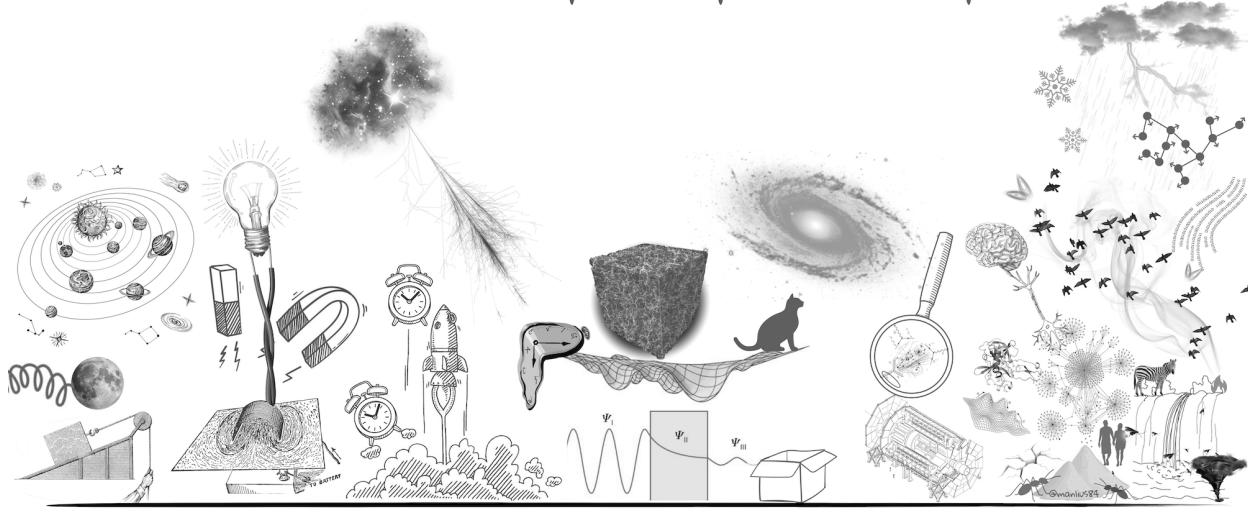


UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

## Areas of physics by complexity



Newton's  
Mechanics

Electro-  
Magnetism

Special  
Relativity

Quantum Mechanics  
General Relativity

Quantum  
Field Theory

Complexity  
Science

## Physics of Complex Networks: Structure and Dynamics

# Final Report

Student: Zara Miriam

Last update: July 15, 2024

# Contents

---

<b>1 Task 01: Ising Model</b>	<b>1</b>
1.1 Task Description . . . . .	1
1.2 Mathematical Model . . . . .	1
1.3 Numerical Simulations . . . . .	2
Appendix . . . . .	5
<b>2 Task 18: Turing Patterns</b>	<b>8</b>
2.1 Task Description . . . . .	8
2.2 Mathematical Model . . . . .	8
2.3 Numerical Simulations . . . . .	9
Appendix . . . . .	14
<b>3 Task 28: Voter Model</b>	<b>18</b>
3.1 Task Description . . . . .	18
3.2 Mathematical Model . . . . .	18
3.3 Numerical Simulations . . . . .	19
<b>4 Task 46: EU transportation network II</b>	<b>22</b>
4.1 Task Description . . . . .	22
4.2 Data extraction . . . . .	22
Appendix . . . . .	25
<b>5 Bibliography</b>	<b>26</b>

# 1 | Task 01: Ising Model

---

## 1.1 | Task Description

---

The Ising model, initially examined in d-dimensional regular lattices, has been demonstrated to undergo phase transitions in certain types of complex networks as well. This task aims to review various analytical results that have been established [1] and to simulate the dynamics of the Ising model on networks.

## 1.2 | Mathematical Model

---

The ferromagnetic Ising model on a network of  $N$  nodes and adjacency matrix  $A$  is described in its most general form by the Hamiltonian:

$$\mathcal{H}(\mathbf{s}) = - \sum_{i < j} J_{i,j} s_i \cdot s_j - \sum_i h_i s_i$$

where  $\mathbf{s} = (s_1, \dots, s_N) \in \{\pm 1\}^N$  is the spin configuration of the nodes, the couplings  $J_{i,j}$  describe the pairwise spin interactions and  $h_i$  is a site-dependent external field. In the following, it will always be assumed  $J_{i,j} \equiv 1 \cdot A_{i,j}$  (only nearest neighbours interactions, of homogeneous strength) and  $h_i \equiv h$  (homogeneous external field), so that the Hamiltonian simplifies to:

$$\mathcal{H}(\mathbf{s}) = - \sum_{i < j} A_{i,j} s_i \cdot s_j - h \cdot \sum_i s_i. \quad (1.1)$$

The network is surrounded by a heat bath at temperature  $T = \frac{1}{\beta}$  ( $k_B \equiv 1$ ). The partition function  $Z$  and the free energy  $F$  are given by:

$$Z(T, \{A_{i,j}\}, h, N) = \sum_{\mathbf{s}} e^{-\beta \mathcal{H}(\mathbf{s})} \quad F = -\frac{1}{\beta} \ln[Z]$$

The most relevant thermodynamic quantities are the internal energy  $E$ , the magnetization  $M$ , the specific heat  $C$  and the magnetic susceptibility  $\chi$ :

$$\begin{aligned} E &= \mathbb{E}[\mathcal{H}(\mathbf{s})], & C &= \left( \frac{\partial E}{\partial T} \right)_h \equiv \frac{\mathbb{E}[E^2] - \mathbb{E}[E]^2}{T^2} \\ M &= \mathbb{E} \left[ \sum_i s_i \right] \equiv \left( \frac{\partial F}{\partial h} \right)_\beta, & \chi &= \left( \frac{\partial M}{\partial h} \right)_T = - \left( \frac{\partial^2 F}{\partial^2 h} \right)_T \equiv \frac{\mathbb{E}[M^2] - \mathbb{E}[M]^2}{T} \end{aligned}$$

A mean field approximation predicts that the existence of a disordered phase depends on the moments  $\langle k^n \rangle$  of the network degree distribution. Specifically, a homogeneous mean field approximation yields the critical temperature

$$T_C^{\text{hom. MF}} = \langle k \rangle. \quad (1.2)$$

[Appendix 1.3] contains the explicit calculation of this formula. According to mean field approximation, the critical temperature is expected to increase with the average degree, which indicates that adding more connections between the nodes helps in maintaining long-range order. The more precise heterogeneous (or degree-based) mean field approximation yields instead:

$$T_C^{\text{het. MF}} = \frac{\langle k^2 \rangle}{\langle k \rangle}. \quad (1.3)$$

In particular, the latter formula is derived under the assumption that the network is uncorrelated, i.e. the nearest neighbour degree distribution is supposed to be that of the configuration model:  $P_{n.n.}(k) = \frac{k \cdot P(k)}{\langle k \rangle}$ . A refined estimation of the critical temperature on uncorrelated networks is found with the replica approach (see [2]):

$$T_C^{\text{replica}} = \left[ -\frac{1}{2} \ln \left[ 2 - \frac{\langle k \rangle}{\langle k^2 \rangle} \right] \right]^{-1} \quad (1.4)$$

The order parameter of the transition is the average magnetization per site,  $s$ :

$$s := \frac{M}{N} = \mathbb{E} \left[ \frac{\sum_i s_i}{N} \right]$$

which is zero in the disordered phase ( $T > T_C$ ), non-zero in the ferromagnetic phase ( $T < T_C$ ) and monotonically decreasing with temperature. Also, the energy  $E(T)$  is expected to have an inflection point at  $T = T_C$ , and the response functions  $C$  and  $\chi$  are both supposed to peak at  $T = T_C$ .

