Tension Propagation in Social Networks

Yassin Bahid

In the past decades, the rise of social media platforms such as Facebook Twitter, and others have fundamentally changed the way social interactions happen. Suddenly, information's transmission rate has known an exponential increase. These new ways of connecting have led to the spread of moral outrage at an unseen rate. Protests, riots, and even revolutions are organized online. The availability of data stemming from these events offers an unprecedented chance to model the spread of outrage in a population. We seek to develop a theoretical framework capable of predicting this spread.

1 Introduction:

The prominence of social media platforms such as Twitter has fundamentally changed interpersonal relations and communications [9]. This societal shift can be seen in how protests arise and are conducted [11]. Whether it is in the Arab spring [8] or more recently during the BLM protests after George Floyd's murder in 2020, social media platforms have played a significant role in the mobilization and organization of protests. One key component to understanding the role of social media in protest is to understand how tension propagates in such a network.

Previous research [7] has focused has focused on using SIR and SIS models to describe the spread of information in the system. These models are epidemiological in nature. They separate agents in a network into three different categories: susceptible, Infected, and recovered(or susceptible again). Agents interact with each other and move from one state to the other with different probabilities offering a While these models have proven their efficacy on numerous occasions, they are threshold models at their core. In that, they only categorize the nodes and offer no real insight into how much tension each vertex holds. At every moment in time, the reaction of different people to different information can vary widely depending on personal sensitivities, socio-economic situation, cultural background, and political affiliation. Thus, a model that accurately describes the tension in a network of individuals needs to portray said notion as a continuous real variable.

1.1 Defining Tension:

We define **Tension** as the stress resulting from a perceived amoral violation.[4]. While this definition might seem highly subjective, studies[6] show that the perceived moral violation has to do more with the reactions around them than any moral convictions. The moral outrage of a particular individual is shaped in two different ways. First, people will change their morale outrage over time through the interaction with other morally outraged people through either positive or negative feedback. Second, they change their internal norms based on the perceived norms of others. Social Media platforms are designed to reinforce both types of change, by offering quantifiable measures of responses in the form of shares and likes. Using these principles, we can infer two main assumptions about tension within a social network:

- Tension is a continuous variable.
- Tension is diffused upon the interaction between different agents in the network.

1.2 Goals:

Through this research project, we seek to develop a robust and modular model to describe the behavior of tension as defined above. We will prioritize Numerical methods that focus on speed without sacrificing too much accuracy. We shall apply our problem to different types of random graphs and study the behavior such a model has on different nodes. We will also look at the dynamics between different communities organized in a hierarchical way.

2 Model:

2.1 Diffusion Equation:

In order to fulfill our assumptions, we find that Diffusion equations are the best tool to model such a system. In physics, given random microscopic interactions between particles, Diffusion equations are the macroscopic behavior of the system. They were first developed by Adolf Fick in 1855. His experiments focused on trying to follow the concentration of certain particles in a solution as they move between connected containers. He then derived a law that some diffusion process must follow. Commonly known as Fick's first law, this rule states that the diffusive flux is related to the gradient of the concentration, i.e:

$$J = -D\frac{\partial T}{\partial x}, \tag{1}$$

where:

- J is the diffusion flux. its dimension is the amount of substance per unit area per unit of time,
- T is the concentration of the value we are following the diffusion of
- D is the diffusion coefficient governing the rate o diffusion of quantity T.

From equation 1, one can derive a parabolic partial differential equation that governs the behavior of T. Given the preservation of mass, or the in the case of a network the preservation of the number of nodes:

$$\begin{aligned} \frac{\partial T}{\partial t} + \frac{\partial}{\partial x}J = 0 &\implies \frac{\partial T}{\partial t} + \frac{\partial}{\partial x}(-D\frac{\partial T}{\partial x}) = 0 \\ &\implies \frac{\partial T}{\partial t} = D\frac{\partial^2 T}{\partial x^2} \end{aligned}$$

Thus the change in the quantity T is analogous to the outward flux at each point towards the others. We modify this equation to account for the self-leveling of the property of tension. In that it decreases after some time. We also account for the random influx of information into the system. We assume that this information is naturally translated into tension. Thus, the equation governing the tension in a certain system will have the following form:

$$\frac{\partial T}{\partial t} = D\Delta T + T(1 - T) + B(t), \quad (2)$$

Where B(t) is random Brownian noise expressing the tension resulting from the influx of information into the system.

2.2 Diffusion Equation on a Graph:

Equation 2 expresses the diffusion of tension on a continuous space. Thus, we need to modify the said equation to fit a graph. In order to do so, we need to define 3 concepts:

- What is a function over a graph?
- What is the gradient over a graph?
- What is the Flux over a graph?

