

Effect Of Obscuring Vision On An Agent Based Model Of A Riot

Angus Blance

March 10, 2023

Contents

1	Abstract	2
2	Introduction	2
3	My Model	2
3.1	What is an Agent Based Model	2
3.2	Mathematical Model	3
4	Computational Approach and Implementation	8
4.1	Moving	10
4.2	Quiets and Actives The Descension to Riot	11
4.3	Police Arresting	12
5	My Results	13
6	Evaluation	14
7	Bibliography	16
8	Appendices (Matlab Code)	17

1 Abstract

2 Introduction

Instances of large-scale disorder occur frequently around the world, resulting in significant damage to property and, at times, human life. Given the destructive nature of such events, it is crucial to develop models that can accurately capture the dynamics of rioting and identify potential strategies for mitigating the damage. In this report, we present an agent-based model (ABM) that replicates the behaviour of riot populations, and we demonstrate how the use of smoke by police forces can effectively reduce the number of participants in a riot.

What Causes A Riot

riots can be caused by many things such as deprivation, poor police relations, legitimacy of a government or just too many people in one area. Once a riot has started it then gets out of control very fast usually epidemic like(ref 2.). Factor's for this can be knowledge of an ongoing riot through social Media or friends and Rational choice theory as in if there are shops being looted and police are not doing anything you are more likely to join. riots can then be continued if there is inadequate policing

3 My Model

3.1 What is an Agent Based Model

An agent-based model (ABM) is a computational modelling technique used to simulate the behaviour of individual agents and their interactions within a system. In an ABM, each agent is typically characterized by a set of rules, behaviours, and attributes that influence its decision-making process and its interactions with other agents in the environment. These agents can be anything from people, animals, or even artificial entities.

ABMs are commonly used to simulate complex systems, such as social systems, ecological systems, or financial markets, where the behaviour of individual agents and their interactions can have a significant impact on the overall behaviour of the system. ABMs can also be used to explore the effects

of different policies or interventions on the system, by adjusting the rules governing the behaviour of the agents and observing the resulting changes in the system's behaviour.

For this study, I employ an ABM to simulate a riot using Epstein's Model for civil violence, which consists of three agents: quiet agents (Q) who are not participating in the riot, active agents (A) who are rioting, and cop agents (C) who represent a potential intervention strategy. Each agent is placed on an N by N board

attributes for these agents are given:

quiet:

- walk around the board at random
- calculates the amount of police and actives within its vicinity
- Perform a series of calculations, taking into account the number of police and active agents, as well as their own risk level, hardship, and perception of the legitimacy of the government. Based on these factors, they decide whether or not to become an active agent.
- if the ratio of active to police is deemed suitable they become active

Active:

- walk around randomly

police:

- walk around randomly
- when near an active they arrest them and take them off the board

3.2 Mathematical Model

In our N by N board, each agent is assigned a unique identification key to distinguish their agent type, which is represented by the numbers 1, 2, and 3 for quiet, police, and active agents, respectively. Additionally, each agent is given a level of hardship (H) and a level of risk, both drawn from a uniform distribution on the interval (0, 1). We store this information in a data structure, as shown in Figure 1.

Once we have assigned each agent a unique identification key and their corresponding levels of hardship and risk, we place them in their designated

crowd =

1.0000	0.7064	0.6949	0
1.0000	0.5819	0.5669	0
1.0000	0.8484	0.2954	0
1.0000	0.8747	0.2338	0
1.0000	0.6758	0.5831	0
1.0000	0.7370	0.4459	0
1.0000	0.1788	0.6315	0

police =

2	0	0	0
2	0	0	0
2	0	0	0

Figure 1: crowd police format

```

{[          2 0 0 0]}  {[          0 0 0 0]}  {[          0 0 0 0]}
{[1 0.6412 0.7994 0]}  {[1 0.4626 0.3955 0]}  {[          0 0 0 0]}
{[          0 0 0 0]}  {[1 0.9792 0.0672 0]}  {[1 0.1153 0.2751 0]}

```

Figure 2: grid beginning

location on the N by N board. This results in a configuration similar to Figure 2, where each cell in the grid represents an agent with its identification key, hardship level (H), and risk level.

Note that the police do not have a hardship or risk level, this is as they do not need it as they do not perform the calculation to become an active. now i made it so each person (rioter, police person or civilian) would move a random direction each iteration given that there was nobody already in that space.

How Civilians Chose To Riot

civilians also have the ability to become a rioter for this i used the aspects from the Epstein model (ref 1.) where we first find the Grievance.

$$G = H(L - 1) \quad (1)$$

where H is the hardship of each specific civilian and L is the legitimacy of the government that person is in which we set as low (0.1 ish) so that we do create a riot.

now we find the probability (p) to be arrested which each citizen needs to calculate before they decide whether or not they wish to riot:

$$P = 1 - EXP(-k(C/A)_v) \quad (2)$$

note the subscript v notates the vision which each civilian has as shown in fig 3.