### 1.3 | Numerical Simulations

For simulations, I chose to focus on scale free networks  $P(k) \sim k^{-\gamma}$ , in particular the Barabasi - Albert (BA) network of parameters  $N, m$  where  $N$  is the number of nodes and  $m$  is the number of links attached for each new node. This choice was motivated by the fact that article [3] could be used for comparison. The degree distribution for such a network is asymptotically given by  $P(k) \sim k^{-3}$ . Since one can only deal with finite size networks, finite size effects must be taken into account in the formulas [1.2, 1.3, 1.4]. The finite size of the network implies the existence of a cutoff degree  $k_{\max}(N)$ , so the right estimation of the moment  $\langle k^n \rangle$  is given by

$$\langle k^n \rangle = \sum_{k=m}^{k_{\max}(N)} k^n P(k),$$

where the cutoff  $k_{\max}(N)$  is defined such that the probability of having a node of degree  $k > k$  in a network of size  $N$  is less than one. With an elementary calculation one can verify that  $k_{\max}(N) = m \cdot N^{\frac{1}{\gamma-1}}$ , which reduces to  $k_{\max} = m \cdot \sqrt{N}$  for a BA network. The average degree is left unchanged by the finite size correction:  $\langle k \rangle = 2m$ , whereas the second moment changes to  $\langle k^2 \rangle \simeq m^2 \ln(N)$  [2]. Hence, the critical temperatures given by the mean field formulas are:

$$T_C^{\text{hom. MF}} = 2m, \quad T_C^{\text{het. MF}} \simeq \frac{m}{2} \ln(N) \quad (1.5)$$

As mentioned earlier, these formulas are derived for uncorrelated networks, which BA networks are not. Nevertheless, the authors of [3] verified through simulations that BA networks also align well with these formulas. The heterogeneous mean field approximation predicts

a logarithmic increase of the temperature with the network size, which means that in the thermodynamic limit ( $N \rightarrow +\infty$ ) the network is expected to be ferromagnetic at all temperatures. The homogeneous mean field is overly simplistic and fails to predict this behaviour.

The algorithm I implemented to simulate the Ising dynamic is the Metropolis-Hastings. As a preliminary check, I first used my code to simulate the Ising model on a regular  $2d$  lattice of  $N = 400$  nodes [Appendix, figures: 1.4 and 1.5]. and then on BA networks of the same size. Computational time for the simulations on BA was much higher: for comparison, obtaining the scaling of thermodynamic quantities with temperature [Appendix, figure: 1.5] took 12 minutes, whereas obtaining the same data for BA [Figure: 1.2] took on average 1h). Handling the adjacency matrix of the network in sparse format resulted in a significant performance improvement, but still the computations were long.

For estimating the critical temperature, I initially planned to derive an estimate from each of the four thermodynamic quantities ( $E, \langle s \rangle, C, \chi$ ) and then average them. However, I found that the peak temperature of the specific heat  $C$  was systematically lower than the peak temperature of the magnetic susceptibility  $\chi$  [see Figure: 1.2]. Among the two peaks, the latter was closer to the theoretical temperature value. This discrepancy was not observed in preliminary simulations on a  $2d$  regular lattice, indicating that it is not an artifact of my code. In fact, this same behavior was also observed by my colleague D. Wellingut [4]. Therefore, I averaged the temperatures obtained from the magnetization and the magnetic susceptibility, disregarding the other two. The final results of my simulations are shown in [Figure: 1.1]. The critical temperature estimated from the simulations is systematically lower than the theoretical predictions, but at least the dependence on  $m$  is similar.

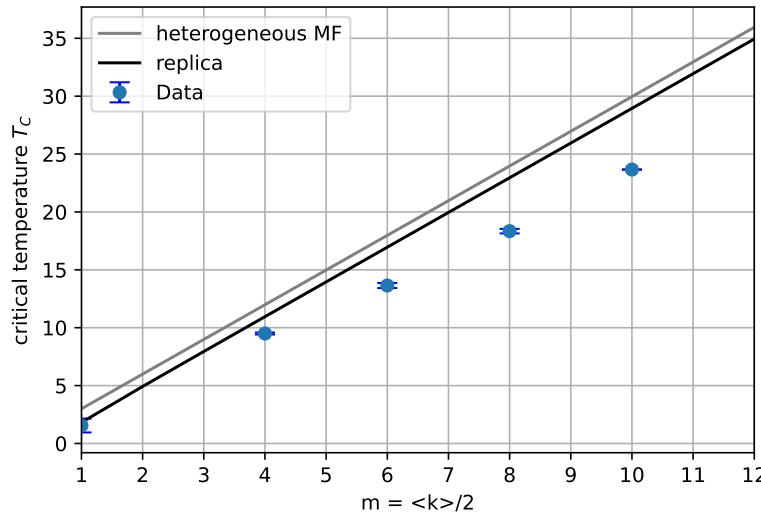


Figure 1.1: Scaling of the critical temperature with the minimum degree, for BA networks of 400 nodes. The data refers to the critical temperature extrapolated from the magnetization and the magnetic susceptibility peak.

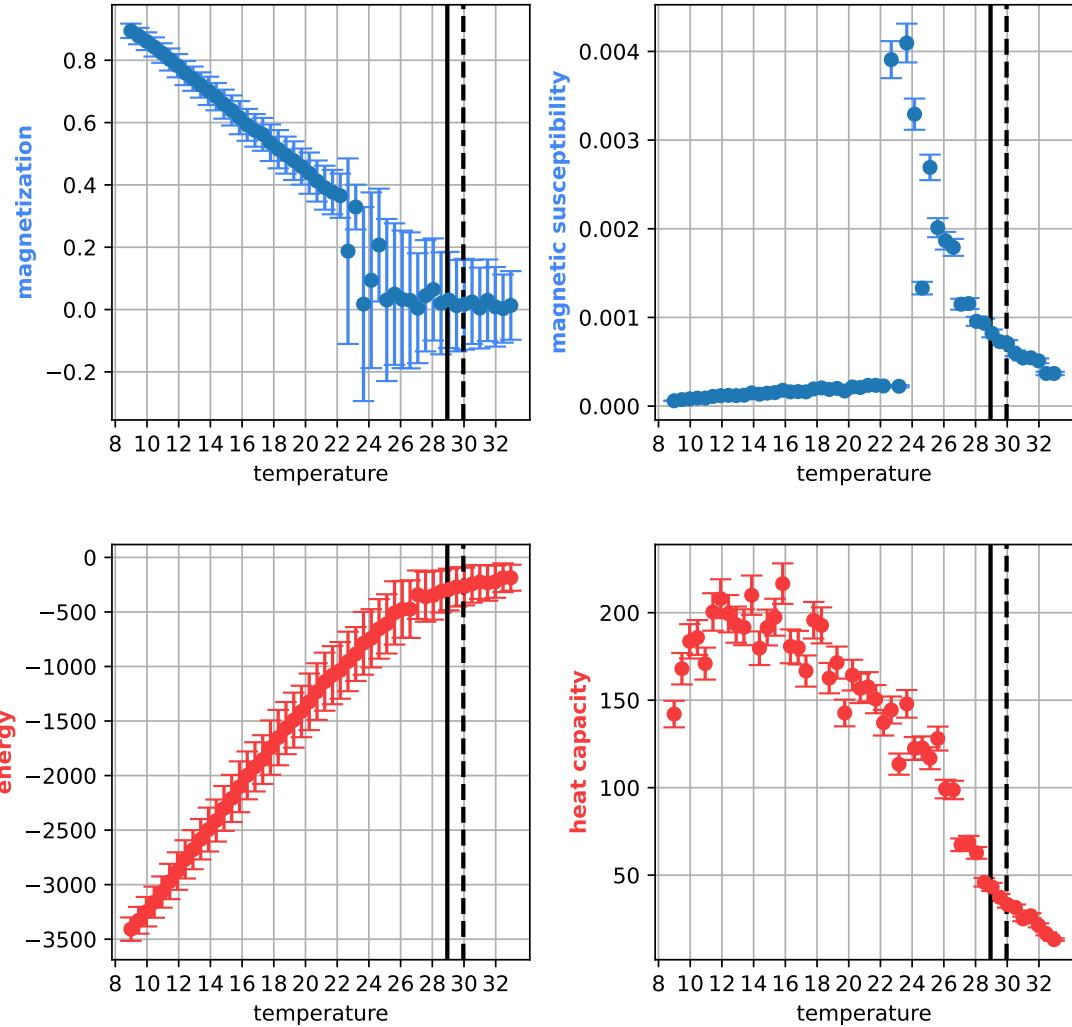


Figure 1.2: Scaling of thermodynamic quantities with temperature. BA network with  $N = 400$  and  $m = 10$ . Metropolis algorithm has been run with 1500 equilibration steps and 700 sweep steps. The dashed line is the heterogeneous mean field temperature 1.3, while the solid line is the replica temperature 1.4. The peaks of the two response functions do not coincide.