2.2.1 Function over graph:

Given a graph $G(V, E)$ Sometimes referred to as **graph signal**, it is a mapping from the space V and a time t of vertices to the space $\mathbb{R}$ of reals. In our case, T is a function that assigns a different real value to each node at each time t.

2.2.2 Gradient of a graph:

Given a Euclidean space, the gradient operator expresses the rate of change of the graph signal in all directions. A graph represents a discrete connection of vertices. Therefore, the rate of change only makes sense in the case where two edges are connected. As for the direction of the change, a graph allows for different directions. However, the direction chosen should not affect the outside flux as we will see in the following sections. So given an edge $e = (u, v)$ m we define the gradient of T at each edge e as:

$$grad(T)|_e\,(t) = T(u, t) - T(v, t)$$

In order to nicely represent the gradient matrix at each edge, we use an **Incidence Matrix**: An nxm logical matrix that represents the relationship, where n is the number of vertices, and m is the number of edges. Given an edge $e = (u, v)$ and a chosen direction we define the incident matrix as such:

$$K_{u,e} = +1$$

$$K_{v,e} = -1$$

Thus, the gradient matrix of T, which represents the rate of change in the entire Matrix, is defined as:

$$grad(T)(t) = K^T T(t)$$

where K^T is the transpose of T, and $T(t)$ is the vector of tensions at all vertices at time t

2.2.3 Graph Laplacian:

On a continuous spectrum, the Laplacian of a function is the derivative of the gradient. Analogously, we define the **Graph Laplacian** represented by Δ_g as the divergence of the gradient Thus:

$$\begin{aligned}\Delta_g T &= div(grad(T)(t)) \\ &= KK^T T(t)\end{aligned}$$

This nxn represents the outward flux of any given node towards its nearest neighbors.

2.2.4 The Graph Partial Differential Equation:

We now have all the pieces to translate our differential equation into a graph. Given a graph $G(V, E)$, the partial differential equation governing the tension diffusion over each node is:

$$\partial_t T = D\Delta_g T + T(1 - T) + B(t) \quad (3)$$

where now B is a vector of Brownian motion representing the tension coming from the chaotic influx of information into each node.

2.3 Numerical Solution:

Given a time interval $[0, t_f]$ and an initial state T_0 of tensions per node, we use the forward Euler scheme in order to generate a solution. We discretize our time interval into n equal time steps. For each time step i, the change in tension T_i is:

$$\begin{aligned}&\frac{T_i - T_{i-1}}{t_i - t_{i-1}} = D\Delta_g T_{i-1} + T_{i-2}(1 - T_{i-1}) + B(t_{i-1}) \\ &\iff T_i = T_{i-1} + \delta t(D\Delta_g T_{i-1} + T_{i-1}(1 - T_{i-1}) + B(t_{i-1})) \\ &\iff T_i = T_{i-1} + \delta t(DKK^T T_{i-1} + T_{i-1}(1 - T_{i-1}) + B(t_{i-1}))\end{aligned}$$

where δt is the time difference between two time steps.

Error and Implementation

Using the Taylor expansion of T around a time step:

$$\begin{aligned}T(t_{n+1}) &= T(t_n + \delta t) \\ &= T(t_n) + \delta t T'(t_n) + \frac{1}{2}\delta t^2 T''(t_n) + O(\delta t^3)\end{aligned}$$

Hence, the local truncation error LTE is:

$$LTE = T(t_{n+1}) - T(t_n) \quad (4)$$

$$= \frac{1}{2}\delta t^2 + O(\delta t^3) \quad (5)$$

Therefore the local accuracy is quadratic $O(\delta t^2)$.

Given a population n, the implementation of the forward Euler costs $n^2.m$, where m in the number of iterations wanted.

2.4 A small Example:

Before moving to big social networks, we focus on a smaller network to exemplify the inner workings o our model. We use a graph with 6 nodes defined as such:

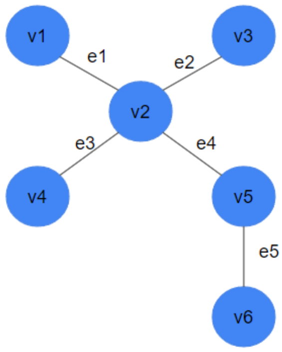


Figure 1: Small 6 vertices and 5 edges Network

2.4.1 Incidence matrix:

We choose to direct the graph such that the positive direction is from the node with the lesser label to the one with the higher one. Thus, the incidence matrix of our particular network is:

$$K = \begin{array}{c} \left(\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & v1 \\ -1 & 1 & 1 & 1 & 0 & v2 \\ 0 & -1 & 0 & 0 & 0 & v3 \\ 0 & 0 & -1 & 0 & 0 & v4 \\ 0 & 0 & 0 & -1 & 1 & v5 \\ 0 & 0 & 0 & 0 & -1 & v6 \\ \hline e1 & e2 & e3 & e4 & e5 & \end{array}\right) \end{array}$$