C is the number of police in the vision v and A is the number of rioters in vision v. The civilian counts himself when counting A as he wants to see how the riot would be once he is added. K is just a constant which we set such that $P = 0,9$ when $C=A=1$, so $k = -\ln(0.1)$

from this we can now define the net risk $N = RPJ$ where J is the jail time one faces once he has been caught by a police officer. For this model it is just set to about 5

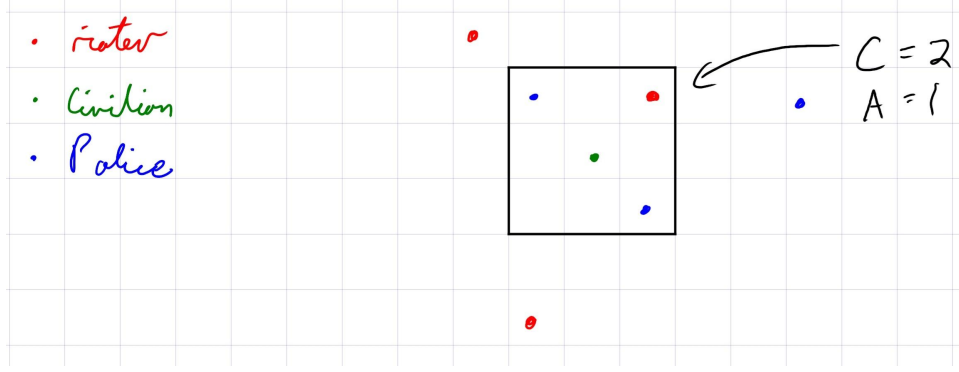


Figure 3: vision example

we now have all we need to create the agent rule:

$$\text{Agent rule : if } G - N > T \text{ riot; otherwise stay a civilian} \quad (3)$$

so the difference in G and N is the expected utility of expressing ones private grievance whilst T is the utility of not.

Police Attributes

The police in the model are pretty simple, they just move around and scan the squares around them every turn, if there is a rioter within those squares he takes them of the board turning it into $[0,0,0]$, an empty space (ie arrested).

along with this every turn each police officer has an opportunity to throw smoke if a rioter is within its specified range for throwing smoke.

Smoke

As smoke does not move like the agents in this model I created a second grid the same size as the grid that our agents are on.

each square has 2 values for it firstly a 0 or 1 representing whether there is no smoke (0) or smoke (1) and another value from 0 to how many iterations the smoke lasts for.

The smoke will be a square $n \times n$ area.

What the smoke actually affects is the probability for a civilian to start rioting as it reduces the number of rioters that the civilian can see within its range v.

I have also made it so they still count police officers even if they are inside of the smoke, this is done as i assume police officers don't generally throw smoke at each-other.

Small Detail

In the Epstein model the rioters and the civilians can move to anywhere within there vision whilst the police officers can only move within the small distance which they can arrest rioters within. I made it so everyone can only move 1 square each tun as i do not believe there is that much of a difference in speed between the general public and police.

4 Computational Approach and Implementation

The code is quite complex and hard to describe through words alone so in this section I wish to use diagrams in order to visually explain my code step by step from the very beginning.

First of all we have our $N \times N$ board say $N=10$ shown in figure 4

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Figure 4: French riot

Now we introduce our agents with attributes as described in our "What is an Agent Based Model" section, Let our Actives be denoted by the colour

red, our quiet to be the colour green and our Cops to be the colour Blue. Say we loop through our hole $N \times N$ grid and place 20 quiet and 10 cops at random we will end up with something like Figure 5

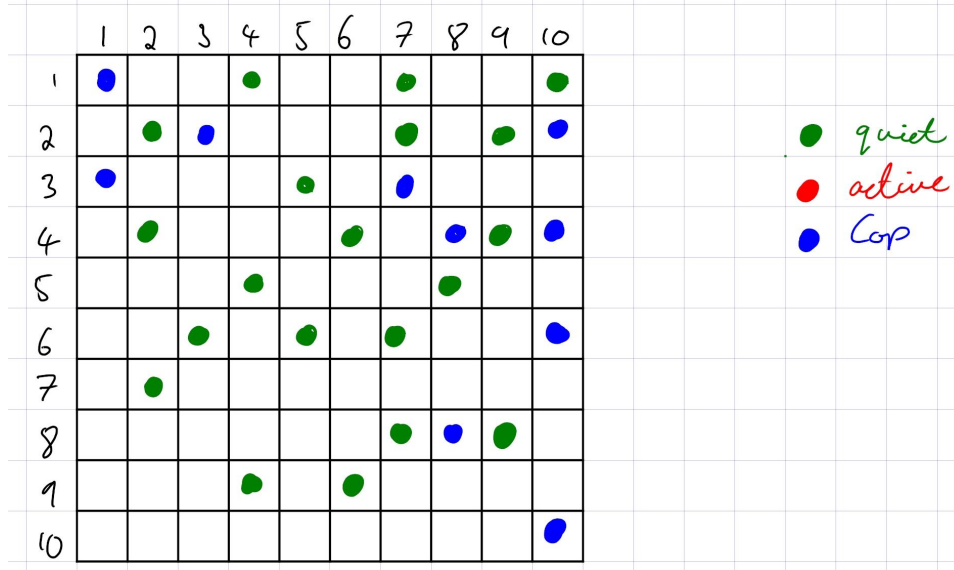


Figure 5: French riot

Great! That's our hole grid set up. But also note that each individual quiet has 2 attributes to them (risk level and Hardship) which is easier to see on a 10×10 example run on our code in Matlab as shown in figure 6

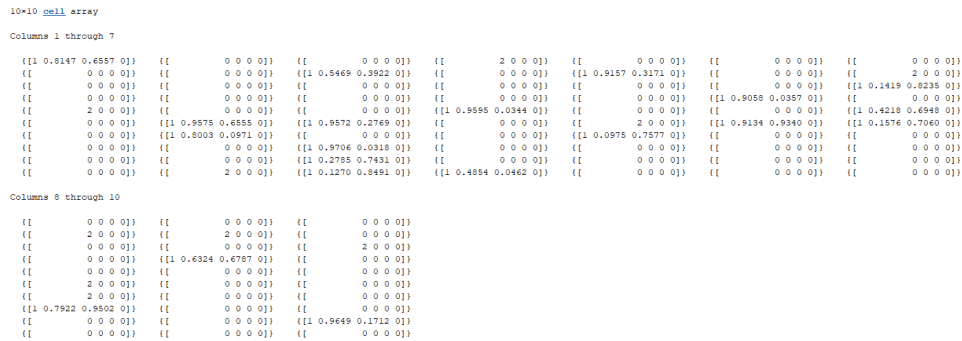


Figure 6: French riot

Notice how we store the risk level and Hardship level by making our board a cell array storing multiple values for each cell. Also see how all of the cops have no hardships or risk, This is because they do not have the attribute to become an active meaning giving them such values will be redundant.