## Appendix

### Homogeneous mean-field calculation for $T_C$

The mean field approximation consists in neglecting the pairwise spin correlations  $\text{corr}(s_i, s_j) = (s_i - \langle s_i \rangle) \cdot (s_j - \langle s_j \rangle) \simeq 0$ . Moreover, the homogeneous mean field supposes that the average magnetization on each node is the same for all nodes  $\langle s_i \rangle \equiv \langle s \rangle$ . We start from the Ising Hamiltonian  $H\{\mathbf{s}\} = -\frac{1}{2} \sum_{i,j} A_{i,j} s_i \cdot s_j - h \cdot \sum_i s_i$  and write the identity:

$$\begin{aligned} s_i \cdot s_j &= [s_i - \langle s \rangle + \langle s \rangle] \cdot [s_j - \langle s \rangle + \langle s \rangle] \\ &= (s_i - \langle s \rangle) \cdot (s_j - \langle s \rangle) + (s_i + s_j) \langle s \rangle - \langle s \rangle^2 \simeq (s_i + s_j) \langle s \rangle - \langle s \rangle^2 \end{aligned}$$

where the latter is obtained disregarding the correlation term. With this substitution, the Hamiltonian becomes

$$H \simeq \frac{1}{2} \langle s \rangle^2 N \langle k \rangle - \sum_i s_i \cdot (\langle k \rangle \langle s \rangle + h) =: H^{MF}(\mathbf{s})$$

This expression can be now used to compute the partition function  $Z$  and, consequently, the free energy per site  $f = F/N$ :

$$\begin{aligned} Z^{MF} &= \sum_{\mathbf{s}} e^{-\beta H^{MF}(\mathbf{s})} = e^{-\beta \frac{N \langle k \rangle \langle s \rangle^2}{2}} \sum_{\mathbf{s}} e^{\beta [(\langle k \rangle \langle s \rangle + h) \sum_l s_l]} \\ &= e^{-\beta \frac{N \langle k \rangle \langle s \rangle^2}{2}} \prod_{i=1}^N \left[ \sum_{s=\pm 1} e^{\beta (\langle k \rangle + h) s_i} \right] \\ &= e^{-\beta \frac{N \langle k \rangle \langle s \rangle^2}{2}} [2 \cosh [\beta (\langle s \rangle \langle k \rangle + h)]]^N \\ f^{MF} &= \frac{F^{MF}}{N} = -\frac{1}{N \beta} \ln Z^{MF} = \frac{1}{2} \langle k \rangle \langle s \rangle^2 - \frac{1}{\beta} \ln [2 \cosh [\beta (\langle s \rangle \langle k \rangle + h)]] \end{aligned}$$

The average magnetization per site is then given by

$$\langle s \rangle = - \left( \frac{\partial f^{MF}}{\partial h} \right)_{\beta} = (\dots) = \tanh[\beta (\langle s \rangle \langle k \rangle + h)]$$

When the external field is off,  $\langle s \rangle$  solves the self consistent equation  $\langle s \rangle = \tanh[\beta \langle s \rangle \langle k \rangle]$ , which is the intersection of  $y = \langle s \rangle$  and  $y = \tanh[\beta \langle s \rangle \langle k \rangle]$ . For  $\beta < \beta_C = \frac{1}{\langle k \rangle}$ , there are no intersection points other than  $\langle s \rangle = 0$  ( $\Rightarrow$  the system is paramagnetic) for  $\beta > \beta_C$  there are two symmetric non-zero intersections ( $\Rightarrow$  the system is ferromagnetic). The critical temperature is thus given by  $T_C = \frac{1}{\beta_C} = \langle k \rangle$ , which is exactly [Eq. 1.2].

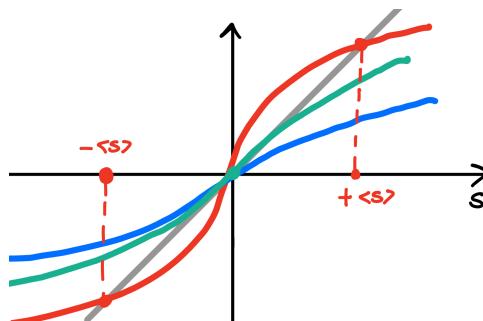


Figure 1.3: Grey line is  $y = s$ , coloured lines are  $y = \tanh(\beta \langle k \rangle s)$ , respectively for  $\beta \langle k \rangle > 1$  (red, ferromagnetic state),  $\beta \langle k \rangle = 1$  (green, critical point) and  $\beta \langle k \rangle < 1$  (blue, paramagnetic state).

## Preliminary Montecarlo simulation on a 2D lattice

An infinite regular 2– dimensional spin lattice with Hamiltonian given by [Eq: 1.1] exhibits a second order phase transition at the critical temperature

$$T_C = \frac{2}{\log 1 + \sqrt{2}} \simeq 2.26 \quad (1.6)$$

For my simulation, I considered a lattice of  $N = 20 \cdot 20 = 400$  nodes. I first simulated the dynamics at fixed temperature, to get an estimate of the number of steps required for the system to equilibrate. I found that  $\simeq 200$  steps were sufficient for a lattice of this size [Fig 1.4]. Then I simulated Ising over a broad range of temperatures around the expected critical temperature and computed the energy, magnetization, heat capacity and magnetic susceptibility [Fig: 1.5].

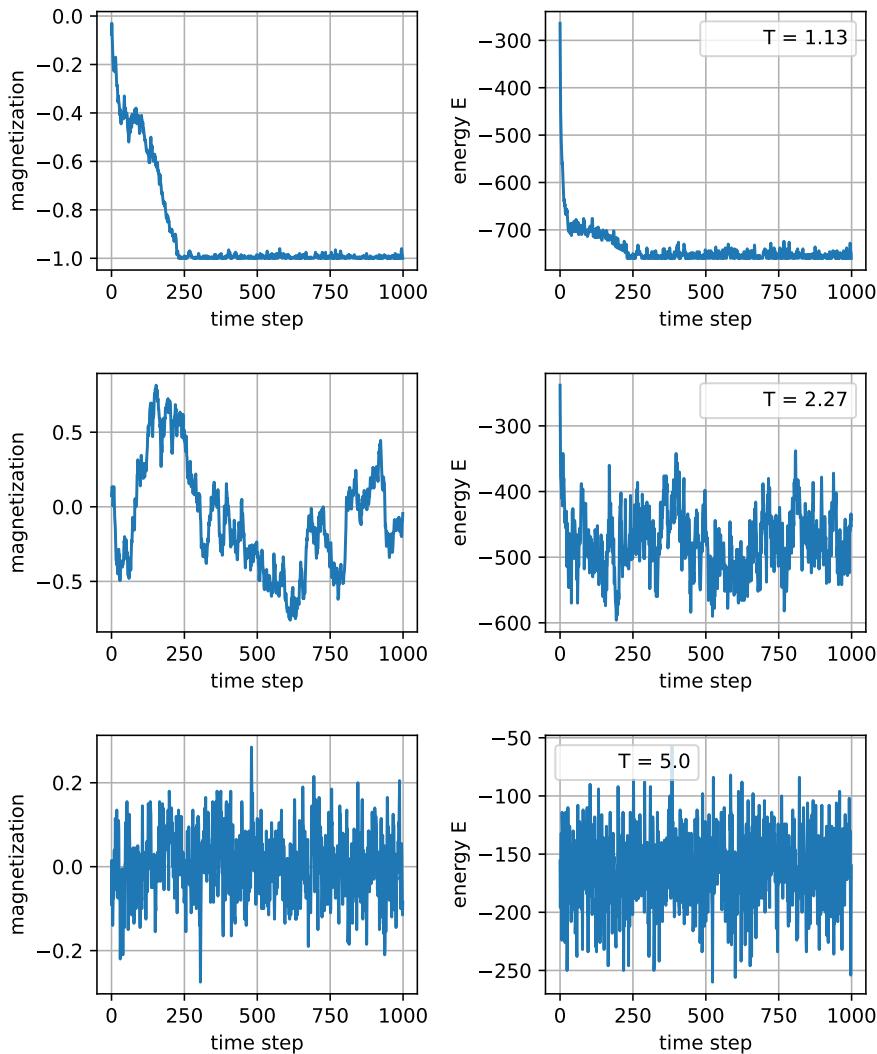


Figure 1.4: Time evolution of the magnetization and energy, respectively below, at and above the critical temperature [Eq: 1.6].