2.4.2 Graph Laplacian:

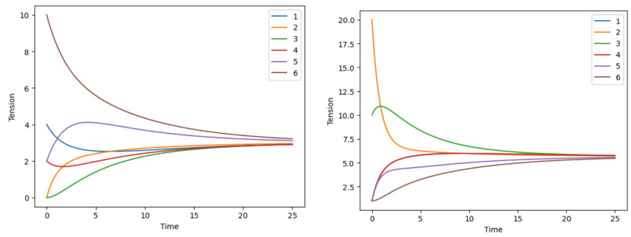
Given the above incidence matrix, the graph Laplacian of the given system is:

$$\Delta_g = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

2.4.3 Numerical Solution:

We choose to ignore the Brownian motion, for now, to focus on the dynamic of the system. We set the diffusion coefficient $D = 0.65$ We use the following two different initial states, and run the simulation for 50 time steps.

$$T_1(0) = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 2 \\ 2 \\ 10 \end{bmatrix} \qquad T_2(0) = \begin{bmatrix} 1 \\ 20 \\ 10 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$



(a) $T_1(0) = \begin{bmatrix} 4 & 0 & 0 & 2 & 2 & 10 \end{bmatrix}$ **(b)** $T_2(0) = \begin{bmatrix} 1 & 20 & 10 & 1 & 1 & 1 \end{bmatrix}$

Figure 2: Tension evolution in a 6 node system with diffusion coefficient $D = 0.65$ and initial state $T_1(0)$ in a, and initial state $T_2(0)$ in b.

In figure 2a the tension flows inward from the outward nodes 1 and 6 towards the inside of the network to more centrally located nodes such as node 2. In figure 2b, the opposite happens. The tension flows from node 2 towards the outward nodes such as nodes 1 and 6. Node 3, which was already In both cases, the model reaches an equilibrium state in which all nodes' tension stabilizes around an equilibrium point.

As one might suspect, the addition of more nodes, the change in connections, and different diffusion coefficients have a wide array of effects on the dynamic behavior of the system. In the following sections, we seek to study said effects.

3 Results:

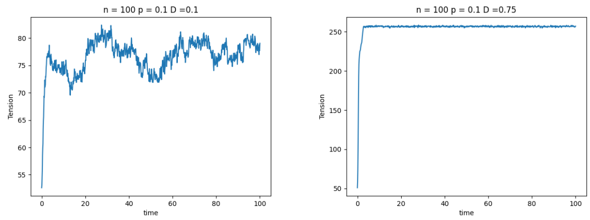
In order to study the behavior of our model, we choose to look at different random graph configurations. First, we look and Erdos-Reyni graphs, before moving to Chung-Lu graphs, and finally Hierarchical Network configuration. We shall look at the effect of each graph and different diffusion coefficients on the dynamic.

3.1 Erdos-Reyni:

First introduced in 1959, Erdos-Reyni graphs[5] are a particular subset of random graphs. Given a fixed number of vertices n, we construct the graph by connecting nodes with a certain probability p of our choosing. We will look at the effect of different probabilities p and diffusion coefficients D on different population graphs. We choose a random initial state for all nodes from a normal distribution between 0 and 1. We run the simulation for a time frame between 0 and 100. We do run the Forward Euler method with a step size of 1000 to reduce the local truncation error.

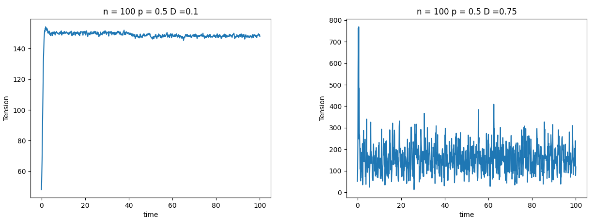
3.1.1 Population $n = 100$:

We first look at a population of 100 nodes:



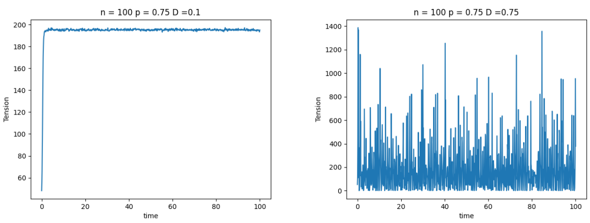
(a) - **(b) -**

Figure 3: Erdos-Reyni total tension for a population $n = 100$, $p = 0.1$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)