4.1 Moving

Now the next thing to do is make each person move, I decided everybody will only move one space (Cell) at a time how they move is shown in figure 7 where the arrows on the diagram on the left show the direction in which each agent will move and the diagram on the right shows the grid after each person has moved

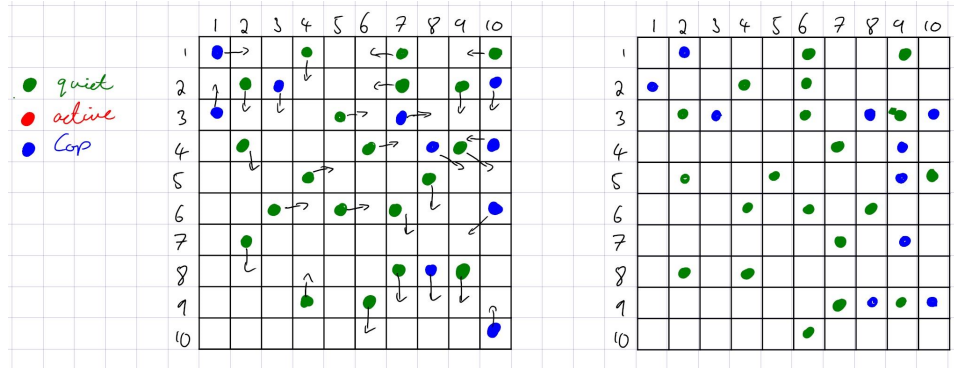


Figure 7: French riot

see how the Cop in column 10 row 4 (10,4) moves to space (9,4) which is already occupied by a quiet? This is allowed to happen due to how I am looping through the board, I do this by going through (column,row) where row = 1,2,...,10 and column = 1,2,...,10 from the top left of the board to the bottom right meaning that the quiet in space (9,4) has already moved to (10,5) before the cop makes his move into the space the quiet was just in.

4.2 Quiets and Actives The Descension to Riot

Just as before we first need to loop through the whole board, first starting from the top left and ending up on the bottom right. whenever we land on a quiet we take its risk and Hardship as well as searching in a $M \times M$ grid around said quiet to collect the number of actives and Cops within its vision, its vision being the $M \times M$ area. One example of this is shown in Figure (fig num)

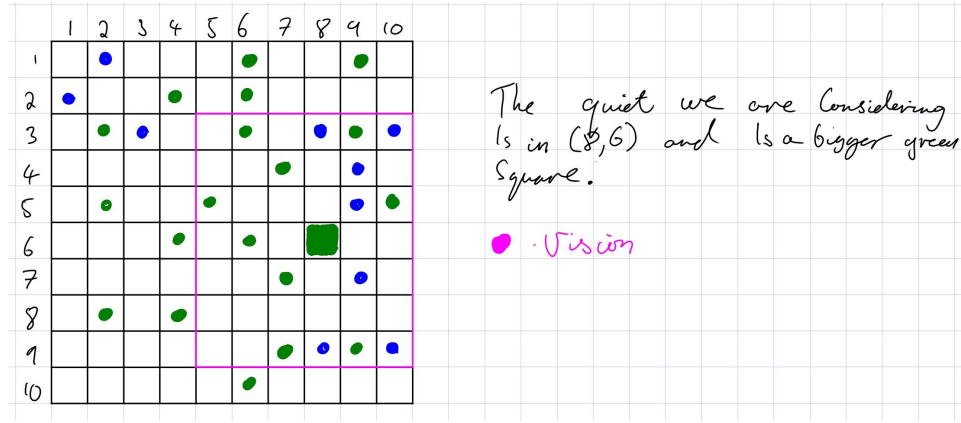


Figure 8: French riot

Before we would have even gotten to this particular active we would have consider all other active in (column, row) for column = 1,2,...,10 and row = 1,2,...,6. In this particular case we assume that non of these quites transitioned into an active.

Now looking at what data we have for our particular case in figure (fig num) we have the number of police = 7 and the number of quiets = 9 not including the quiet we are currently considering. This is as when a quiet is considering to become and active it is thinking, if I am an active what is the probability of me getting arrested so it considers itself an active meaning we have actives = 1 in the area.