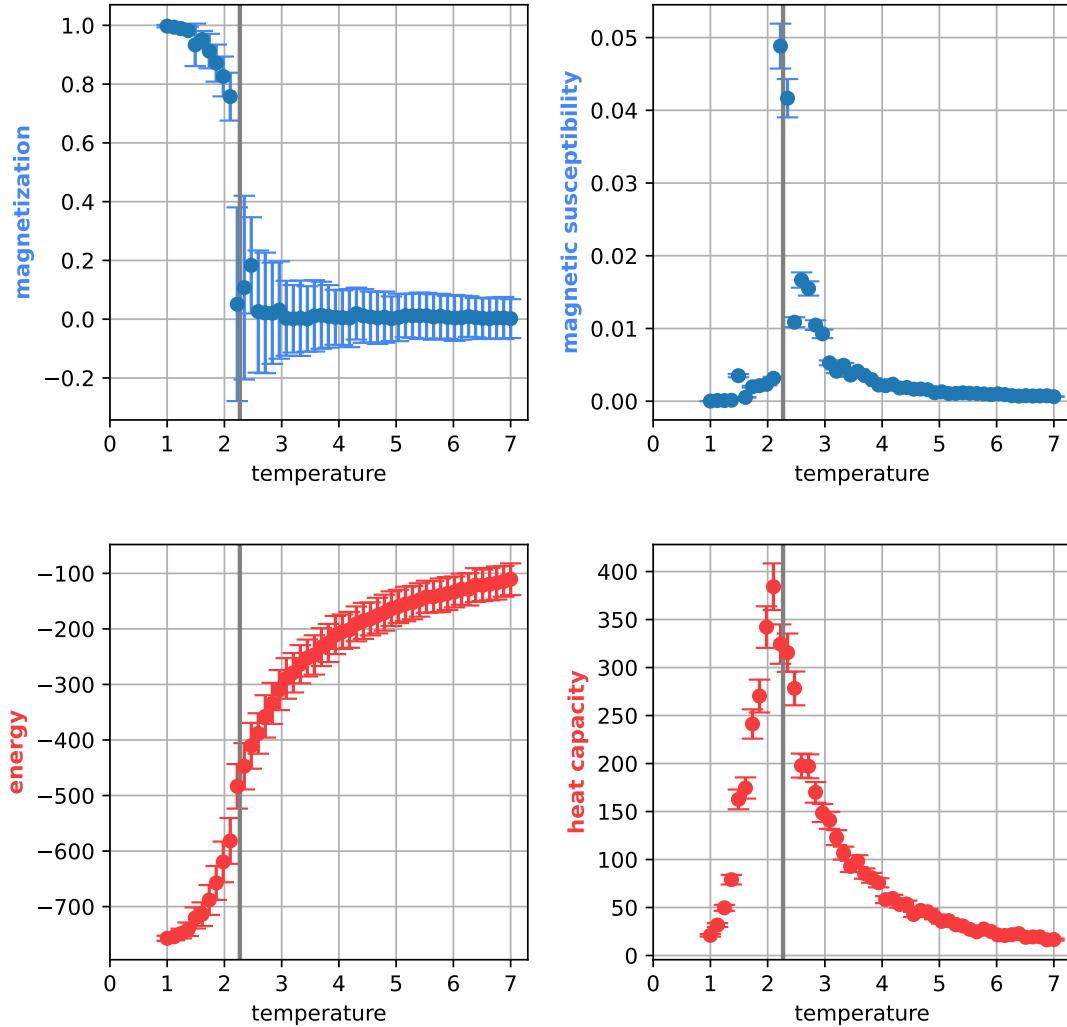


Figure 1.5: Dependence of thermodynamic quantities on temperature, for a 2d regular square lattice of 400 nodes. The vertical grey line marks the theoretical critical temperature [Eq: 1.6]. One can see that the peaks of  $C$  and  $\chi$  are very close to the theoretical temperature. With 500 equilibration steps and 500 sweep steps at each temperature, the code took 12 minutes to execute.

## 2 | Task 18: Turing Patterns

---

### 2.1 | Task Description

---

The goal of this task is to simulate a Turing activator-inhibitor dynamics on networks, possibly replicating some of the results reported in [5].

### 2.2 | Mathematical Model

---

An activator-inhibitor system on a network can be defined in very close analogy to what one does on a continuous medium. The equations for the system are of the reaction-diffusion type:

$$\begin{cases} \dot{u}_i(t) = f(u_i v_i) - \epsilon [L u(t)]_i \\ \dot{v}_i(t) = g(u_i v_i) - \epsilon \sigma [L v(t)]_i \end{cases}$$

Here,  $\mathbf{u} = (u_1, \dots, u_N)$  and  $\mathbf{v} = (v_1, \dots, v_N)$  are, respectively, the concentrations of the activator and the inhibitor substance on the  $N$  nodes of the network. The reactions take place on each node and are encoded in the reaction terms  $f$  and  $g$ .  $L$  is the laplacian matrix, defined as  $L = D - A$  where  $D_{i,j} = k_i \cdot \delta_{i,j}$  and  $A$  is the adjacency matrix. The diffusivity of the activator substance is  $\epsilon$ , while that of the inhibitor is  $\epsilon \sigma$ , so that  $\sigma$  is the ratio between them. The reaction terms are supposed to satisfy two basic requirements:

1. There exists a homogeneous equilibrium  $(\bar{u}, \bar{v})$  which is linearly stable in the absence of diffusion (indeed, the key concept of the Turing model is that the instability is driven by diffusion). In mathematical terms [Appendix: 2.3]:

$$\begin{pmatrix} f(\bar{u}, \bar{v}) \\ g(\bar{u}, \bar{v}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \begin{cases} \text{tr}(J) = f_u + g_v < 0 \\ \det(J) = f_u \cdot g_v - f_v \cdot g_u > 0 \end{cases}, \quad J(\bar{u}, \bar{v}) := \begin{pmatrix} f_u & f_v \\ g_u & g_v \end{pmatrix} \quad (2.1)$$

2. Correct qualitative behaviour of reactions in the neighborhood of the fixed point  $(\bar{u}, \bar{v})$ : the activator  $u$  is supposed to enhance its own production and the production of the inhibitor  $v$ . Viceversa, the inhibitor  $v$  is supposed to suppress the production of both the activator  $u$  and itself. The functions  $f, g$  need to reflect this behaviour, at least in a neighbourhood of the equilibrium  $(\bar{u}, \bar{v})$ . In mathematical terms:



