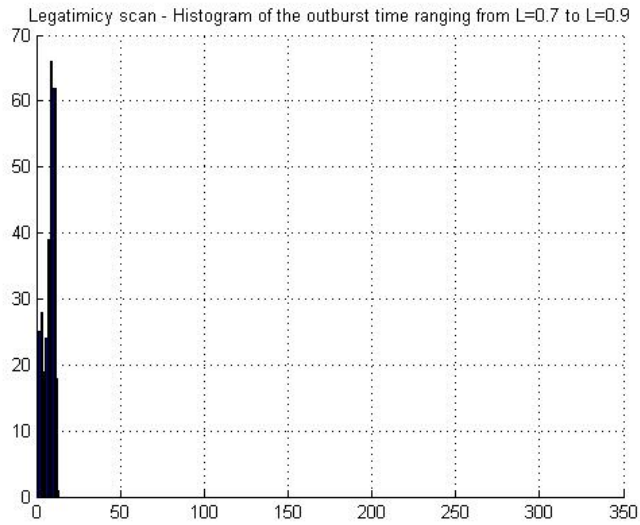
Figure 15: Time serie plot of legitimacy scan for a legitimacy $L = 0.80$

region of the steep peak of the outburst time distribution. But there can not be extracted much more information. In order to explore the distributions in more detail we plotted both the mean waiting time and the mean outburst time with respect to L . This can be seen in fig. 18. You can easily see that for low values of Legitimacy the slope is constant. At the critical point, which is approximately at $L = 0.83$, the waiting time explodes. On the other side, the mean outburst time does not show such an explosive behavior. From a sociological point of view this behavior could be explained with help of a critical legitimacy threshold. Epstein pulls this together with *social breakdowns*. Above the threshold (high loyalty of the citizen) the system is stable and the probability of an outburst is small. Below the threshold the state forces can not supress the agrieved citizens and outburst occur more often. The interesting thing is not which fit or which analytical distribution describes the situation best. The crucial part is the quite “powerful statistical regularity”¹⁷ underlying the dynamics. We can support this view and we can vaildate his findings.

¹⁷Epstein, P.7246



(a) Waiting time distribution, Legitimacy scan



(b) Outburst time distribution, Legitimacy scan

Figure 16: Waiting time distribution and outbursts.

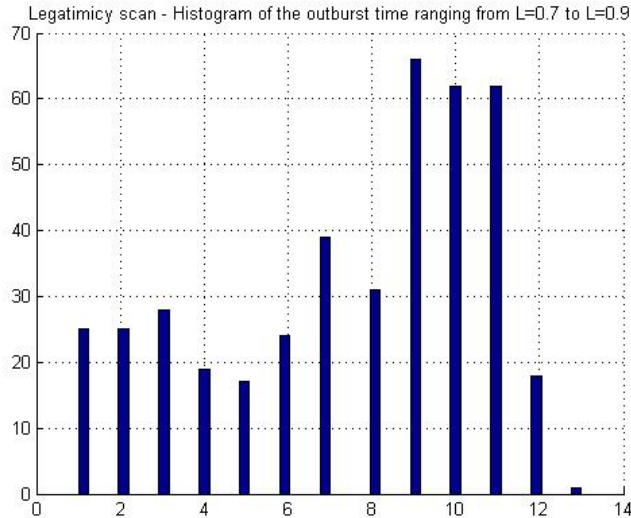


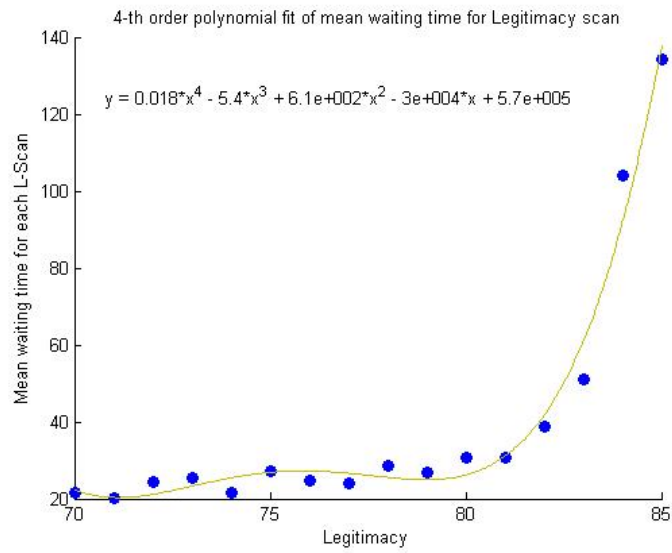
Figure 17: Zoom in maximum region of the outburst time distribution

2. Vision scan

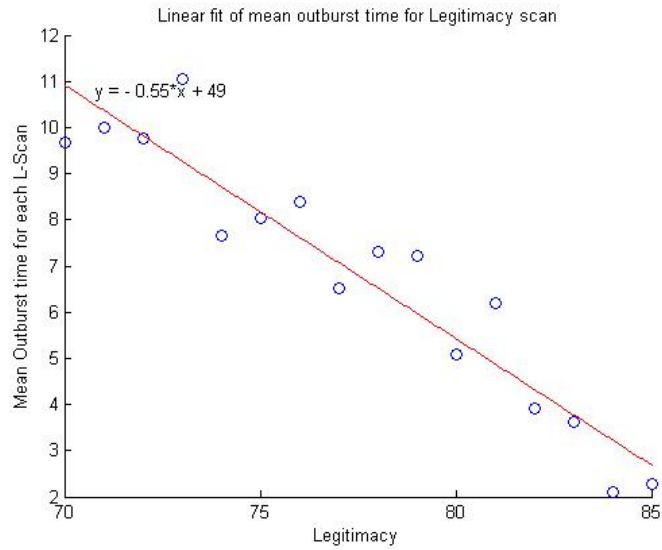
In the previous section we followed Epstein in his approach. The vision is an exogenous parameter on the region $(0; 9)$. A parameter scan of the vision in an interval from $v = 0$ where agents and cops have no vision to a value of $v = 9$ where full information is available to cops and agents. The step size was chosen to be 0.5. A selection of activity pattern at different vision values is depicted in figure 19. Without any information a constant number of agents is active. If the agents grievance computed in the beginning of the run is higher than $T = 0.1$ the agent goes active and stays active since the vision of the police is zero as well and no agents are jailed. With local information activity stays high until a certain threshold is reached and the emergence of punctuated equilibria is observed. More information leads to less but more severe outbursts.

Transforming this finding to a real world scenario little information can be regarded as difficult terrain with hideouts and bad reach by the central power (here the cops) which has been shown to have significant correlation with the outbursts of civil violence [4].