For This particular quiet to become active we need equation 3 to be satisfied. Lets explore this more mathematically looking at each variable (3) individually. First lets look at N

$$N = RPJ \quad (4)$$

Where R is the level of risk aversion they will take given to each individual quiet, higher level of risk aversion the less likely they are to take risks. J is the jail time and P is the probability to get arrested given by

$$P = 1 - EXP(-k(C/A)_v) \quad (5)$$

We have C and A from figure (fig num) and k is a constant set such that $p = 0.9$ when $C=1$ and $A = 1$ so $k = \ln(0.1)$ subbing these values in we get $P = 0.9999999$ for our specific case, which makes sence as there are no actives within the vicinity of our quiet and there are many police meaning the probability of arrest should be high. Also note as $C \rightarrow \infty \Rightarrow P \rightarrow 1$ and as $A \rightarrow \infty \Rightarrow P \rightarrow 0$.

say in our particular case $J = 5$ which is a smallish jail sentence which should mean a rioter is more likely to riot then our $N = R \times 4.9999995$ and using this and substiduting G from (1) we obtain

$$G - N = H(L - 1) - R \times 4.9999995 \quad (6)$$

Say we want our quiet to become active we want (6) to be Large such that it is $\geq T$ meaning we want $H(L-1)$ the total grievance of this person to be large and $R \times 4.9999995$ to be small. First for $H(L-1)$ to be large we need want the hardship of our specific individual to be large and the legitimacy of our government to be low, Note that just having a high level of hardship does not mean our quiet will become active, we need both a high hardship AS WELL as a low perceived legitimacy of the government. next for $R \times 4.9999995$ to be low we need our individuals level of risk aversion to be low. Notice that for our agent to become active that shorter jail centances as well as a smaller probability of arrest make this more likely to happen

4.3 Police Arresting

police move just the same way as the quiets and actives, However they cannot become active. The main characteristics Cops have is the ability to arrest this is done by them just happening to walking about and searching similarly to quiets "scanning" the area around them, except there "vision" is much smaller than that of the quiets and actives. IF an active is within a cops vision it then arrests them and either takes them off the board if the jail time is very high or puts them into "jail" a matrix storing all actives that have been caught by police. What an arrest would look like in my model

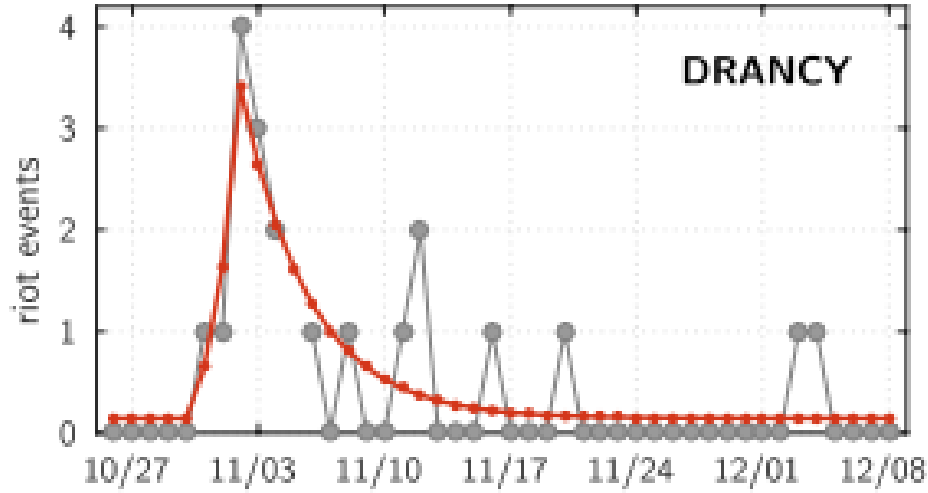


Figure 9: French riot

can be shown in figure 8 where on the left a cop is next to an active and in the next the active has been put into jail. The jail sentence J is set to 5 meaning it takes 5 turns for our agents to be set loose from our jail, not how the jail inmate's number next to him is the time he has spent in jail which goes down by 1 after the turn has taken place

5 My Results

What is expected of this model is for when i implement the smoke into the riot that the number of agents that are rioters will significantly decrease. to show this i will be looking at the total number of rioters within the grid at iteration n . These results should be similar to fig.4 from a suburb in Paris during the 2005 French riots. (ref 2.). Except for this model we are only looking from the peak of the riot until the eventual stop of the riot. This is as the probability for everyone to riot is very high at the beginning of the program due to there being many more civilians to police people.

Once i run my program i get graphs like fig. 5 where the blue line is a riot without any smoke and the red line is a riot with. We see that the 2 lines look as if they are perpendicular. This is because of the agent rule, the agent rule is the comparison of a linear equation on the LHS and a constant on the RHS so smoke decreases the number of people in the riot but does

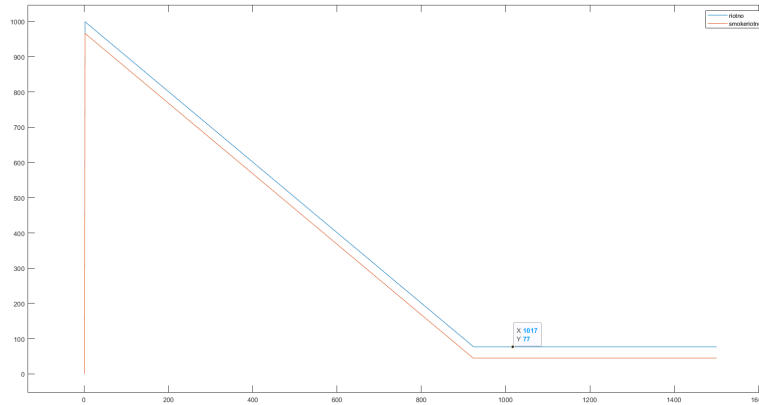


Figure 10: Riot With Smoke vs Without

not decrease the rate at the population of the riot tends to 0. If you want to increase the rate at which the number of rioting members tends to 0 you can do this by increasing the number of police people

Notice how there are still rioters at the end of the graph in fig 5? This is due to our rioters having a range greater than that of the distance of which the police can arrest them from, so the rioters become rioters once there are so few police in there vision but once there are too many within there vision they become a civilian again evading the law.