The conditions for pattern emergence are found through linear stability analysis of the uniform stationary state  $(\bar{u}, \bar{v})$  with respect to the perturbation. The perturbation is expanded

in the orthonormal basis of the laplacian eigenvectors  $\{\Phi^{(n)}\}_{n=1}^N$ :

$$\begin{pmatrix} \delta u_i(t) \\ \delta v_i(t) \end{pmatrix} = \begin{pmatrix} 1 \\ B_n \end{pmatrix} \cdot \sum_{n=1}^N c_n \Phi_i^{(n)} e^{\lambda_n t}$$

where coefficients  $c_n$  are determined by the initial conditions. The growth rate  $\lambda_n$  of the  $n$ -mode is found through the resolution of an eigenvalue problem that depends on  $(\Lambda_n, \sigma, \epsilon)$ , where  $\Lambda_n$  is the laplacian eigenvalue associated with eigenvector  $\Phi^{(n)}$ . In the following, the laplacian eigenvalues are supposed to be ordered as  $0 = \Lambda_1 \leq \Lambda_2 \leq (\dots) \leq \Lambda_N$ . The function  $\text{Re}\{\lambda_n\}(\Lambda_n|\epsilon, \sigma)$  is usually referred to as the *dispersion relation*. Whether the  $n$ -mode amplitude grows with time or decays depends on the sign of  $\text{Re}\{\lambda_n\}$ . One finds that the dispersion curve  $\text{Re}\{\lambda\}(\Lambda|\epsilon, \sigma)$  lies entirely below the x axis for  $\sigma < \sigma_C$  ( $\Rightarrow$  no instability), while assumes positive values in some range of positive  $\Lambda$ 's for all  $\sigma > \sigma_C$  ( $\Rightarrow$  instability). This critical threshold  $\sigma_C$  is a function of the jacobian  $J$  [Eq: 2.1] alone:

$$\sigma > \sigma_c := \frac{(f_u g_v - 2 f_v g_u) + 2 \sqrt{f_v g_u (f_v g_u - f_u g_v)}}{f_u^2} \quad (2.3)$$

and always satisfies  $\sigma_C > 1$ , i.e. the diffusivity of the inhibitor substance is higher than that of the activator (this is in fact the key reason why the diffusion process can induce patterns: see [Appendix: 2.7] for discussion).

For  $\sigma$  slightly above the critical threshold, the unstable mode(s) will correspond to eigenvalues  $\Lambda_n$  located closely around the *critical eigenvalue*

$$\Lambda_c(\epsilon) := \frac{1}{\epsilon} \sqrt{\frac{\det[J]}{\sigma_c}} \quad (2.4)$$

The full derivation of the above equations [Eq: 2.4, 2.3] is presented in [Appendix: 2.3]. The subsequent evolution of the pattern is non-linear and eventually results in a steady state (that can be either stationary or time dependent).

## 2.3 | Numerical Simulations

As in [5], the reaction terms are set as

$$\begin{cases} f(u, v) = [\frac{a+b u - u^2}{c} - v] u \\ g(u, v) = [u - (1 + d v)] \cdot v \end{cases} \quad (2.5)$$

with parameters  $a = 35$ ,  $b = 16$ ,  $c = 9$ ,  $d = 2/5$ , obtaining a linearly stable fixed point  $(\bar{u}, \bar{v}) = (5, 10)$  and a critical diffusion ratio  $\sigma_c \simeq 15.5$ .

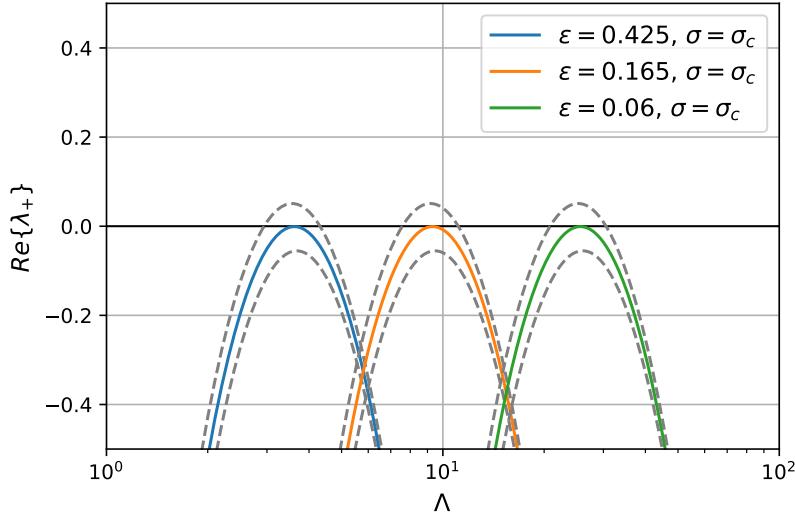


Figure 2.1: Dispersion relation for the chosen reaction kinetics [Eq. 2.5] at different values of the activator diffusivity  $\epsilon$ . The dashed grey lines are the dispersion curves slightly above and below the critical diffusivity ratio  $\sigma_c$ . The critical eigenvalue  $\Lambda_c$  [Eq. 2.4] is inversely proportional to the activator diffusivity  $\epsilon$ .

Following the choice of the authors, I simulated the dynamics on Barabasi-Albert scale free networks. The parameters I chose are  $N = 200$  for the number of nodes,  $\langle k \rangle = 10$  for the mean degree,  $\epsilon = 0.12$  for the activator diffusivity and  $\sigma = 15.6 > \sigma_c = 15.5$  for the diffusivity ratio. The reason for choosing a diffusivity ratio that is only slightly above the critical threshold is to keep the number of allowed modes low.

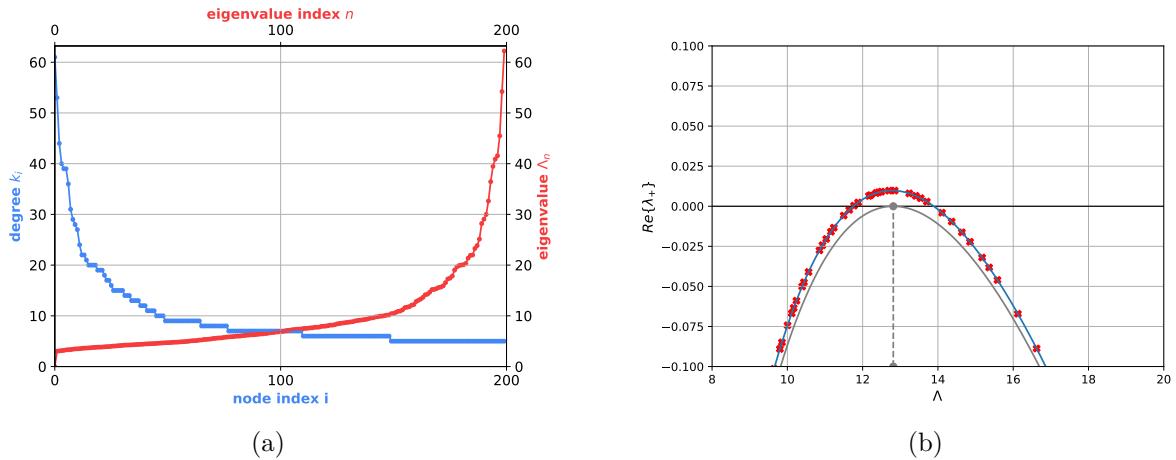


Figure 2.2: Left subfigure (a) shows the degree spectrum and the laplacian eigenvalue spectrum of the network used in the simulation. Network nodes are indexed by decreasing degree. Right subfigure (b) shows the dispersion relation curve. For the chosen simulation parameters, the critical eigenvalue is  $\Lambda_c \approx 12.8141$ . The closest assumed eigenvalue in the network happens to be  $\Lambda_n \approx 12.64$ , of index  $n = 157$ . However, as one can see from the graph, the network eigenvalue spectrum comprehends several other allowed modes (red crosses) in the unstable range, and they are expected to contribute to pattern initiation as well.

The initial homogeneous state  $(\bar{u}, \bar{v}) = (5, 10)$  was perturbed with  $(\delta u_i, \delta v_i)$  sampled from a uniform distribution within the range  $[-0.5, +0.5]$ . The evolution of the activator concentration on nodes at different times is shown in [Figure 2.3]. Also, the GitHub repository [6] contains a .mp4 video of the whole evolution.