(a) - **(b) -**

Figure 4: Erdos-Reyni Total tension for a population $n = 100$, $p = 0.5$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)

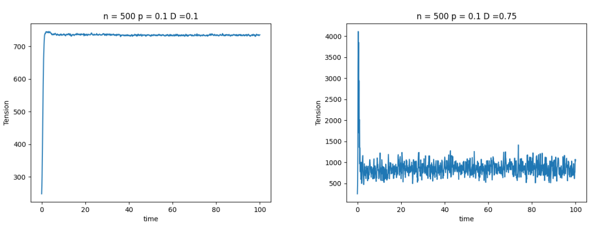


(a) - **(b) -**

Figure 5: Erdos-Reyni Total tension for a population $n = 100$, $p = 0.75$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)

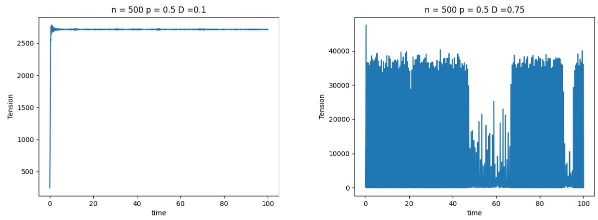
For a population of 100, it seems that keeping the diffusion coefficient low leads the system to stay in equilibrium. If the diffusion coefficient is high, the tension spikes before assuming a wave-like shape. Increasing the probability of connection p does increase what said equilibrium is when the Diffusion is small. This increase in p causes an increase in the magnitudes of the spikes when the diffusion is high enough to cause a wave-like behavior.

3.1.2 Population $n = 500$:



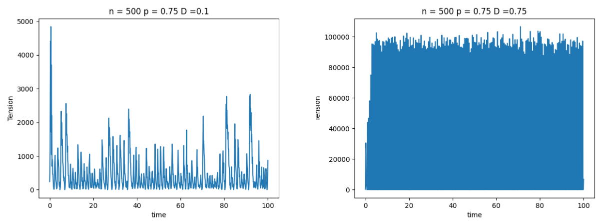
(a) - **(b) -**

Figure 6: Erdos-Reyni total tension for a population $n = 500$, $p = 0.1$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)



(a) - **(b) -**

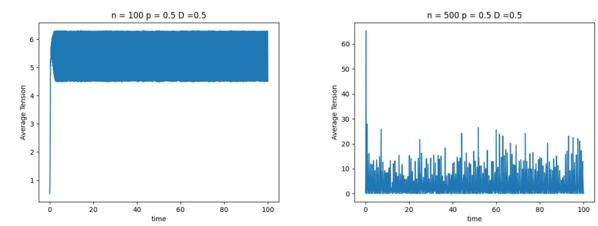
Figure 7: Erdos-Reyni Total tension for a population $n = 500$, $p = 0.5$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)



(a) - **(b) -**

Figure 8: Erdos-Reyni Total tension for a population $n = 500$, $p = 0.75$, and $D = 0.1$ in (a) and $D = 0.75$ in (b)

There seems to be a very similar behavior to the smaller population. There are three main differences that are more apparent in the figures presented in **Appendix A**. The first is that as the population grows, the diffusion and population needed in order to achieve a wave like behavior is smaller. The second is that the average tension per node for the same Diffusion Coefficient and probability of connection is higher as seen in figure 9. It is worth noting that the small variation seen in 9b is due to the Brownian motion and stays relatively small compared to the statistic studied. While the probability of connection between nodes is the same, given the higher population, each node will be more likely to have more neighbors as the population gets larger. Thus more tension will flow within each individual node. The final observation is that given a high Diffusion equation and a high probability of connection, the system reaches a high-level variation while still reaching high levels of tension. Hence, small changes in tension in one node tend to be diffused to more nodes more prominently which explains the high level of variation.



(a) - **(b) -**

Figure 9: Erdos-Reyni Average tension for populations $n = 100$ (a), and $n = 500$, with $p = 0.5$, and $D = 0.5$ in (a) and $D = 0.75$ in D

3.2 Different Degree Nodes:

So far, we have only looked at graphs with relatively homogeneous degrees. As one might suspect graphs that do display discrepancies in degree between nodes lead to different behaviors. We seek to study such behavior if the following section. We use Chung-Lu graphs[3], where given a number of nodes n, and an expected degree vector $\vec{k}$, we generate a random graph where the probability of connection between two nodes u, and v is

$$p_{u,v} = \frac{k_u k_v}{\sum_{i=0}^{n} k_i}$$

Hence, for each node u, it's expected degree is:

$$E(deg(u)) = k_u(1 - \frac{k_u}{\sum_{i=0}^{n} k_i})$$

We run the simulation for a time interval of 0 to 100 for 1000 steps. We run the dynamic for 5000 randomly generated Chung-ly graphs for different populations. We compute the maximum tension reached for each degree, and the time needed to reach it. We do so for multiple population sizes.