6 Evaluation

What went well

The model discussed is able to create a general riot and has key features in what creates a riot. From this Smoke was implemented which the police could throw which was shown to decrease the number of people rioting compared to people rioting when there is no smoke. showing the effect of obscuring vision in a riot

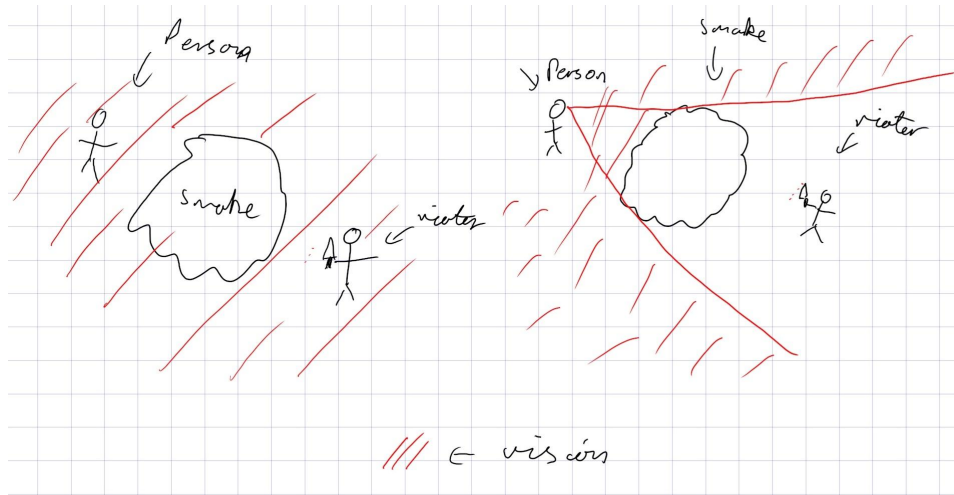


Figure 11: civilian vision

To Improve

I could make more realistic smoke, as in this current model civilians can see the side of the smoke that they are facing as well as the area on the other side of this smoke as shown in fig. 6 showing how my model works on the left and a more realistic vision on the right.

I have also made a pretty simple riot program where there are civilians rioters and police people. when rioters are near civilians they have a chance to make them rioters. When a police person is near a rioter they take them of the board. This has an epidemic like effect as the number of rioters multiplies depending on how many rioters there are giving us fig. 7.

I can then improve this model to show how smoke affects it by giving the rioters vision to seek out civilians to turn into rioters and then see how the obstruction of there vision via smoke affects how quickly the riot can be stopped. This works as an effective model as we know riots behave in an epidemic fashion as seen in the article (ref 2.)

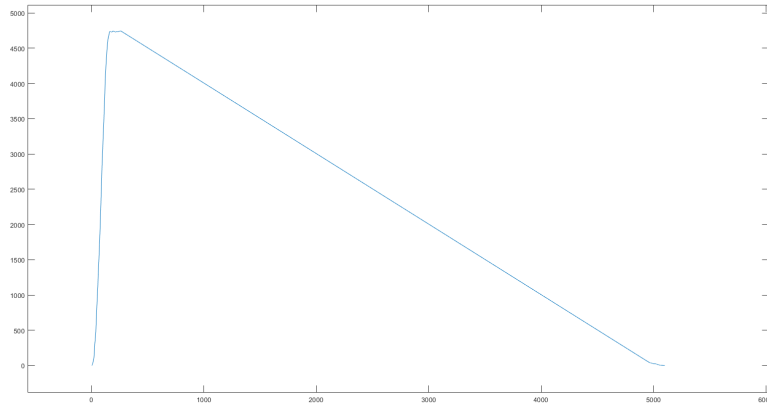


Figure 12: epedemic like program

7 Bibliography

References

1. Epstein, Joshua M. Modeling civil violence: An agent-based computational approach. 2002
2. Bonnasse-Gahot, Laurent and Berestycki, Henri and Depuiset, Marie-Aude and Gordon, Mirta B and Roché, Sebastian and Rodriguez, Nancy and Nadal, Jean-Pierre. Epidemiological modelling of the 2005 French riots: a spreading wave and the role of contagion. 2018

8 Appendices (Matlab Code)

here is my code for my riot simulation

```
1  clc
2  %%creating paramaters
3  crowd_n = 7000;
4  police_n = 3000;
5  grid_size = 100;
6  vision_l = 2;
7  smoke_size = 1;
8  %%number of iterations
9  no = 1500;
10
11  %%legitimacy of goverment
12  L = 0.1;
13
14  % T is a value which if  $G - N > T$  a active begins to
    riot (xpected utility
15  % of not expressing there private grievance)
16  T = 0.3;
17
18
19  % [crowd,police] = create_crowd_police(crowd_n,
    police_n);
20  % grid = rand_initialize_grid(crowd_n, crowd,
    police_n, police, grid_size);
21  % grid = agents_smoke(grid,grid_size,L,vision_l,T);
22  % sgrid = smoke_grid_init(grid_size);
23  %
24  % sgrid = smoke(grid,grid_size,smoke_size,sgrid)
25  %
26  %
27  % smokeriotno = run_till_no_smoke(grid_size,L,
    vision_l,T,crowd_n, police_n,no,sgrid,smoke_size);
28  % riotno = run_till_no_nosmoke(grid_size,L,vision_l,T,
    crowd_n, police_n, no);
29  % plot(1:length(riotno),riotno,1:length(smokeriotno),
    smokeriotno)
30  %
```

```

31 %
32 % legend('riotno ','smokeriotno ')
33 riotno = run_till_no_nosmoke_smoke(grid_size,L,
    vision_l,T,crowd_n, police_n, no,sgrid,smoke_size);
34 plot(1:length(riotno),riotno)
35
36
37 %%any functions
38 %{
39
40 run through a riot until all rioters have been arested
41 %}
42 function riotno = run_till_no_smoke(grid_size,L,
    vision_l,T,crowd_n, police_n, no,sgrid,smoke_size)
43 %create the grid
44 [crowd,police] = create_crowd_police(crowd_n,
    police_n);
45 grid = rand_initialize_grid(crowd_n, crowd,
    police_n, police, grid_size);
46 %create the smoke grid
47 sgrid = smoke_grid_init(grid_size);
48 %make agents rioters
49 grid = agents_smoke(grid,grid_size,L,vision_l,T,
    sgrid);
50 riotno = zeros(100,1);
51 agentno = zeros(100,1);
52
53 count = 1;
54 riotno(count,1)=0;
55
56 for i = 1:no
57
58     count = count+1;
59
60
61     [sgrid,grid] = policemen_smoke(grid, grid_size
        ,smoke_size,sgrid);
62
63     grid = agents_smoke(grid,grid_size,L,vision_l,
        T,sgrid);