In fig. 20 you can see a different result than in the Legitimacy

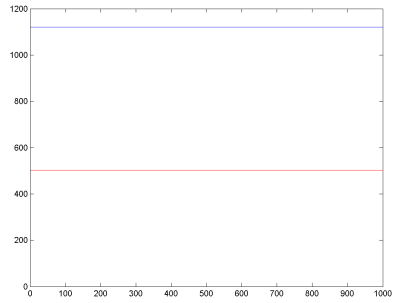


(a) Mean waiting time distribution with fit

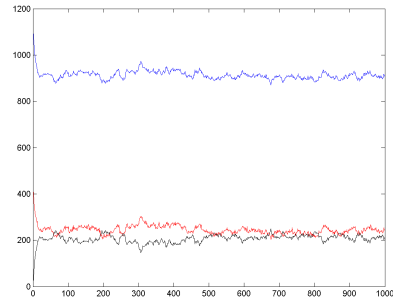


(b) Mean outburst time distribution with fit

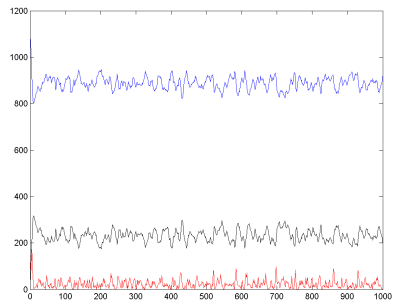
Figure 18: Fitting of waiting time distribution and outburst.



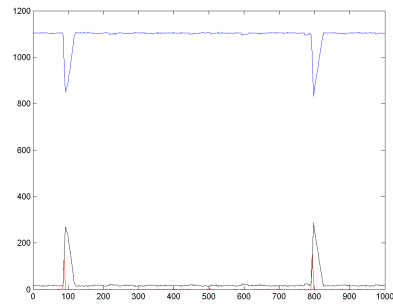
(a) $v = 0$



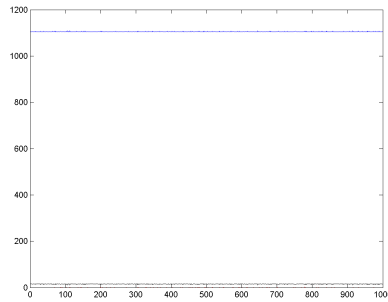
(b) $v = 1$



(c) $v = 5$

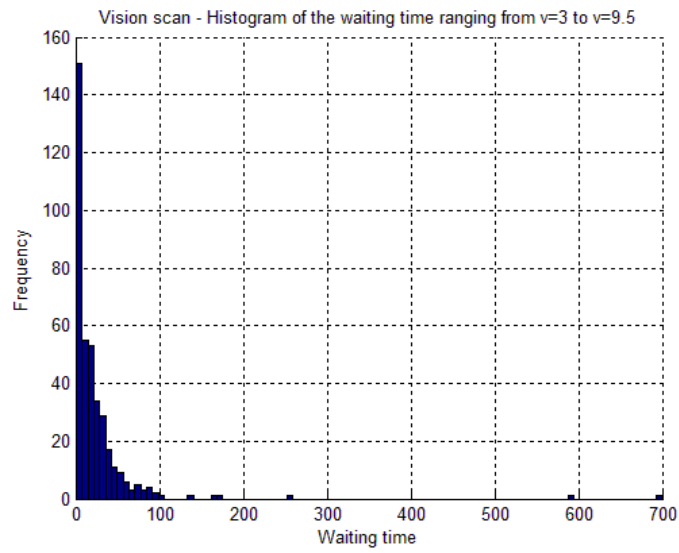


(d) $v = 8$

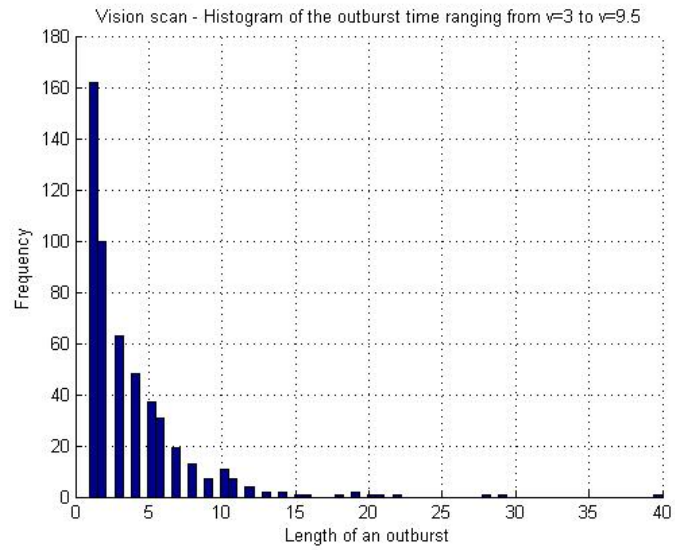


(e) $v = 10$

Figure 19: Selected activity patterns for varying vision and $L = 0.82$.



(a) Waiting time distribution, vision scan



(b) Outburst time distribution, vision scan

Figure 20: Waiting time distribution and outbursts.

section. The distributions look very similar and they have both an envelope of an exponentially decaying function. Continuing the successful approach in the previous section we plotted again the mean time distributions in fig. 21.

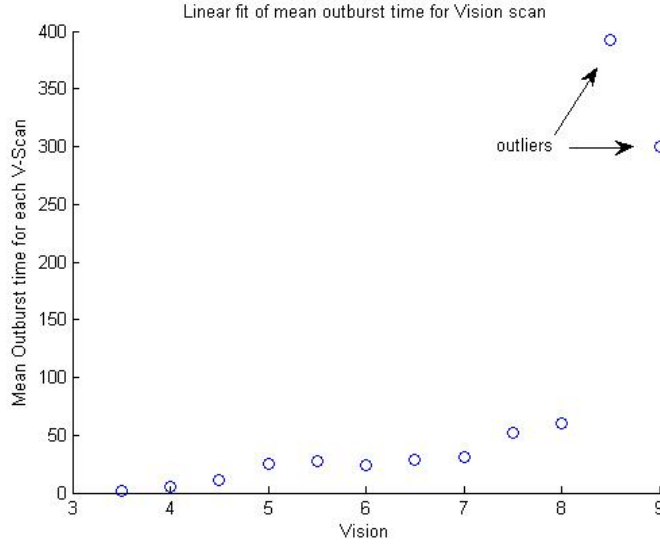


Figure 21: Mean outburst time with outliers

But for the vision scan we observe a linear behavior below the threshold of a vision about 8. Then the outbursttime explodes. This can be explained by the greater range of the vision of an agent. Because it sees much more the possibility is much higher that there is a cop within its vision. Below the threshold the behavior is linear as it can be seen in fig. 22.