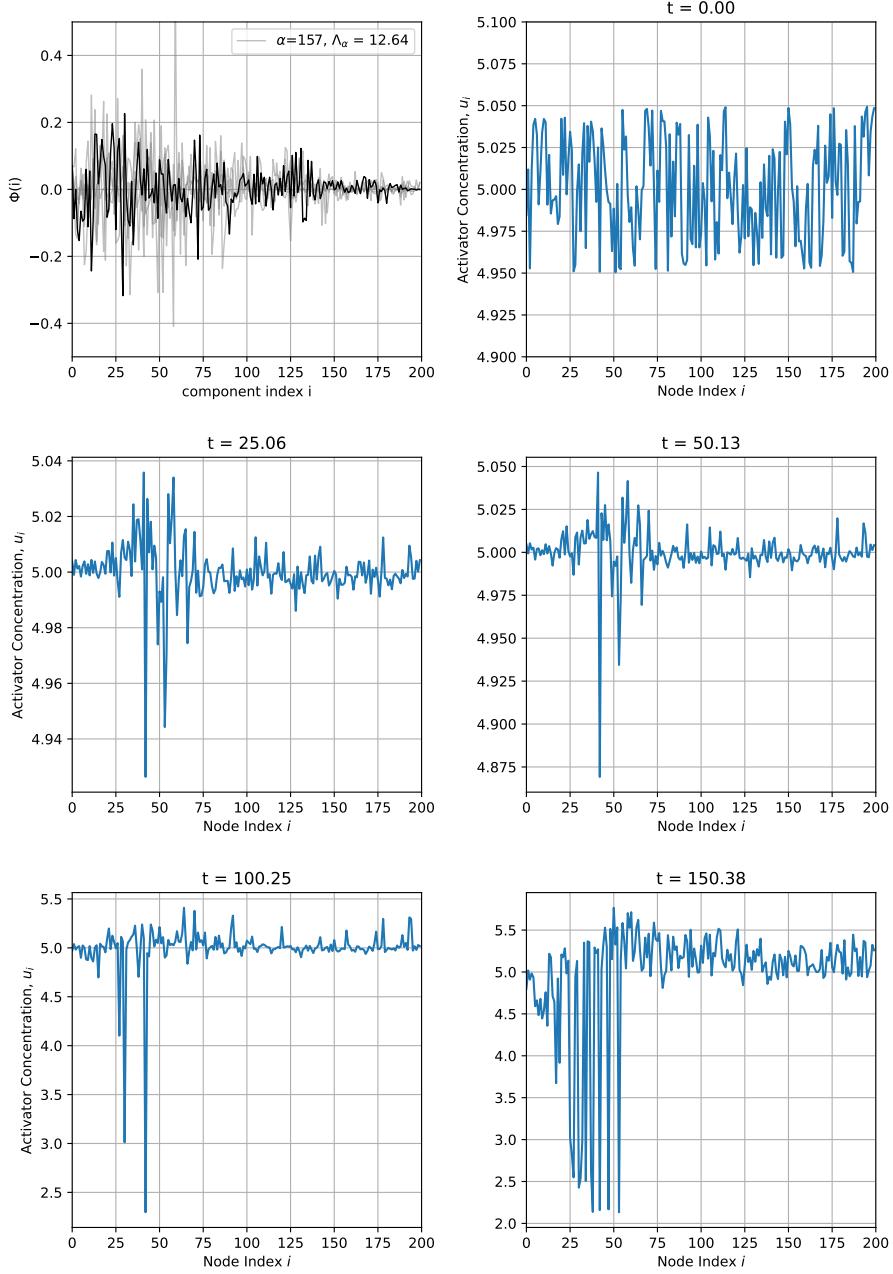


Figure 2.3: Time evolution of the activator concentration. The subfigure at top left corner shows the components of the critical eigenvector  $n = 157$  (black) and its closest neighbours  $n = 156, 158, 159$  in the unstable range (grey).

In the early stage, the pattern is expected to grow proportionally to the critical eigenvector, which is the mode of largest growing rate:  $\delta u(t), \delta v(t) \propto \Phi^{(n)}$ . In fact, when the initial uniform perturbation dies out and the pattern starts to grow, the activator substance distribution is located similarly to the critical eigenvector (see snapshot  $t = 25.06$ ). Later on, the evolution becomes strongly non-linear and the pattern is reshaped (see snapshot at  $t = 100.25$ ), till it settles into a stationary state (see snapshot at  $t = 150.38$ ) where nodes are split into one activator-rich group and one activator-poor group. The authors found that this peculiar behaviour is well described in the framework of a mean field theory [5].

One thing that can be noticed in the initial stage pattern [Figure 2.3, snapshot at  $t = 25.06$ ] is that the significant variations of the activator level are localized on a subset of nodes of close degree ( $k \sim 25 - 75$ ). That is, only a specific subset of the network undergoes differentiation. This effect was not a fortuity but is due to the fact that the laplacian eigenvectors in a large graph with a broad degree distribution tend to be localized around a specific degree  $\bar{k}(\Lambda)$ . Also, as authors report, a simple relation  $\bar{k}(\Lambda) \simeq \Lambda$  seems to hold. This property was indeed verified for the BA network that I used in the simulation [Figure: 2.4, 2.5].

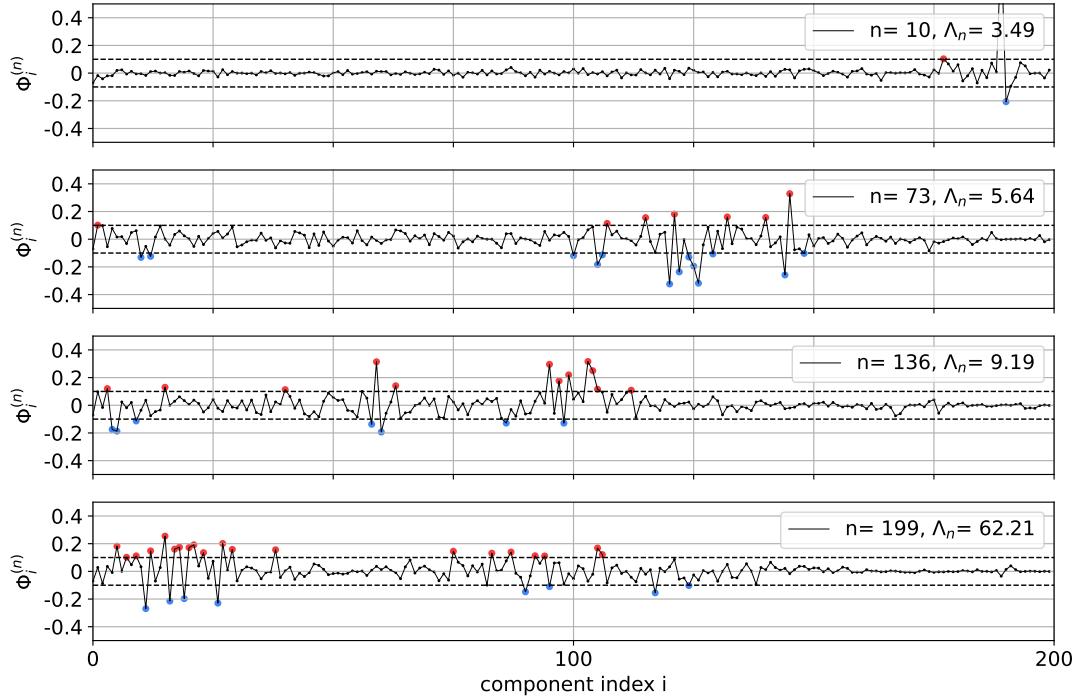


Figure 2.4: Localization of laplacian eigenvectors in a BA with  $N = 200$  and  $\langle k \rangle = 10$ . Nodes are indexed by decreasing degree (i.e. node 1 has the maximum degree, and so on). One can see that  $\bar{k}(\Lambda)$  grows with  $\Lambda$ .