3.2.1 All Degrees Represented:

First, we look at graphs with a varying number of degrees, from sparsely connected nodes to fully connected ones.

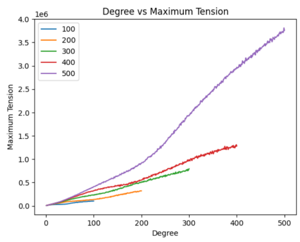


Figure 10: Average Maximum tension attained per degree in a Graph with every degree is represented for a population = $100, 200, 300, 400, 500$**.**

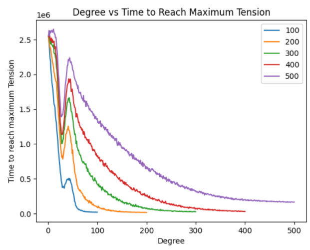


Figure 11: Average time to reach maximum tension attained per degree in a Graph with every degree is represented for a population = $100, 200, 300, 400, 500$**.**

In Figure 10, one can see that regardless of the population, as the degree gets higher, the maximum tension reached does too. As the population grows, just as one sees in the case with Erdos-Reyni graphs, the maximum tension observed grows for all nodes. When one looks at the time at which these maximums are reached, in figure 11, an interesting pattern emerges. There appears to be a dip in the time needed to reach the maximum tension for nodes with degrees between 25 and 100. There is a slight shift to the right of the degrees of said dip as the population grows. However, the shift stays relatively minimal. We attribute this shift to the way the Chung-Lu graph generation algorithm operates. The vertices affected have a high probability of being connected to very high degree nodes which reach their maximum tension very fast given their high connectivity with all other nodes. However, the nodes experiencing the dip in time have a lower probability of being connected to other nodes in between them and the high degree vertices. Hence, once the tension from the very high degree nodes reaches them, they are unlikely to receive any more tension from other nodes, which leads to them attaining their maximum quickly. The vertices with even smaller degrees are relatively isolated. Therefore, tension takes longer to get diffused to them and thus needs a longer time to reach a maximum.

3.2.2 One Popular Node :

Under this model, we choose to make one node's connections significantly larger than others. For example, for a population of 100, we choose one random node for which the degree will be larger than 90 and choose a random degree for all other nodes that is less than 20. We shall look at the differences in maximum tension and time needed to reach it between the popular node and the "regular", low degree nodes.

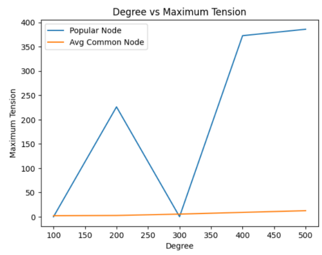
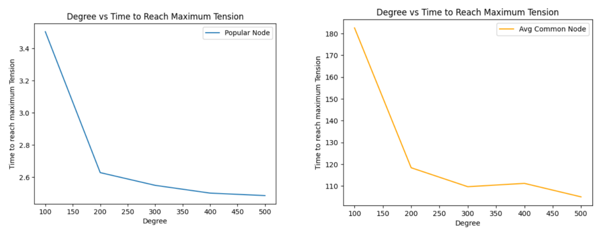


Figure 12: Maximum Tension reached by the most popular nodes and the average tension reached by non-popular nodes for different populations.



(a) - **(b) -**

Figure 13: Average time needed to reach maximum tension for popular node vs lower degree nodes.

One can see a rather strange pattern for the popular node. As one can see in fig. 12, the maximum tension for the popular node drops significantly for a population of size 200. While probably due to the highly stochastic nature of the model. This figure also highlights the independence that the popular node has from the common node, as they do not dip the same way. Thus the popular node does not necessarily influence the common nodes as heavily as one might expect under this model. The times at which said maximums occur seem to reflect each other. while the lower degree nodes do need a significantly longer time to reach the maximums, the two functions mimic their rate of change.

3.3 Hierarchically Structured Population

In an effort to accurately encapsulate the inequality of people's interaction, Watts [10] proposes a hierarchical structure under which a population N is divided into different communities c. Each community has the same number of agents $n = \frac{N}{c}$. It is assumed that each community is fully connected. In that, we assume that all nodes interact with each other, and will diffuse tension to one another. The communities are all assumed to be organized in a tree like fashion, fig 14. The distance between each community is defined as the inverse of the tree distance between said nodes.