```

```

64
65         riotno(count,1)=count_keys(grid,3);
66         agentno(count-1,1) = count_keys(grid,1);
67
68
69
70     end
71
72 end
73
74
75
76 function riotno = run_till_no_nosmoke_smoke(grid_size ,
        L,vision_l,T,crowd_n, police_n, no,sgrid,smoke_size
        )
77     %create the grid
78     [crowd,police] = create_crowd_police(crowd_n,
        police_n);
79     grid = rand_initialize_grid(crowd_n, crowd,
        police_n, police, grid_size);
80     %create the smoke grid
81     sgrid = smoke_grid_init(grid_size);
82     %make agents rioters
83     grid = agents_smoke(grid,grid_size,L,vision_l,T,
        sgrid);
84     riotno = zeros(no,1);
85     agentno = zeros(100,1);
86
87     count = 1;
88     riotno(count,1)=0;
89
90     for i = 1:no/2
91
92         count = count+1;
93
94
95         grid = policemen(grid, grid_size);
96         grid = agents_nosmoke(grid,grid_size,L,
            vision_l,T);
97

```

```

98         riotno(count,1)=count_keys(grid,3);
99         agentno(count-1,1) = count_keys(grid,1);
100
101     end
102     for i = 1:no/2
103
104         count = count+1;
105
106
107         [sgrid,grid] = policemen_smoke(grid, grid_size
108             ,smoke_size,sgrid);
109
110         grid = agents_smoke(grid,grid_size,L,vision_l,
111             T,sgrid);
112
113         riotno(count,1)=count_keys(grid,3);
114         agentno(count-1,1) = count_keys(grid,1);
115
116     end
117 end
118
119
120
121
122 function riotno = run_till_no_nosmoke(grid_size,L,
123     vision_l,T,crowd_n, police_n, no)
124     [crowd,police] = create_crowd_police(crowd_n,
125         police_n);
126     grid = rand_initialize_grid(crowd_n, crowd,
127         police_n, police, grid_size);
128
129     grid = agents_nosmoke(grid,grid_size,L,vision_l,T)
130     ;
131     riotno = zeros(100,1);
132     agentno = zeros(100,1);
133
134     count = 1;
135     riotno(count,1)=0;

```

```

132     for i = 1:no
133         count = count+1;
134
135
136         grid = policemen(grid , grid_size);
137         grid = agents_nosmoke(grid ,grid_size ,L,
            vision_l ,T);
138
139         riotno(count ,1)=count_keys(grid ,3);
140         agentno(count-1,1) = count_keys(grid ,1);
141
142
143
144     end
145
146 end
147
148 %%
149 function grid = agents_smoke(grid ,grid_size ,L,vision_l
    ,T,sgrid)
150     grid = move_rand(1, grid_size , grid);
151     grid = move_rand(3,grid_size , grid);
152     grid = agent_to_riot_smoke(grid ,grid_size ,L,
        vision_l ,T,sgrid);
153 end
154
155 function grid = agents_nosmoke(grid ,grid_size ,L,
    vision_l ,T)
156     grid = move_rand(1, grid_size , grid);
157     grid = move_rand(3,grid_size , grid);
158     grid = agent_to_riot_nosmoke(grid ,grid_size ,L,
        vision_l ,T);
159 end
160
161
162 function [sgrid , grid] = policemen_smoke(grid ,
    grid_size ,smoke_size ,sgrid)
163     grid = move_rand(2, grid_size , grid);
164     sgrid = smoke(grid ,grid_size ,smoke_size ,sgrid);
165     grid = riot_to_arrest(grid_size , grid);

```

```

166
167 end
168
169
170
171 function grid = policemen(grid, grid_size)
172     grid = move_rand(2, grid_size, grid);
173     grid = riot_to_arrest(grid_size, grid);
174
175 end
176
177
178
179
180
181
182
183
184
185
186 %{
187 grievance():
188     -Hardship G (physical or economic deprivation)
189       drawn from U(0,1) ]
190     uniform distrobution on 0,1
191
192     -Legitamacy L percived legitamacy of a regime.
193       This will be arbitary
194     number from 0,1
195
196 }%
197
198 function G = grievance(H,L)
199     G = H*(1-L);
200
201 end
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