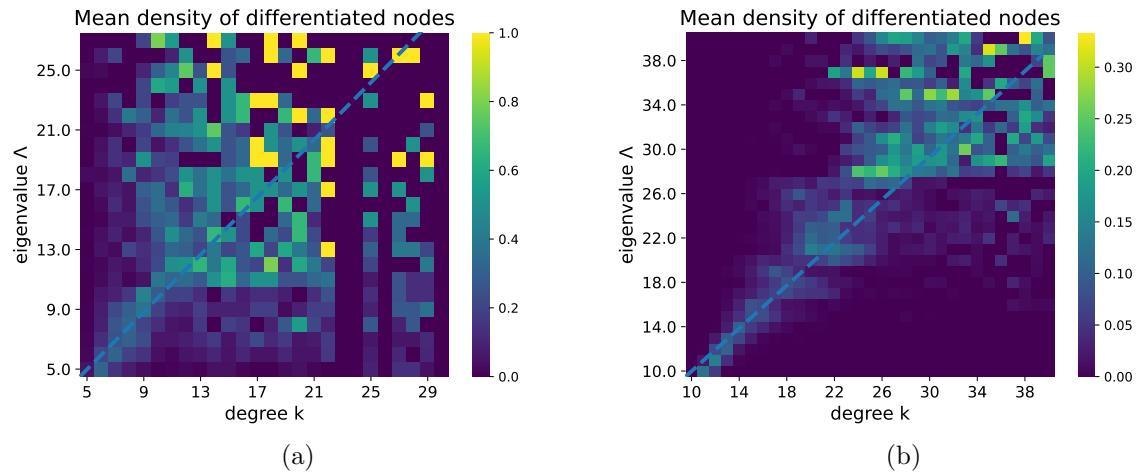


Figure 2.5: Localization of laplacian eigenvectors in a BA graph with  $N = 200$ ,  $\langle k \rangle = 10$  (subfigure a) and  $N = 1000$ ,  $\langle k \rangle = 20$  (subfigure b). Eigenvalues have been grouped into bins of unitary width. The population of nodes inside each degree group,  $N_k$ , was calculated. The heatmap represents the mean density of differentiated nodes for each degree group  $z = N_k^{\text{diff}}(\Lambda)/N_k$ . A node  $i$  of degree  $k$  is considered differentiated with respect to eigenvalue  $\Lambda_n$  if the eigenvector component  $\Phi_i^{(n)}$  satisfies  $|\Phi_i^{(n)}| > 0.1$ . I found, like authors [5] report, that approximately  $\bar{k}(\Lambda) \propto \Lambda$  (dashed line) and that the localization effect is more pronounced in the larger graph (subfigure b).

## Appendix

### Linear Stability of a 2x2 autonomous system

Say  $(\bar{x}, \bar{y}) \in \mathbb{R}^2$  is a fixed point (or equilibrium) of the autonomous system

$$\begin{pmatrix} \dot{x}(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} f[x(t), y(t)] \\ g[x(t), y(t)] \end{pmatrix}$$

which, namely, means that  $f(\bar{x}, \bar{y}) = g(\bar{x}, \bar{y}) = 0$ . Now we want to study how the system evolves if a small perturbation  $(\delta x, \delta y)$  is added to the equilibrium. We can linearize the system:

$$\begin{pmatrix} \delta \dot{x}(t) \\ \delta \dot{y}(t) \end{pmatrix} = \begin{pmatrix} \frac{\delta f}{\delta x}|_{(\bar{x}, \bar{y})} & \frac{\delta f}{\delta y}|_{(\bar{x}, \bar{y})} \\ \frac{\delta g}{\delta x}|_{(\bar{x}, \bar{y})} & \frac{\delta g}{\delta y}|_{(\bar{x}, \bar{y})} \end{pmatrix} \cdot \begin{pmatrix} \delta x(t) \\ \delta y(t) \end{pmatrix} + o(||(\delta x, \delta y)||)$$

The latter is a homogeneous linear system with general solution:

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \alpha_1 \mathbf{v}_1 e^{\lambda_1 t} + \alpha_2 \mathbf{v}_2 e^{\lambda_2 t}$$

where  $\lambda_1, \lambda_2$  are the eigenvalues of the jacobian and  $\mathbf{v}_1, \mathbf{v}_2$  are the corresponding eigenvectors, and  $\alpha_1, \alpha_2$  are coefficients determined by the initial condition. The eigenvalues of a  $2 \times 2$  matrix can either be both real or a pair of complex conjugates ( $\lambda_1 = \lambda, \lambda_2 = \bar{\lambda}$ ). Whether the amplitude of the perturbation dies out exponentially or explodes depends on the sign of  $\operatorname{Re}\{\lambda\}$ . The linear stability requirement is

$$\operatorname{Re}\{\lambda\} < 0 \quad \text{for both } \lambda \text{'s}$$

The eigenvalues  $\lambda_1, \lambda_2$  are the roots of the characteristic polynomial of the jacobian  $J$ :

$$p_J(\lambda) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) = \lambda^2 - \operatorname{tr}[J] \cdot \lambda + \det[J]$$

Then we have the following cases, depending on the discriminant  $\Delta^2 = \operatorname{tr}[J]^2 - 4 \det[J]$ :

$$\begin{cases} \Delta^2 > 0 : & \lambda_1, \lambda_2 = \frac{\operatorname{tr}[J] \pm \Delta}{2} \\ \Delta^2 = 0 : & \lambda_1 = \lambda_2 = \frac{\operatorname{tr}[J]}{2} \in \mathbb{R} \\ \Delta^2 < 0 : & \lambda_1, \lambda_2 = \alpha \pm i\beta \in \mathbb{C} \end{cases}$$

One can easily see from the polynomial  $p_J(\lambda)$  that in the case  $\Delta^2 \geq 0$ , the requirement that both roots are negative is satisfied  $\iff \operatorname{tr}[J] < 0, \det[J] > 0$ . Instead, in the case  $\Delta^2 < 0$ , it is obligatorily  $\det[J] = \lambda \cdot \bar{\lambda} = |\lambda|^2 > 0$ , and  $\operatorname{tr}[J] = 2 \cdot \operatorname{Re}(\lambda)$ . Then again the requirement holds  $\iff \operatorname{tr}[J] < 0$ .

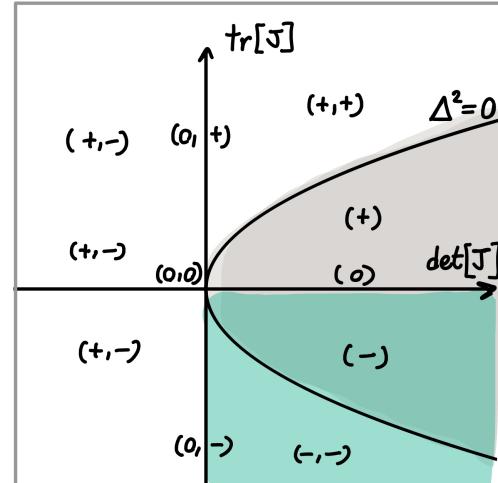


Figure 2.6: Bifurcation diagram for a  $2 \times 2$  homogeneous system.  $(\pm, \pm)$  are the signs of  $\operatorname{Re}\{\lambda\}$  in the various regions of the  $(\operatorname{tr}[J], \det[J])$  plane. Green marks the stability region and grey marks the region where eigenvalues are complex.

## An intuition of how diffusion can produce patterns

[Figure 2.7] can help understand how the activation - inhibition mechanism can induce the formation of heterogeneous patterns. Graph 1 shows the initial condition of the system. Suppose that the concentration of the activator is relatively higher than in other regions by random fluctuation. By the self-enhancing property of the activator, the concentration of activator increases at the center region (graph 2), followed by the increase of inhibitor at the neighboring region A. As the **diffusion rate of inhibitor is larger** than that of the activator, substantial amounts of inhibitor move toward the lateral regions. This depresses the activator function, resulting in the decrease of the activator concentration there (graph 3). Decrease of activator causes the decrease of inhibitor in the wider region (graph 4). At the region B, as inhibitor concentration is gotten lower, activator becomes relatively dominant than inhibitor. This situation is enough to start the local self activation at region B (graph 5).