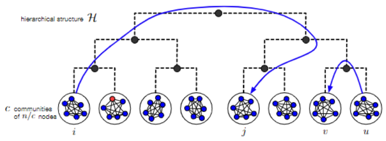


Figure 14: Hierarchy of fully connected nodes.[1]

This sort of hierarchy is well-adjusted to model the spread of dynamics between tight, highly connected communities, that are somewhat separated. For example, multiple users of different subreddits under the hierarchical organization that is Reddit. Or, multiple graduate students in each department under the organization that is a university. Within each community, the tension will be diffused as it does in the regular model. At each time step, we decide whether or not there will be interactions between two nodes with probability p. If there is interaction, two nodes from two randomly chosen communities diffuse tension between each other. The probability with which said two nodes depends on the inverse of the distance between the communities. Unlike the model used in [10], at no point do nodes physically move from one community to another. We also omit the Brownian motion represented by $B(t)$ in 2. We initialize our model with one random node from a random community with an extremely high tension while setting all other node's tensions to 0. Using these specific last two conditions, we mimic the tracking of one particular topic causing tension in the whole system. We seek to study the effect of different combinations of D and p on the behavior of the system. To summarize, we run the following algorithm given a particular set of p and q:

- 1 - Choose a random node that has high tension and set all others to 0.
- 2 - At each time step, we execute one forward Euler step to find the new tension for each node.
- 3 - Afterwards, If interaction happens with probability p,
- 3.1 - Choose two nodes from two communities with probability inversely $\propto$ distance.
- 3.2 - Share tension of the two nodes with diffusion D

$$T(node_1, t_i) = T(node_1, t_i) + D * T(node_2, t_i) - D * T(node_1, t_i)$$

$$T(node_2, t_i) = T(node_2, t_i) + D * T(node_1, t_i) - D * T(node_2, t_i)$$

- 3.3 - Repeat steps 2 to and 3

we run our model for a population of 800 divided into 8 communities. The initial tension for the randomly chosen node is 100.

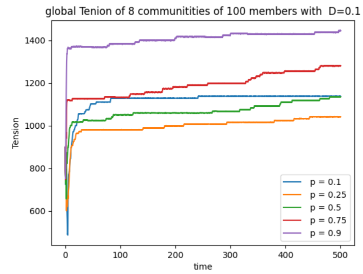


Figure 15: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.1$

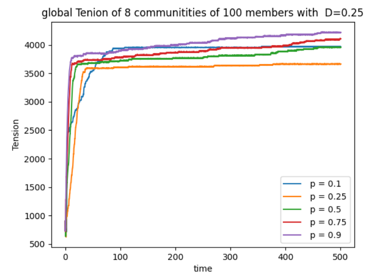


Figure 16: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.25$.

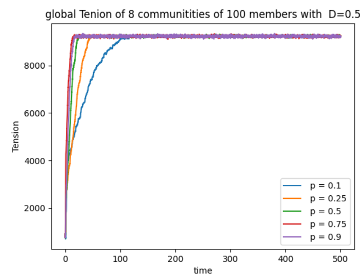


Figure 17: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.5$.

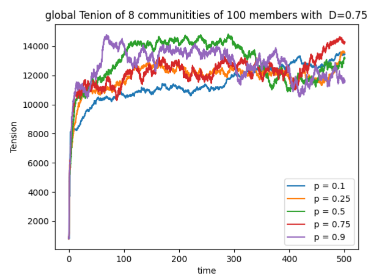


Figure 18: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.75$.

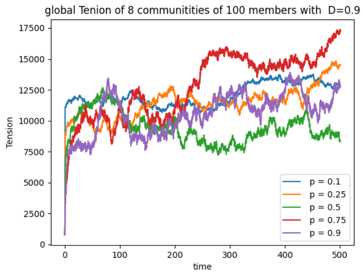


Figure 19: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.25$.

Looking at figures 20 to 24, one can see the different effects of the diffusion and probability of interaction. If the diffusion coefficient is low, there is not much activity regardless of the probability p. This observation means that regardless of the probability p, and given enough time, an interaction is bound to occur. Thus spread of information will happen between communities although be it at a later date. The model reaches an equilibrium state for all probability p. It is only when the diffusion reaches 0.75 or higher that we see any spikes and activity. When $D = 0.9$, the model starts growing. One possible interpretation is that under this model, in order for the tension to grow and expand to multiple communities and survive, they need to have a high level of "empathy" with each other. In that, although there is clear communication between all groups, represented with p, they need to deeply care about each other in order to properly spread tension.