```

204     -cops in the area C
205 C/A changes for every rioter depending on how many C's
      and A's are within
206 the vision of each crowd member within a set vision
207 %}
208 function P = arrest_probability(k,C,A)
209     P = 1 - exp(-(k*C/A));
210 end
211
212
213 %{
214 create_crowd_police()
215 -crowd_n:    number of people in crowd
216 -police_n:   number of police
217
218 -crowd:      our crowd with h r vals and key 1
219 -police:     our police with h r vals and key 2
220
221 creates the crowd aswell as the police such that:
222 -in the crowd each member has a number 1 in row1 to
      represent that they
223 are non rioters aswell as a hardship (H) row2 and a
      level of
224 risk they are willing to take (R) in row3. form:
225 [0,h1,r1; 0,h2,r2; ...; 0,hn,rn]
226
227 -in the police we have our key as 2 instead of 1 and
      we have H=R=0 as all
228 the police do is arrest and have no R H
229 %}
230 function [crowd,police] = create_crowd_police(crowd_n,
      police_n)
231     %take our H and R randomly from the uniform
      distribution between 0-1
232     H = unifrnd(0,1,1,crowd_n)';
233     R = unifrnd(0,1,1,crowd_n)';
234     %make a list of 1's from each person in the crowd
235     crowdkey = ones(1,crowd_n)';
236     %put crowd in the form
237     crowd = [crowdkey,H, R, zeros(1,crowd_n)'];

```

```

238     %same but police
239     police = [2*ones(1,police_n) ',zeros(1,police_n) ',
               zeros(1,police_n) ',zeros(1,police_n) '];
240 end
241
242 %{
243 rand_initialize_grid()
244 -crowd_n:    number of people in crowd
245 -crowd:      from create_crowd_police()
246 -ploice_n:   number of police
247 -police:     from create_crowd_police
248 -grid_size:  how big we want our grid
249
250 -grid:       our initialized grid
251 %}
252
253 function grid = rand_initialize_grid(crowd_n, crowd,
                                     police_n, police, grid_size)
254
255     %first start our grid out as being a grid of zeros
256     %size
257     for i = 1:grid_size
258         for j = 1:grid_size
259             grid{i,j} = [0,0,0,0];
260         end
261     end
262
263
264
265     %    now we put in our crowd, distrobuted randomly
266     %    throughout the grid
267     while crowd_n ~= 0
268
269         i = randi(grid_size,1);
270         j = randi(grid_size,1);
271         if grid{i,j}(1) == 0
272             grid{i,j}=crowd(crowd_n,1:4);
273             crowd_n = crowd_n - 1;
274         end

```



```

275     end
276
277
278     %now we put the desired number of ploice randomly
        into our grid
279     while police_n ~=0
280         i = randi(grid_size,1);
281         j = randi(grid_size,1);
282         if grid{i,j}(1) == 0
283             grid{i,j}=police(police_n,1:4);
284             police_n = police_n - 1;
285         end
286     end
287 end
288
289 %{
290 grid:         the grid
291 grid_size:    size of the grid
292 vision_l:     length of the vision
293
294 v:            vision at point i j
295
296 takes in the grid and a point of the grid, it then
        returns a subsection
297 ofthe grid around this point
298 %}
299 function v = vision(i,j,grid, vision_l, grid_size)
300
301     w = i - vision_l;
302     e = i + vision_l;
303     s = j + vision_l;
304     n = j - vision_l;
305     if w < 1
306         w = 1;
307     end
308
309     if e > grid_size
310         e = grid_size;
311     end
312

```

```

313     if s > grid_size
314         s = grid_size;
315     end
316
317     if n < 1
318         n = 1;
319     end
320     dns = s-n+1;
321     dew = e-w+1;
322     v = grid(w:e,n:s);
323
324     %v = reshape({ grid(w:e,n:s) },[dew,dns]);
325
326 end
327
328
329
330 %{
331
332
333 takes in the grid and checks the number of rioters
334     around each agent with
335 the number of police and uses equations to evaluate if
336     the agent becomes a
337 rioter aswell
338
339 function grid = agent_to_riot_nosmoke(grid,grid_size,L
340     ,vision_l,T)
341     %loop through each space in the grid
342     for i = 1:grid_size
343         for j = 1:grid_size
344             %if we land on an agent
345             if grid{i,j}(1)==1
346                 %find its grievance number
347                 G = grievance(grid{i,j}(2),L);
348
349                 %find its risk probablility
350                 R = grid{i,j}(3);

```

```

350
351 %find the vision at that point
352 v = vision(i,j, grid, vision_l, grid_size)
353 ;
354
355 %count number of police in vision
356 C=count_keys(v,2);
357
358 %count number of actives(rioters) in
359 vision
360 A=count_keys(v,3)+1;
361 A
362 %with our C and A we can now find the
363 probability for our guy
364 %at i j to be arrested
365 P = arrest_probability(-log(0.1),C,A);
366
367 %our agents net risk
368 N = R*P;
369
370 %expected utility of publicly expressing
371 ones private grievanc
372 gmn = G-N;
373
374 if gmn >T
375     grid{i,j}(1) = 3;
376 else
377     grid{i,j}(1) = 1;
378 end
379
380 end
381 end
382 %grid;
383 end
384
385

```

```

386
387 %{
388
389 turns agents into rioters taken into account the smoke
390 %}
391 function grid = agent_to_riot_smoke(grid,grid_size,L,
    vision_l,T,sgrid)
392     %loop through each space in the grid
393     for i = 1:grid_size
394         for j = 1:grid_size
395             %if we land on an agent
396             if grid{i,j}(1)==1
397                 %find its grievance number
398                 G = grievance(grid{i,j}(2),L);
399
400                 %find its risk probability
401                 R = grid{i,j}(3);
402
403                 %find the vision at that point
404                 v = vision(i,j, grid, vision_l, grid_size)
405                     ;
406
407                 %count number of police in vision
408                 C=count_keys(v,2);
409
410                 %count number of actives(rioters) in
411                 vision excluding those
412                 %with smoke
413                 A=count_keys_smoke(v,3,sgrid)+1;
414                 A
415                 %with our C and A we can now find the
416                 probability for our guy
417                 %at i j to be arrested
418                 P = arrest_probability(-log(0.1),C,A);
419
420                 %our agents net risk
421                 N = R*P;
422
423                 %expected utility of publicly expressing
424                 ones private grievance

```