The waiting time does not show any regular behavior. The outburst time can be easily fitted by a linear function.²³

6.3 Model II

Following the good results gained from the floor function in Model I we tried to integrate the modified agent rule to Model II, as well. According to the above section we scanned the Model II with respect to the legitimacy. The result was a very steep transition from the non-rebellions state to a rebellios state. The transition occured a value

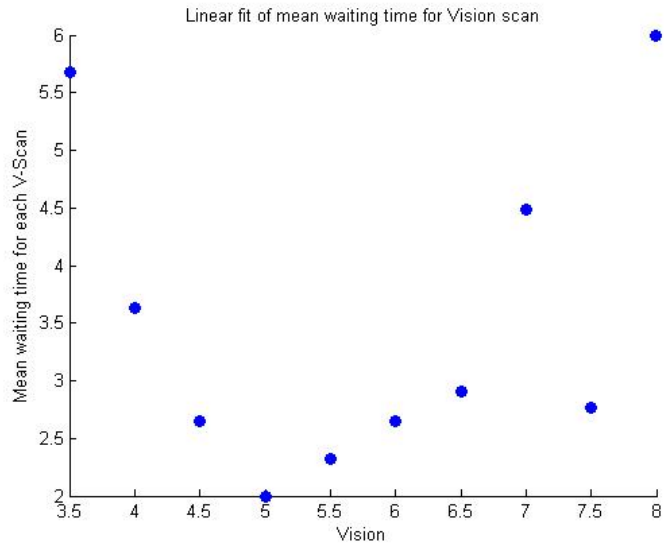


Figure 22: Mean waiting time distribution with fit

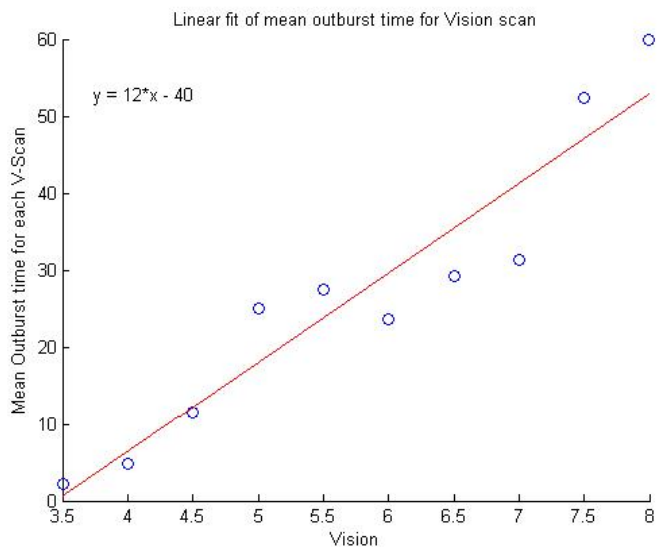


Figure 23: Mean outburst time distribution with fit

for the legitimacy of $L = 0.895$ and is a real steeply increasing step function. After all, the floor function did not change much of the behaviour of Model II.

7 Summary and Outlook

7.1 Comparison to the Epstein Model: Summary and Conclusions

- Implement the general Epstein model and test it by varying several parameters.
- Recognize the limitations and significances of the Epstein modeling approach in a qualitative manner.

7.2 Outlook

7.2.1 Methods

- Modify the police or the state force in the Epstein model to have a bias. A method for a more sophisticated modeling of the police has to be elaborated. Parameter variation analysis to test the models sensitivity to police bias and injustice. A possibility to model the police could be to introduce a parameter model similar to the one used to model the agents of the different ethnicities.
- According to Wimmer et al. in this approach only the special case of a rebellion is considered and attempted to be modeled.
- Differentiate between the ethnic groups and introduce different levels of grievance, state acceptance and so forth.
- Endogenizing of external parameters as L, J or T
- The assumption that agents leave the prison as aggrieved as they entered could be loosen.
- The definition of at least 50 agents performing an outburst is quite strict. This threshold is discussable.
- Introduce field parameters to account for a more likely scenario. Adjust the grid of the model to account for a more realistic country.
- The NetLogo implementation gives also some hints on “things to try” e.g.

7.2.2 Goals

- Account for the reality where the police force is never such a homogeneous or fair body. Attempt to confirm the findings of Wimmer et al. who claim the share in state power is the main reason for civil violence.
- Find a stable solution where a minority group holds power (police favors minority group) and a majority which is out of power. Be able to make a statement about the robustness of such a solution by the means of the parameter analysis.
- Discovering a measure for emergent phenomena [1, Epstein, P. 7245]
- Furthergoing investigation of *social breakdowns*. Epstein compares the social breakdowns to failure rates of electrical and mechanical systems [1, Epstein, P. 7246]. An approach could be the endogenizing of parameters and the exploration of their sensitivity concerning the model.
- Understand the effects of heterogeneity of grievance and state acceptance among different ethnic groups.
- Make a statement about the effect of a more realistic grid shape and size to the model.

8 References

References

- [1] Joshua M. Eppstein; Modeling civil violence: An agent-based computational approach (pnas.org/cgi/doi/10.1073/pnas.092080299, 2002)
- [2] Joshua M. Eppstein; Generative Social Science: Studies in agent-based computational modeling (Princeton University Press, 2002)
- [3] Lars-Erik Cederman (ETH Zurich), Adreas Wimmer (UCLA), Brian Min (UCLA); Ethnic Politics and Armed Conflict: A Configurational Analysis of a New Global Data Set (American Sociological Review, 2009)
- [4] James D. Fearon and David D. Laitin; Ethnicity, Insurgency and Civil War (American Political Science Review 2003)
- [5] Timothy R. Gulden; Spatial and temporal patterns in civil violence - Guatemala, 1977-1986 (Politics and the life sciences, 2002)
- [6] Modeling the Size of Wars: From Billard Balls to Sandpiles (American Political Science Review, 2003)
- [7] Wilensky, U.; NetLogo (Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/>, 1999)
- [8] Wilensky, U.; NetLogo Rebellion model (Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/Rebellion>, 2004)
- [9] Sam Mundy, Tim Oberhauser; Modeling Civil Violence (http://www.soms.ethz.ch/teaching/MatlabFall10/projects/CivilViolence_Mundy_Oberhauser.pdf, 2010)

A MATLAB Code

Available from https://svn.vis.ethz.ch/svn/usoler_civil-viol10/
by svn.