3.4 Conflicting Communities Dynamic

So far, we have only looked at what could be considered communities with "homogeneous" beliefs. Our model assumes that interaction between the different communities will always lead to diffusion which means a reduction in the tension for the node holding the highest value. Realistically, certain interactions will increase the tension, such as exchanges between communities belonging to different political ideologies, or even the interactions between different gangs in real life[2]. In order to model these interactions, we modify our algorithm such that interactions between nodes from different communities add to the tension. Step 3.2 from the previous section's algorithm becomes:

$$T(node_1, t_i) = T(node_1, t_i) + A * T(node_1, t_i)$$

$$T(node_2, t_i) = T(node_2, t_i) + A * T(node_2, t_i)$$

where A is a constant that expresses the amount with which the tension increases after friction between two nodes. The amount added to each Agent depends on the tension already present in said node. Hence, under our model, the tension, or outrage already present in node makes it more sensitive to any attacks on its moral and increases its outrage. We fix $A = 2$ and study the effect of p and D. We run the model for two communities only of 100 agents.

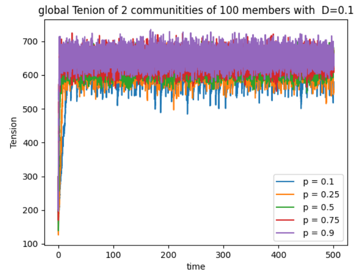


Figure 20: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.1$

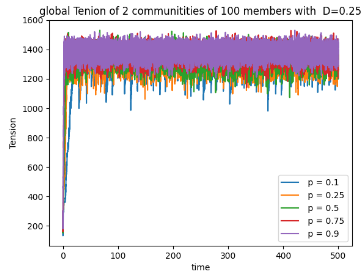


Figure 21: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.25$.

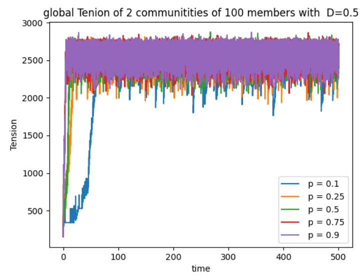


Figure 22: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.5$.

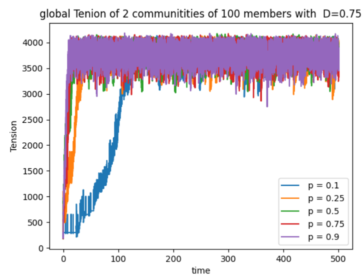


Figure 23: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.75$.

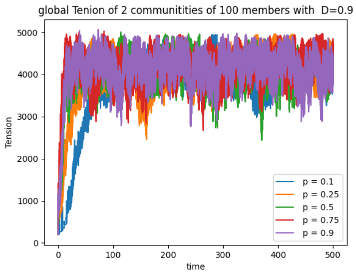


Figure 24: Total System Tension for different probabilities of community with diffusion coefficient set to $D = 0.25$.

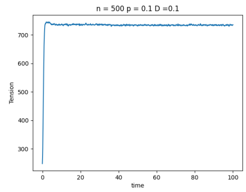
The model now stabilizes at every Diffusion coefficient. It does take longer to reach said maximum if p is smaller. The maximum grows with D. However, as D grows there is more instability and variation. The behavior is due to the nonlinearity of the system. The high diffusion makes it so that anytime an agent reaches a maximum tension, it is diffused through the system.

4 Conclusion:

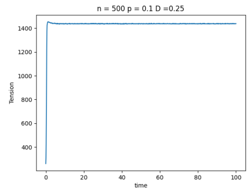
The diffusion equation developed in this paper is a powerful tool to model the different ways agents interact and spread moral outrage. In the case of an Erdos-Reyni type configuration, we saw the effect of different population sizes on the variability and behavior of said graphs given different diffusion equations. Under the Chung-Lu random graph model, we saw interesting behaviors that arise from the algorithms used to build these graphs. We saw how outrage can reach a certain interval of degrees a lot faster than others. When looking at hierarchically connected communities, we see how little the rate of interaction between communities matters. This observation can be interpreted as the natural way information will eventually reach enough people, given enough time. It is only when there is empathy, seen as a high diffusion coefficient in our model, between the different communities that we see an increase in overall tension in the system. When two communities are actively fighting with each other, a very different pattern occurs, where high diffusion, means a higher maximum occurs, but with a higher level of variation.

This Tension Diffusion Equation model for graphs offers a good first step in modeling outrage propagation. In future work, we seek to put this dynamic to the test using real-life data. We will also modify the model to reflect the translation between active social engagement in a virtual platform and real life actions such as protests.