```

421         gmn = G-N;
422
423         if gmn > T
424             grid{i,j}(1) = 3;
425         else
426             grid{i,j}(1) = 1;
427         end
428
429     end
430
431 end
432 end
433 %grid;
434 end
435
436
437
438
439
440
441
442 %{
443 grid:    the matrix you want to count the keys in
444 key:     the key you want to count
445
446 n:       the count
447
448 counts the number of a given key in a given matrix
449 %}
450 function n = count_keys(grid, key)
451     n=0;
452     [r,c] = size(grid);
453     for x = 1:r
454         for y = 1:c
455             if grid{x,y}(1)==key
456                 n=n+1;
457             end
458
459         end
460     end

```

```

461 end
462
463 %{
464
465 count keys in matrix excluding the ones with smoke on
    them
466 %}
467
468
469 function n = count_keys_smoke(grid,key,sgrid)
470     n=0;
471     [r,c] = size(grid);
472     for x = 1:r
473         for y = 1:c
474             if grid{x,y}(1)==key && sgrid{x,y}(1) == 0
475                 n=n+1;
476             end
477
478         end
479     end
480 end
481
482 %{
483 grid_size:    size of grid
484 grid:         current grid
485
486 grid:         updated grid
487
488 looks at every police person and arrests a rioter if it
    is near by
489 %}
490
491 function grid = riot_to_arrest(grid_size, grid)
492     %for every (i,j)th place on the grid
493     for i = 1:grid_size
494         for j = 1:grid_size
495             %if the (i,j)th place is a policeman
496             if grid{i,j}(1) == 2
497                 %check every square 3*3 around the
                    police man

```

```

498         for n = -1:1
499             for m = -1:1
500                 %assignn new (i,j)th position
                    we are considering
501                 positioni = i + n;
502                 positionj = j + m;
503                 %check point is on grid and it
                    is a rioter
504                 if positioni <= grid_size &&
                    positionj <= grid_size &&
                    positioni > 0 && positionj
                    > 0 && grid{positioni ,
                    positionj}(1) == 3
505                     %arrest that rioter
506                     grid{positioni , positionj}
                        = [0,0,0];
507                     return
508
509                 end
510             end
511         end
512     end
513 end
514 end
515 end
516
517 %{
518 crowd_type:      the type of crowd you want to move
519 grid_size:       the size of the grid
520 grid:            the current grid
521
522 grid:            the grid after all the agents of a
                    certain type have moved
523 %}
524
525 function grid = move_rand(crowd_type , grid_size , grid)
526     v = 0;
527     %for every (i,j)th place on the grid
528     for i = 1:grid_size
529         for j = 1:grid_size

```

```

530         %while we reach a place with the correct
           crowd type, and we
531         %have made a certain number of attempts v
532         while grid{i,j}(1) == crowd_type && v < 9
533             %create a random number -1 to 1 for n-
               s plane and w-e plane
534             %and take the absolute value to get
               rid of negatives
535             directionns = i + randi([-1,1],1);
536             directionwe = j + randi([-1,1],1);
537             v = v + 1;
538             %check if the new position is inside
               of the grid and not =0
539             if directionns <= grid_size &&
               directionwe <= grid_size &&
               directionns > 0 && directionwe > 0
540                 %if that space is empty (=0)
541                 if grid{directionns, directionwe}
                   (1) == 0
542                     %move that person to that
                       location
543                     grid{directionns, directionwe}
                       = grid{i,j};
544                     %remove that person from where
                       he was previously
545                     grid{i,j} = [0,0,0];
546                 end
547             end
548         end
549     end
550 end
551
552 end
553
554
555
556 function sgrid = smoke(grid, grid_size, smoke_size, sgrid
    )
557     %go through each square on grid
558     for i = 1:grid_size

```



```

559         for j = 1:grid_size
560
561             if grid{i,j}(1) == 2
562                 %throw smoke at fools
563                 sgrid = throwsmoke(i,j,grid,smoke_size,
564                                     grid_size,sgrid);
565             end
566         end
567     end
568 end
569
570
571
572
573 function sgrid = throwsmoke(i,j,grid,smoke_size,
574                             grid_size,sgrid)
575     L = length(smoke_size);
576     %search around the police officer for rioters
577     for n = -L:L
578         for m = -L:L
579             %assignn new (i,j)th position we are
580             %considering
581             positioni = i + n;
582             positionj = j + m;
583
584             %check point is on grid and it is a rioter
585             if positioni <= grid_size && positionj <=
586                 grid_size && positioni > 0 && positionj
587                 > 0 && grid{positioni,positionj}(1) ==
588                 3
589                 %create an square area of length
590                 vision_l around our rioter
591                 w = positioni - smoke_size;

```

```

592         w = 1;
593     end
594
595     if e > grid_size
596         e = grid_size;
597     end
598
599     if s > grid_size
600         s = grid_size;
601     end
602
603     if north < 1
604         north = 1;
605     end
606
607     for k = w:e
608         for o = north:s
609
610
611             sgrid{k,o}(1) = 1;
612         end
613     end
614
615
616     end
617 end
618 end
619 end
620
621
622
623
624 function sgrid = smoke_grid_init(grid_size)
625     sgrid = {};
626     for i = 1:grid_size
627         for j = 1:grid_size
628             sgrid{i,j} = [0,0];
629         end
630     end
631 end

```