A.1 Algorithm

```
% EPSTEIN MODEL I
%
% need to verify: [ok] Fig. 1
% (reproduce)      [ok] Individual Deceptive Behavior / p.3
%                (without jail and non-cop movement)
%                [ok] Free Assembly Catalyzes Rebellious Outbursts / p.3
%                (equilibrium)
%                [ok] Salami Tactics of Corruption / p.5
%                (but other legitimacy (is an outburst effect))
%                - Cop Reductions / p.5
%                - all the other figures...
%                - ( Stylized Facts Generated in Model I / p.6 )
%
% DONE:
% (see main_ep_2.m for mor info)
%
%
%   Date:
%     $Id: main_ep_1.m 46 2010-12-05 13:33:51Z usoler $
%   Author:
%     Grimm Alexander, Koller Michael, Solèr Ursin
%     Mundy Sam, Oberhauser Tim (some code snippets)

clear;
clc;

% init needed vars
global gridsize vis ag map max_age unique_ag_ID legitimacy

% RUN 1; Individual Deceptive Behavior
% AND DEACTIVATE MOTION FOR NON-COPS, DEACTIVATE ARRESTING
%init_ep_1_run_01;

% RUN 1; Fig. 1 / test set with lower legitimacy
% AND DEACTIVATE MOTION
%init_ep_1_run_01_test_01;

% RUN 2
%init_ep_1_run_02;

% RUN 2; test lower legitimacy
%init_ep_1_run_02_test_01;

% RUN 2; Fig. 2 / Free Assembly Catalyzes Rebellious Outbursts
%init_ep_1_parascan_65;
%config = {'init_ep_1_parascan_65'};

% RUN 3+4; Salami Tactics of Corruption
% AND ACTIVATE LEGITIMACY REDUCTION
%init_ep_1_run_03;
config = {'init_ep_1_run_03'};

% RUN 5
```

```
%init_ep_1_run_05;

stat_count = 1; % (not used yet)

for param = config
    % run init script and init vars
    param
    close all;
    j = batch(char(param));
    wait(j);
    load(j);

    % setup vision from radius
    vis.max = floor(vis.real);
    vis.origin = vis.max+1;
    vis.size = (2*vis.origin-1) * ones(1,2);
    vis.pattern = zeros(vis.size);
    for i = 1:vis.size(1)
        for j = 1:vis.size(2)
            vis.pattern(i,j) = ...
                (sqrt( (i-vis.origin)^2 + (j-vis.origin)^2 )<=vis.real);
        end
    end
    vis.pattern(vis.origin,vis.origin) = 0;
    %clear vis.origin;
    %imagesc(vis.pattern)

    unique_ag_ID = 1;

    % statistics loop
    %figure('Position',[0 0 900 400])
    clear meas_ag meas_prison;
    meas_map = zeros([stat_count step_max_count gridsize]);
    for stat = 1:stat_count
        % create/place agents
        ag = []; % agents (A, cop)
        map = zeros(gridsize); % map/grid
        prison = []; % prison
        max_age = 0; % not used in this model (!)
        add_agent([numA],numC);

        % simulation loop
        for step = 1:step_max_count
            step

            % legitimacy reduction, comment out if not RUN 3/4
            % if step == 20
            %     legitimacy = 0.40; % text says 0.7 diagramm 0.65,
            %         % need 0.45 for 660 prisoners
            % end
            % legitimacy = legitimacy - 0.005; % leads to about 500 pris.
        end
    end
end
```

```

% activation and ageing in the free world
for ag_i = randperm(length(ag)) % rand. but each just once
% skip prisoners and deaths
if (ag(ag_i).type==0)
    continue
end

% agent pos
pos = ag(ag_i).position;

% movement to random site in vision
% find free position within agent sight
%if ag(ag_i).type == -1 % for RUN 1
[~,~,~,neighbors] = get_view(map, pos, vis, gridsize);
free=(neighbors == 0);
newsite = free .* vis.pattern .* rand(vis.size);
m=max(max(newsite));
if (m~=0)
    [y,x]= find( newsite==m );
    newpos = get_ind( pos + [y x] - vis.origin );
    ag(ag_i).position = newpos;
    map(pos(1),pos(2)) = 0;
    map(newpos(1),newpos(2)) = ag(ag_i).ID;
end
%end

% agent new pos
pos = ag(ag_i).position;

% agent sight
[~,~,view,neighbors] = get_view(map, pos, vis, gridsize);

if ag(ag_i).type == 1
%     % for legitimacy jump re-evaluate grievance
%     % comment out if NOT RUN 3/4
%     ag(ag_i).grievance=ag(ag_i).hardship*(1-
legitimacy);

% count all in sight
[cops,actives] = count_agent(view, ag);

% activate agents
%P = 1 - exp(-k*((cops+1)/(actives+1)));
% necessary for puncuated equilibrium
%http://ccl.northwestern.edu/netlogo/models/Rebellion
P = 1 - exp(-k*round(cops/(actives+1)));
%P = 1 - exp(-k*floor(cops/(actives+1)));
%N = ag(ag_i).risk_aversion * P * ag(ag_i).jail_term;
N = ag(ag_i).risk_aversion * P;
if (ag(ag_i).grievance - N) > threshold
    ag(ag_i).active = 1;
else
    ag(ag_i).active = 0;
end

```

```

    % age
    ag(ag_i).age = ag(ag_i).age + 1;
elseif ag(ag_i).type == -1
    % arrest rand. active in sight, comment out for RUN 1
    active=[ag(view).active];
    vis.ind=find(active);
    if vis.ind
        i=view( vis.ind( randi(length(vis.ind)) ) );

        % put to jail
        ag(i).jail_term = randi([0,jail_term_max]);
        ag(i).jail_time = 0;
        %prison(end+1) = ag(i);
        prison(end+1).position = ag(i).position;
        prison(end).active = ag(i).active;
        prison(end).hardship = ag(i).hardship;
        prison(end).grievance = ag(i).grievance;
        prison(end).criminal = ag(i).criminal;
        prison(end).age = ag(i).age;
        prison(end).risk_aversion = ag(i).risk_aversion;
        prison(end).jail_term = ag(i).jail_term;
        prison(end).jail_time = ag(i).jail_time;
        prison(end).ID = ag(i).ID;
        prison(end).type = ag(i).type;
        prison(end).max_age = ag(i).max_age;
        ag(i).type = 0;
        map( map==i ) = 0;
    end
end
end

% release and ageing in prison
free_ag = [];
for ag_i = 1:length(prison)
    % age
    prison(ag_i).age = prison(ag_i).age + 1;

    % release from prison
    if prison(ag_i).jail_time >= prison(ag_i).jail_term
        % free it
        ag( prison(ag_i).ID ) = prison(ag_i);

        % place it on the grid
        free = find(map==0);
        i = free( randi(length(free)) );
        map( i ) = prison(ag_i).ID;
        [y,x] = ind2sub(gridsize,i);
        ag( prison(ag_i).ID ).position = [y x];
        ag( prison(ag_i).ID ).active = 0;

        % remove from jail
        free_ag(end+1) = ag_i;
    end
end

% jail time