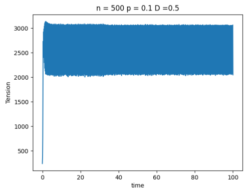
Appendix A:



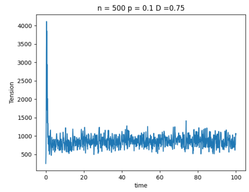
(a) -



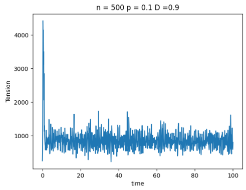
(b) -



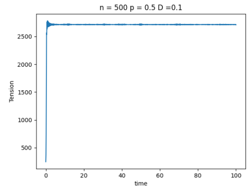
(c) -



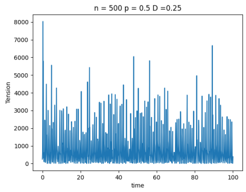
(d) -



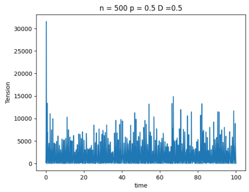
(e) -



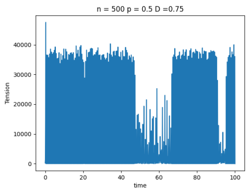
(f) -



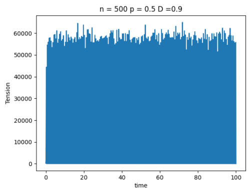
(g) -



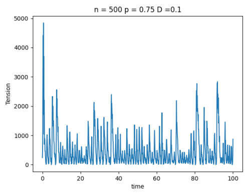
(h) -



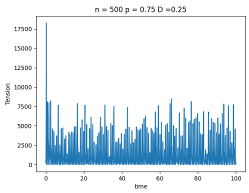
(i) -



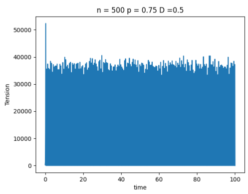
(j) -



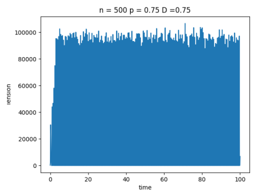
(k) -



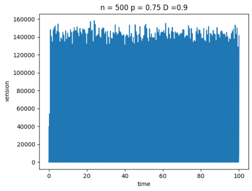
(l) -



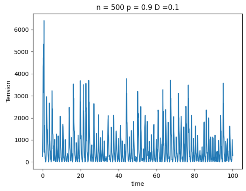
(m) -



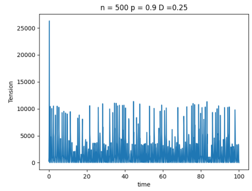
(n) -



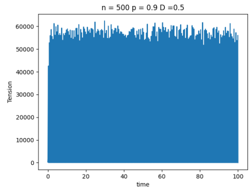
(a) -



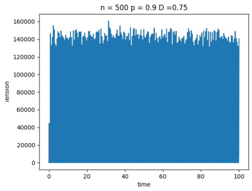
(b) -



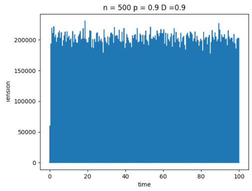
(c) -



(d) -



(e) -



(f) -

References

[1] Advanced spreading processes.

[2] Alethea BT Barbaro, Nancy Rodriguez, Havva Yoldaş, and Nicola Zamponi. Analysis of a cross-diffusion model for rival gangs interaction in a city. *arXiv preprint arXiv:2009.04189*, 2020.

[3] Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2):125–145, 2002.

[4] Phoebe C Ellsworth, Klaus R Scherer, Richard J Davidson, and H Hill Goldsmith. Handbook of affective sciences, 2003.

[5] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[6] Herbert Gintis, Samuel Bowles, Robert T Boyd, Ernst Fehr, et al. *Moral sentiments and material interests: The foundations of cooperation in economic life*, volume 6. MIT press, 2005.

[7] Maryam Maleki, Esther Mead, Mohammad Arani, and Nitin Agarwal. Using an epidemiological model to study the spread of misinformation during the black lives matter movement. *arXiv preprint arXiv:2103.12191*, 2021.

[8] Adam Smidi and Saif Shahin. Social media and social mobilisation in the middle east: A survey of research on the arab spring. *India Quarterly*, 73(2):196–209, 2017.

[9] Kalpathy Subramanian. Influence of social media in interpersonal communication. *INTERNATIONAL JOURNAL OF SCIENTIFIC PROGRESS AND RESEARCH (IJSPR)*, 109:pages 70–75, 08 2017.

[10] Duncan J Watts, Roby Muhamad, Daniel C Medina, and Peter S Dodds. Multiscale, resurgent epidemics in a hierarchical metapopulation model. *Proceedings of the National Academy of Sciences*, 102(32):11157–11162, 2005.

[11] Thomas Zeitzoff. How social media is changing conflict. *Journal of Conflict Resolution*, 61(9):1970–1991, 2017.