```



```
        prison(ag_i).jail_time = prison(ag_i).jail_time + 1;
    end
    prison( free_ag ) = [];

    % OUTPUT and display
    %show_plots();

    % MEASSUREMENT (store data for later use)
    meas_map(stat,step,,:) = map;
    meas_ag(stat,step,:) = ag(:);
    meas_prison(stat,step,:) = {prison(:)};
end

save( ['dat/',...
      'data_',char(param),'_',sprintf('%04i',stat),'.mat' ],...
      'meas_map', 'meas_ag', 'meas_prison');
%save( ['dat/',
      'data_',char(param),'_',sprintf('%04i',stat),'.mat' ] );
show_final_plots( squeeze(meas_map(stat, :, :, :)), ...
                  squeeze(meas_ag(stat, :, :)), ...
                  meas_prison(stat, :, : ) );
end
end
```

```

function [] = add_agent(numA,numC)
%ADD_AGENT Add agents to the simulation.
%   ADD_AGENT(numA,numC) add numA different agent types and, numC cops
%   to the map and agent list ready to be used.
%
%   Date:
%       $Id: add_agent.m 38 2010-11-26 23:05:54Z usoler $
%   Author:
%       Grimm Alexander, Koller Michael, Solèr Ursin
%       Mundy Sam, Oberhauser Tim (some code snippets)

global ag map gridsize legitimacy max_age unique_ag_ID

% create/place agents (COULD MAY BE DONE SMARTER)
free=(map==0);
placement=rand(gridsize) .* free;
hardship=rand(gridsize);
risk_aversion=rand(gridsize);
for type=1:length(numA)
    for i=1:numA(type)
        if isempty(find(placement, 1))
            break
        end
        [unn,y]=max(placement);
        [~,x]=max(unn);

        %ag(ind)           = new_ag;
        ag(end+1).position = [y(x) x];
        ag(end).active     = 0;
        ag(end).hardship   = hardship(y(x),x);
        ag(end).grievance  = ag(end).hardship * (1-legitimacy);
        ag(end).criminal   = 0;
        ag(end).age        = 0;
        ag(end).risk_aversion = risk_aversion(y(x),x); % fixed for life
        ag(end).jail_term  = 1; % neutral
        ag(end).jail_time  = 0;
        ag(end).ID         = unique_ag_ID;
        ag(end).type       = type; % -1=cop,0=removed,1...=agents A...
        ag(end).max_age    = randi([0,max_age]);

        unique_ag_ID = unique_ag_ID + 1;

        map(y(x),x)=ag(end).ID;

        placement(y(x),x)=0;
    end
end

% place cops
for i=1:numC
    if isempty(find(placement, 1))
        break
    end
    [unn,y]=max(placement);

```

```
[~,x]=max(unn);

ag(end+1).position = [y(x),x];
ag(end).active = 0;
ag(end).hardship = 0;
ag(end).grievance = 0;
ag(end).criminal = 0;
ag(end).age = 0;
ag(end).risk_aversion = 0;
ag(end).jail_term = 0;
ag(end).jail_time = 0;
ag(end).ID = unique_ag_ID;
ag(end).type = -1;
ag(end).max_age = inf;

unique_ag_ID = unique_ag_ID + 1;

map(y(x),x)=ag(end).ID;

placement(y(x),x)=0;
end
end
```

```
function [y,x,ag,neighbors] = get_view(map, pos, vis, gridsize)
%GET_VIEW Get local view/sight of an given agent at pos.
% GET_VIEW(map, pos, vis, gridsize) determines the local environment↵
matrix
% of the agent at pos. This matrix is filtered with the individual view
% pattern. A list of all agents in sight and their positions is↵
returned.
%
% Date:
% $Id: get_view.m 38 2010-11-26 23:05:54Z usoler $
% Author:
% Grimm Alexander, Koller Michael, Solèr Ursin

% agent sight
lb = pos-vis.max;
ub = pos+vis.max;
uy = mod( (lb(1):ub(1)) - 1, gridsize(1) ) + 1; % get_ind inlined
ux = mod( (lb(2):ub(2)) - 1, gridsize(1) ) + 1; %
neighbors = map(uy, ux);
view = neighbors .* vis.pattern;

% list agents in sight
[y,x,ag] = find(view);
ag = ag';

end
```

```
function [cops,actives] = count_agent(view, ag)
%COUNT_AGENT Count all agents within the given view.
% COUNT_AGENT(view, ag) Extracts all agents and cops from view, counts
them
% and returns that result.
%
% Date:
% $Id: count_agent.m 38 2010-11-26 23:05:54Z usoler $
% Author:
% Grimm Alexander, Koller Michael, Solèr Ursin

% count other within agent sight
typ = [ag(view).type];
cops = sum( (typ == -1) );
actives = sum( (typ >= 1) & ([ag(view).active] == 1) );

end
```

```
function [] = show_final_plots(meas_map,meas_ag,meas_prison)
%SHOW_FINAL_PLOTS Shows plots in very end.
% SHOW_FINAL_PLOTS(meas_map,meas_ag,meas_prison) take all simulation
% status relevant data and display the measurement results of them all
% at once.
%
% Date:
% $Id: show_final_plots.m 44 2010-12-04 23:11:14Z usoler $
% Author:
% Grimm Alexander, Koller Michael, Solèr Ursin
% Mundy Sam, Oberhauser Tim (some code snippets)

global ag map gridsize vis

f=figure('Position',[0 0 900 800]);
s = size(meas_map);
for step = 1:s(1);
    % restore data
    map(:, :) = meas_map(step, :, :);
    ag(:) = meas_ag(step, :);

    % PROCESS measured data and prepare for display
    map_ag_all = (map~=0);
    map_ag_active = zeros(gridsize);
    map_ag_griev = zeros(gridsize);
    map_ag_type = zeros(gridsize);
    map_ag_ca_r = zeros(gridsize);
    for a = ag
        if a.type~=0
            pos = a.position;
            map_ag_active(pos(1),pos(2)) = a.active;
            map_ag_griev(pos(1),pos(2)) = a.grievance;
            map_ag_type(pos(1),pos(2)) = a.type;

            % count other within agent sight (slow)
            [~,~,view,~] = get_view(map, pos, vis, gridsize);
            [cops,actives] = count_agent(view, ag);
            map_ag_ca_r(pos(1),pos(2)) = (cops/(actives+1));
        end
    end
    map_cop = (map_ag_type==1);

    prison = meas_prison{1,step,:};

    % OUTPUT and display
    spl=subplot(2,2,1,'Color',[0.8 0.8 0.8]);
    cla reset;
    [row,col] = find(map_ag_type==1);
    pop_size.A = length(row);
    scatter( row-.5, col-.5, ...
            'MarkerFaceColor', [0 0 1], ...
            'MarkerEdgeColor', [0 0 1], ...
```

```

        'Parent', sp1 );
%   scatter( row-.5, col-.5, 14, ...
%           'MarkerFaceColor', [0 0 1], ...
%           'MarkerEdgeColor', [0 0 1], ...
%           'Parent', sp1 );
hold on;
[row,col] = find(map_ag_type==2);
pop_size.B = length(row);
scatter( row-.5, col-.5, ...
        'MarkerFaceColor', [0 1 0], ...
        'MarkerEdgeColor', [0 1 0], ...
        'Parent', sp1 );
%   scatter( row-.5, col-.5, 14, ...
%           'MarkerFaceColor', [0 1 0], ...
%           'MarkerEdgeColor', [0 1 0], ...
%           'Parent', sp1 );
[row,col] = find(map_cop);
scatter( row-.5, col-.5, ...
        'MarkerFaceColor', [0 0 0], ...
        'MarkerEdgeColor', [0 0 0], ...
        'Parent', sp1 );
%   scatter( row-.5, col-.5, 14, ...
%           'MarkerFaceColor', [0 0 0], ...
%           'MarkerEdgeColor', [0 0 0], ...
%           'Parent', sp1 );
[row,col] = find(map_ag_active);
%scatter( row-.5, col-.5, ...
%        'MarkerFaceColor', [1 0 0], ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
scatter( row-.5, col-.5, '.', ...
        'MarkerFaceColor', [1 0 0], ...
        'MarkerEdgeColor', [1 0 0], ...
        'Parent', sp1 );
%scatter( row-.5, col-.5, 'x', ...
%        'MarkerFaceColor', [1 0 0], ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
%scatter( row-.5, col-.5, 'o', ...
%        'MarkerFaceColor', 'none', ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
xlim([0 40]);
ylim([0 40]);
%F(step) = getframe();

subplot(2,2,2);
%colormap('Pink')
colormap('Copper')
cop_scale = 2*max(max(map_ag_griev));
imagesc(rot90(-(map_ag_griev+cop_scale*map_cop)));

subplot(2,2,4);
imagesc(rot90(map_ag_ca_r));

```

```
drawnow;
%print(f, '-dpng', ['img/', 'output_', sprintf('%04i', step), '.png']);
%%saveas(f, 'output.png');

%mov(step) = getframe(f);

% MEASURE data
meas(step,:) = [ step length(find(map_ag_active)) length(prison) ...
                pop_size.A pop_size.B ];

pause(.2);
end

disp('step    active    prison    pop A    pop B');
meas
figure();
plot(meas(:,1),meas(:,2),'r')
hold on;
plot(meas(:,1),meas(:,3),'k')
plot(meas(:,1),meas(:,4),'b')
plot(meas(:,1),meas(:,5),'g')

% Create AVI file.
%movie(mov,1)
%movie2avi(mov, 'output.avi', 'compression', 'None', 'fps', 3);
%!mencoder mf://img/*.png -mf fps=3:type=png -ovc lavc -lavcopts
vcodec=mpeg4:mbd=2:trell -oac copy -o img/output.avi

end
```



```
function [map_ag_active,map_ag_all] = show_plots()
%SHOW_PLOTS Shows plots every iteration.
% SHOW_PLOTS() take all simulation status relevant data and display
% the measurement results of them.
%
% Date:
% $Id: show_plots.m 36 2010-11-23 20:36:39Z usoler $
% Author:
% Grimm Alexander, Koller Michael, Solèr Ursin
% Mundy Sam, Oberhauser Tim (some code snippets)

global ag map gridsize

% PROCESS measured data and prepare for display
map_ag_all = (map~=0);
map_ag_active = zeros(gridsize);
map_ag_griev = zeros(gridsize);
map_ag_type = zeros(gridsize);
for a = ag
    if a.type~=0
        pos = a.position;
        map_ag_active(pos(1),pos(2)) = a.active;
        map_ag_griev(pos(1),pos(2)) = a.grievance;
        map_ag_type(pos(1),pos(2)) = a.type;
    end
end
map_cop = (map_ag_type==1);

% OUTPUT and display
sp1=subplot(1,2,1,'Color',[0.8 0.8 0.8]);
cla reset;
[row,col] = find(map_ag_type==1);
scatter( row-.5, col-.5, ...
        'MarkerFaceColor', [0 0 1], ...
        'MarkerEdgeColor', [0 0 1], ...
        'Parent', sp1 );

hold on;
[row,col] = find(map_ag_type==2);
scatter( row-.5, col-.5, ...
        'MarkerFaceColor', [0 1 0], ...
        'MarkerEdgeColor', [0 1 0], ...
        'Parent', sp1 );

[row,col] = find(map_cop);
scatter( row-.5, col-.5, ...
        'MarkerFaceColor', [0 0 0], ...
        'MarkerEdgeColor', [0 0 0], ...
        'Parent', sp1 );

[row,col] = find(map_ag_active);
%scatter( row-.5, col-.5, ...
%        'MarkerFaceColor', [1 0 0], ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
scatter( row-.5, col-.5, '.', ...
        'MarkerFaceColor', [1 0 0], ...
```

```
        'MarkerEdgeColor', [1 0 0], ...
        'Parent', sp1 );
%scatter( row-.5, col-.5, 'x', ...
%        'MarkerFaceColor', [1 0 0], ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
%scatter( row-.5, col-.5, 'o', ...
%        'MarkerFaceColor', 'none', ...
%        'MarkerEdgeColor', [1 0 0], ...
%        'Parent', sp1 );
xlim([0 40]);
ylim([0 40]);
subplot(1,2,2);
%colormap('Pink')
colormap('Copper')
cop_scale = 2*max(max(map_ag_griev));
imagesc(rot90(-(map_ag_griev+cop_scale*map_cop)));
%pause(100);
drawnow;

end
```

```
% EPSTEIN MODEL II
%
% need to verify: [ok] Peaceful Coexistence / p.6
% (reproduce)      [ok] Ethnic Cleansing / p.7
%                [ok] Safe Havens / p.7
%                - Cop Density and Extinction Times / p.7
%                - ( Summary Of Model II Results / p.8 )
%
% DONE:
%   Look at http://www.math.ucla.edu/~getreuer/matopt.pdf to get further
%   info on how to speed up.
%   * M-Lint Messages: all with exception of those related to Array
%                       Preallocation are solved
%   * The Profiler: use with 'profile on' and 'profile report' some
%                       optimizations were done
%   * Vectorization and Inlining Simple Functions
%
%
%   Date:
%       $Id: main_ep_2.m 48 2010-12-06 20:08:57Z usoler $
%   Author:
%       Grimm Alexander, Koller Michael, Solèr Ursin
%       Mundy Sam, Oberhauser Tim (some code snippets)

clear;
clc;

% init needed vars
global gridsize vis ag map max_age unique_ag_ID legitimacy

% RUN 6
%init_ep_2_run_06;
%config = {'init_ep_2_run_06'};

% RUN 7; ethnic cleansing / safe havens (activate cop adding)
%init_ep_2_run_07;
config = {'init_ep_2_run_07'};

stat_count = 1; % (not used yet)

for param = config
    % run init script and init vars
    param
    j = batch(char(param));
    wait(j);
    load(j);

    %step_max_count = 10;

    % setup vision from radius
    vis.max = floor(vis.real);
```

```

vis.origin = vis.max+1;
vis.size = (2*vis.origin-1) * ones(1,2);
vis.pattern = zeros(vis.size);
for i = 1:vis.size(1)
    for j = 1:vis.size(2)
        vis.pattern(i,j) = ...
            (sqrt( (i-vis.origin)^2 + (j-vis.origin)^2 )<=vis.real);
    end
end
vis.pattern(vis.origin,vis.origin) = 0;
%clear vis.origin;
%imagesc(vis.pattern)

unique_ag_ID = 1;

% statistics loop
%figure('Position',[0 0 900 800])
clear meas_ag meas_prison;
meas_map = zeros([stat_count step_max_count gridsize]);
for stat = 1:stat_count
    % create/place agents
    ag = []; % agents (A, cop)
    map = zeros(gridsize); % map/grid
    prison = []; % prison

    add_agent(numA,numC);

    % simulation loop
    for step = 1:step_max_count
        step

%           % add cops, comment out if not RUN 7
%           if step == 50
%               add_agent([],0.04*(gridsize(1)*gridsize(2)));
%           end

        % activation and ageing in the free world
        for ag_i = randperm(length(ag)) % rand. but each just once
            % skip prisoners and deaths
            if (ag(ag_i).type==0)
                continue
            end

            if length(find(map==ag_i))>1
                ag_i
                disp('problem!');
            end

            % agent pos
            pos = ag(ag_i).position;

            % movement to random site in vision
            % find free position within agent sight

```

```

[~,~,~,neighbors] = get_view(map, pos, vis, gridsize);
free=(neighbors == 0);
newsite = free .* vis.pattern .* rand(vis.size);
m=max(max(newsite));
if (m~=0)
    [y,x]= find( newsite==m );
    newpos = get_ind( pos + [y x] - vis.origin );
    ag(ag_i).position = newpos;
    map(pos(1),pos(2)) = 0;
    map(newpos(1),newpos(2)) = ag(ag_i).ID; % =ag_i
end

% agent new pos
pos = ag(ag_i).position;

% agent sight
[~,~,~,neighbors] = get_view(map, pos, vis, gridsize);

if ag(ag_i).type >= 1
    % age
    ag(ag_i).age = ag(ag_i).age + 1;
    if ag(ag_i).age >= ag(ag_i).max_age
        % agent dies
        ag(ag_i).type = 0;
        map( map==ag_i ) = 0;
        continue;
    end

    % birth/cloning to random site in vision
    if rand() < cloning_probability
        % find free position within agent sight
        free=(neighbors == 0);
        newsite = free .* vis.pattern .* rand(vis.size);
        m=max(max(newsite));
        if (m~=0)
            [y,x]= find( newsite==m );
            newpos = get_ind( pos + [y x] - vis.origin );

            ag(end+1) = ag(ag_i); % hards., griev., type
            ag(end).position      = newpos;
            ag(end).active        = 0;
            ag(end).criminal      = 0;
            ag(end).age           = 0;
            ag(end).risk_aversion = rand(); % fixed
            ag(end).jail_term     = 1;      % neutral
            ag(end).jail_time     = 0;
            ag(end).ID            = unique_ag_ID;
            ag(end).max_age       = randi([0,max_age]);

            unique_ag_ID = unique_ag_ID + 1;

            map(newpos(1),newpos(2)) = ag(end).ID;
        end
    end
end

```

```

% count all in sight
[~,~,view,~] = get_view(map, pos, vis, gridsize);
[cops,actives] = count_agent(view, ag);

% activate agents
%P = 1 - exp(-k*((cops+1)/(actives+1)));
P = 1 - exp(-k*(cops/(actives+1)));
%N = ag(ag_i).risk_aversion * P * ag(ag_i).jail_term;
N = ag(ag_i).risk_aversion * P;
if (ag(ag_i).grievance - N) > threshold
    %ag(ag_i).active = 1;

    % now kill an foreign agent
    type=[ag(view).type];
    % ( '.*' here is like '&&' vectorized )
    vis.ind=find((type~=ag(ag_i).type).*(type~-1));
    if vis.ind
        ag(ag_i).active = 1;

        i=view( vis.ind( randi(length(vis.ind)) ) );

        ag(i).type = 0;
        map( map==i ) = 0;
    else
        ag(ag_i).active = 0;
    end
else
    ag(ag_i).active = 0;
end
elseif ag(ag_i).type == -1
    % arrest random active in sight
    active=[ag(view).active];
    vis.ind=find(active);
    if vis.ind
        i=view( vis.ind( randi(length(vis.ind)) ) );
        %ID=[ag(:).ID];
        %i = find( ID==ag(ag_i).ID );

        % put to jail
        %prison(end+1) = ag(i);
        prison(end+1).position = ag(i).position;
        prison(end).active = ag(i).active;
        prison(end).hardship = ag(i).hardship;
        prison(end).grievance = ag(i).grievance;
        prison(end).criminal = ag(i).criminal;
        prison(end).age = ag(i).age;
        prison(end).risk_aversion = ag(i).risk_aversion;
        prison(end).jail_term = randi([0, jail_term_max]);

        prison(end).jail_time = 0;
        prison(end).ID = ag(i).ID;
        prison(end).type = ag(i).type;
        prison(end).max_age = ag(i).max_age;
        ag(i).type = 0;
        map( map==i ) = 0;
    end
end

```


A.2 Configuration


```
% EPSTEIN MODEL INITIALIZATION
% RUN 1; Individual Deceptive Behavior
%
%
%   Date:
%       $Id: init_ep_1_run_01.m 44 2010-12-04 23:11:14Z usoler $
%   Author:
%       Grimm Alexander, Koller Michael, Solèr Ursin

%% init needed vars
gridsize = [40, 40];
N_tot = gridsize(1)*gridsize(2);

numA = round(N_tot*0.7); % agent A
numC = round(N_tot*0.04); % agent cop

legitimacy = 0.89; % every value below leads to activity

vis.real = 1.7;

k = log(10);

threshold = 0.1;

% and is the same as jail_time_max according to epstein paper p.4
jail_term_max = 15;

step_max_count = 100;
```

```
% EPSTEIN MODEL I
% RUN 2; Free Assembly Catalyzes Rebellious Outbursts
% (from $Id: init_ep_1_parascan_65.m 44 2010-12-04 23:11:14Z usoler $)
%
%
%   Date:
%   $Id: init_ep_1_parascan_65.m 44 2010-12-04 23:11:14Z usoler $
%   Author:
%       Grimm Alexander, Koller Michael, Solèr Ursin

%% init needed vars
gridsize = [40, 40];
N_tot = gridsize(1)*gridsize(2);

numA = round(N_tot*0.7); % agent A
numC = round(N_tot*0.04); % agent cop

legitimacy = 0.82; % every value below leads to activity (round)
%legitimacy = 0.86; % for floor, thus round makes more sence!

vis.real = 7;

k = log(10);

threshold = 0.1;

% and is the same as jail_time_max according to epstein paper p.4
jail_term_max = 30;

step_max_count = 100;
```

```
% EPSTEIN MODEL INITIALIZATION
% RUN 3(+4); Salami Tactics of Corruption
%
%
%   Date:
%       $Id: init_ep_1_run_03.m 44 2010-12-04 23:11:14Z usoler $
%   Author:
%       Grimm Alexander, Koller Michael, Solèr Ursin

%% init needed vars
gridsize = [40, 40];
N_tot = gridsize(1)*gridsize(2);

numA = round(N_tot*0.7); % agent A
numC = round(N_tot*0.074); % agent cop

legitimacy = 0.9; % every value below leads to activity

vis.real = 7.;

k = log(10);

threshold = 0.1;

% and is the same as jail_time_max according to epstein paper p.4
%jail_term_max = inf;
jail_term_max = 2^52;

%step_max_count = 60; % for jump
step_max_count = 180; % for small continous reductions
```

```
% EPSTEIN MODEL II
% RUN 6; peaceful coexistence
%
%
%   Date:
%   $Id: init_ep_2_run_06.m 32 2010-11-19 21:19:00Z usoler $
%   Author:
%   Grimm Alexander, Koller Michael, Solèr Ursin

%% init needed vars
gridsize = [40, 40];
%gridsize = [10, 10];
N_tot = gridsize(1)*gridsize(2);

numA = [round(N_tot*0.35) round(N_tot*0.35)]; % agents [A,B, ...]
numC = round(N_tot*0.0); % agent cop

legitimacy = 0.9; % every value below leads to activity

vis.real = 1.7;

k = log(10);

threshold = 0.1;

% and is the same as jail_time_max according to epstein paper p.4
jail_term_max = 15;

max_age = 200;
cloning_probability = 0.05;

step_max_count = 20;
```

```
% EPSTEIN MODEL II
% RUN 7; ethnic cleansing / safe havens (activate cop adding)
%
%
%   Date:
%   $Id: init_ep_2_run_07.m 32 2010-11-19 21:19:00Z usoler $
%   Author:
%   Grimm Alexander, Koller Michael, Solèr Ursin

%% init needed vars
gridsize = [40, 40];
%gridsize = [10, 10];
N_tot = gridsize(1)*gridsize(2);

numA = [round(N_tot*0.35) round(N_tot*0.35)]; % agents [A,B, ...]
numC = round(N_tot*0.0); % agent cop

legitimacy = 0.8; % ...

vis.real = 1.7;

k = log(10);

threshold = 0.1;

% and is the same as jail_time_max according to epstein paper p.4
jail_term_max = 15;

max_age = 200;
%cloning_probability = 0.05;
cloning_probability = 0.30;

step_max_count = 200;
```

B NetLogo Code

Available from https://svn.vis.ethz.ch/svn/usoler_civil-viol10/
by svn.

```

breed [agents agent]
breed [cops cop]

globals [
  k                ; factor for determining arrest probability
  threshold        ; by how much must G > N to make someone rebel?
]

agents-own [
  risk-aversion    ; R, fixed for the agent's lifetime, ranging from 0-1
  (inclusive)
  perceived-hardship ; H, also ranging from 0-1 (inclusive)
  active?          ; if true, then the agent is actively rebelling
  jail-term        ; how many turns in jail remain? (if 0, the agent is not in
jail)
]

patches-own [
  neighborhood    ; surrounding patches within the vision radius
]

to setup
  clear-all

  ;; set globals
  set k 2.3
  set threshold 0.1

  ask patches [
    ;; make background a slightly dark gray
    set pcolor gray - 1
    ;; cache patch neighborhoods
    set neighborhood patches in-radius vision
  ]

  ;; create cops
  create-cops round (initial-cop-density * .01 * count patches) [
    move-to one-of patches with [not any? turtles-here]
    display-cop
  ]

  ;; create agents
  create-agents round (initial-agent-density * .01 * count patches) [
    move-to one-of patches with [not any? turtles-here]
    set heading 0
    set risk-aversion random-float 1.0
    set perceived-hardship random-float 1.0
    set active? false
    set jail-term 0
    display-agent
  ]

  ;; plot initial state of system
  update-plots
end

to go
  ask turtles [
    ; Rule M: Move to a random site within your vision
    if (breed = agents and jail-term = 0) or breed = cops
      [ move ]
    ; Rule A: Determine if each agent should be active or quiet
    if breed = agents and jail-term = 0 [ determine-behavior ]
    ; Rule C: Cops arrest a random active agent within their radius
  ]

```

```

    if breed = cops [ enforce ]
  ]
  ; Jailed agents get their term reduced at the end of each clock tick
  ask agents
    [ if jail-term > 0 [ set jail-term jail-term - 1 ] ]
  ; update agent display
  ask agents [ display-agent ]
  ask cops [ display-cop ]
  ; advance clock and update plots
  tick
  update-plots
end

;; AGENT AND COP BEHAVIOR

;; move to an empty patch
to move ;; turtle procedure
  if movement? or (breed = cops) [
    ;; move to a patch in vision; candidate patches are
    ;; empty or contain only jailed agents
    let targets neighborhood with
      [not any? cops-here and all? agents-here [jail-term > 0]]
    if any? targets [ move-to one-of targets ]
  ]
end

;; AGENT BEHAVIOR

to determine-behavior
  set active? (grievance - risk-aversion * estimated-arrest-probability >
threshold)
end

to-report grievance
  report perceived-hardship * (1 - government-legitimacy)
end

to-report estimated-arrest-probability
  let C count cops-on neighborhood
  let A 1 + count (agents-on neighborhood) with [active?]
  ;; See Information tab for a discussion of the following formula
  report 1 - exp (- k * floor (C / A))
end

;; COP BEHAVIOR

to enforce
  if any? (agents-on neighborhood) with [active?] [
    ;; arrest suspect
    let suspect one-of (agents-on neighborhood) with [active?]
    ask suspect [
      set active? false
      set jail-term random max-jail-term
    ]
    move-to suspect ;; move to patch of the jailed agent
  ]
end

;; VISUALIZATION OF AGENTS AND COPS

to display-agent ;; agent procedure
  ifelse visualization = "2D"
    [ display-agent-2D ]
    [ display-agent-3D ]

```



```

end

to display-agent-2D ;; agent procedure
  set shape "circle"
  ifelse active?
    [ set color red ]
    [ ifelse jail-term > 0
      [ set color black + 3 ]
      [ set color scale-color green grievance 1.5 -0.5 ] ]
end

to display-agent-3D ;; agent procedure
  set color scale-color green grievance 1.5 -0.5
  ifelse active?
    [ set shape "person active" ]
    [ ifelse jail-term > 0
      [ set shape "person jailed" ]
      [ set shape "person quiet" ] ]
end

to display-cop
  set color cyan
  ifelse visualization = "2D"
    [ set shape "triangle" ]
    [ set shape "person soldier" ]
end

;; PLOTTING

to update-plots
  let active-count count agents with [active?]
  let jailed-count count agents with [jail-term > 0]

  set-current-plot "Active agents"
  plot active-count

  set-current-plot "All agent types"
  set-current-plot-pen "active"
  plot active-count
  set-current-plot-pen "jailed"
  plot jailed-count
  set-current-plot-pen "quiet"
  plot count agents - active-count - jailed-count
end

; Copyright 2004 Uri Wilensky. All rights reserved.
; The full copyright notice is in the Information tab.

```

```
.  
. .  
to-report estimated-arrest-probability  
  let C count cops-on neighborhood  
  let A 1 + count (agents-on neighborhood) with [active?]  
  ;; See Information tab for a discussion of the following formula  
  report 1 - exp (- k * round (C / A))  
end  
  
. .  
. .
```

```

.
.
.
to go
  ask turtles [
    ; Rule M: Move to a random site within your vision
    if (breed = agents and jail-term = 0) or breed = cops
      [ move ]
    ; Rule A: Determine if each agent should be active or quiet
    if breed = agents and jail-term = 0 [ determine-behavior ]
;   ; Rule C: Cops arrest a random active agent within their radius
;   if breed = cops [ enforce ]
  ]
; Jailed agents get their term reduced at the end of each clock tick
ask agents
  [ if jail-term > 0 [ set jail-term jail-term - 1 ] ]
; update agent display
ask agents [ display-agent ]
ask cops [ display-cop ]
; advance clock and update plots
tick
update-plots
end
.
.
.

```