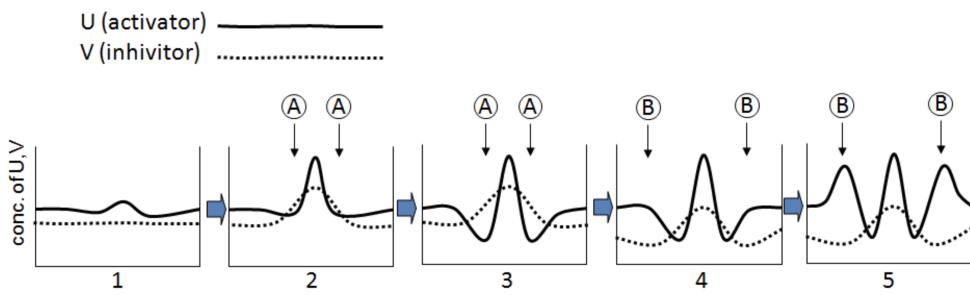


Figure 2.7: Picture taken from [7].

## Conditions for diffusion-driven instability

Diffusion driven instability is investigated by means of linear stability analysis. A small random perturbation  $\delta u_i, \delta v_i$  is added at each node, starting from the homogeneous equilibrium state. A linearized system of equations is obtained for the evolution of the perturbation:

$$\begin{cases} u_i(t) = \bar{u} + \delta u_i(t) \\ v_i(t) = \bar{v} + \delta v_i(t) \end{cases} \rightarrow \begin{cases} \delta \dot{u}_i(t) \simeq f_u \delta u_i(t) + f_v \delta v_i(t) - \epsilon [L \cdot (\bar{u} + \delta u(t))]_i \\ \delta \dot{v}_i(t) \simeq g_u \delta u_i(t) + g_v \delta v_i(t) - \epsilon \sigma [L(\bar{v} + \delta v(t))]_i \end{cases}$$

But  $L\bar{u} = L\bar{v} = 0$ , since the constant vectors  $u$  and  $v$  are eigenvectors of the laplacian with eigenvalue zero. In the case of a continuous medium, the perturbation is expanded as a Fourier series of plane waves. The rationale of this is that it makes the partial differential equations turn into an eigenvalue problem. In the network case, we instead write the perturbation as:

$$\begin{pmatrix} \delta u_i(t) \\ \delta v_i(t) \end{pmatrix} = \begin{pmatrix} 1 \\ B_n \end{pmatrix} \cdot \sum_{n=1}^N c_n \Phi_i^{(n)} e^{\lambda_n t}$$

Where  $\Phi^{(n)}$  is the  $n$ -th eigenvector of the laplacian matrix  $L$  and its corresponding eigenvalue is  $\Lambda_n$  ( $0 = \Lambda_1 \leq \Lambda_2 \leq \dots \leq \Lambda_N$ ). The coefficients  $\{c_n\}$  are determined by the initial conditions ( $t = 0$ ). By doing this, the system of equations is turned into a  $\Lambda$ -dependent eigenvalue problem. In fact, if we substitute the trial function

$$\begin{pmatrix} \delta u_i(t) \\ \delta v_i(t) \end{pmatrix} = \begin{pmatrix} 1 \\ B_n \end{pmatrix} \cdot c_n \Phi_i^{(n)} e^{\lambda_n t}$$

into the linearized system of equations, we get:

$$\begin{pmatrix} \delta \dot{u}_i \\ \delta \dot{v}_i \end{pmatrix} = \begin{pmatrix} f_u - \epsilon \Lambda_n & f_v \\ g_u & g_v - \epsilon \sigma \Lambda_n \end{pmatrix} \cdot \begin{pmatrix} \delta u_i \\ \delta v_i \end{pmatrix}$$

$$\rightarrow \lambda_n \cdot \begin{pmatrix} 1 \\ B_n \end{pmatrix} = \begin{pmatrix} f_u - \epsilon \Lambda_n & f_v \\ g_u & g_v - \epsilon \sigma \Lambda_n \end{pmatrix} \cdot \begin{pmatrix} 1 \\ B_n \end{pmatrix}$$

Or, in compact form:

$$\lambda_n \mathbf{v}_n = M(\Lambda_n, \sigma, \epsilon) \mathbf{v}_n \quad (2.6)$$

The differences between the network case and the continuous medium case end here. From this point onwards, the steps are exactly the same. The goal is to find the range of parameters  $(\sigma, \epsilon)$  that produce  $\Re\{\lambda_{\pm}(\Lambda)\} > 0$  for a positive range of  $\Lambda$ 's.

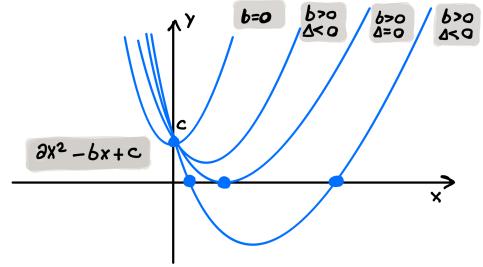
Necessarily, at least one of the following conditions must hold:

$$\begin{cases} \det[M(\Lambda, \sigma, \epsilon)] < 0 \\ \text{tr}[M(\Lambda, \sigma, \epsilon)] > 0 \end{cases}$$

But  $\text{tr}[M(\Lambda, \sigma, \epsilon)] = \text{tr}[J] - \epsilon \Lambda (1 + \sigma) < 0$  because the homogeneous state is a stable equilibrium, then the only possibility is that  $\det[M(\Lambda, \sigma, \epsilon)] < 0$ . The next steps involve some simple algebra calculations:

$$\det[M(\Lambda, \sigma, \epsilon)] = \epsilon^2 \sigma \Lambda^4 - \epsilon (g_v + \sigma f_u) \Lambda + \det[J] \stackrel{!}{\leq} 0 \quad \text{for some } \Lambda > 0$$

This is a parabola of kind  $y = ax^2 - bx + c$  with  $(a, b, c > 0)$ , so the minimum is reached at coordinates  $(x_{min}, y_{min}) = (\frac{b}{2a}, c - \frac{b^2}{4a})$ . The parabola touches the  $x$  axis ( $y_{min} \leq 0$ )  $\iff \Delta^2 = b^2 - 4ac \geq 0$ .



We require:

$$\begin{cases} x_{min}(\epsilon, \sigma) > 0 \\ y_{min}(\epsilon, \sigma) \leq 0 \end{cases} \quad \begin{cases} \epsilon(g_v + \sigma f_u) > 0 \\ [\epsilon(g_v + \sigma f_u)]^2 \geq 4\epsilon^2 \sigma \det[J] \end{cases}$$

From the first one, one notices that there exists a minimum value threshold for  $\sigma > \sigma_{min} = \frac{|g_v|}{f_u}$ . Additionally, since  $\text{tr}[J] = (-|g_v| + f_u) < 0$ , then  $\sigma_{min} > 1$ : a necessary condition for pattern initiation is that the inhibitor diffuses faster than the activator (but it is still not sufficient). Proceeding,

$$\begin{cases} \sigma > \sigma_{min} > 1 \\ f_u^2 \sigma^2 + 2(f_u g_v - 2 \det[J]) \sigma + g_v^2 \geq 0 \end{cases} \quad \begin{cases} \sigma > \sigma_{min} > 1 \\ \sigma < \sigma_- (< \sigma_{min}) \quad \text{or} \quad \sigma > \sigma_+ \end{cases}$$

where  $\sigma_{\pm}$  are the roots of the quadratic equation:

$$\sigma_{\pm} = \frac{(f_u g_v - 2 f_v g_u) \pm 2 \sqrt{f_v g_u (f_v g_u - f_u g_v)}}{f_u^2}.$$

The function  $y_{min}(\sigma)$  reaches its maximum for  $\sigma = \sigma_{min}$  and goes to  $-\infty$  for both  $\sigma \rightarrow 0^+$  and  $\sigma \rightarrow +\infty$ . Additionally, the lower-branch root  $\sigma_-$  is below the threshold value  $\sigma_{min}$ . Then, both our requirements are satisfied if and only if  $\sigma > \sigma_+$ . In summary, the necessary and sufficient condition for instability is

$$\sigma > \sigma_c := \frac{(f_u g_v - 2 f_v g_u) + 2 \sqrt{f_v g_u (f_v g_u - f_u g_v)}}{f_u^2}$$

which is the critical threshold in [Eq: 2.3].

### Critical eigenvalue and graph finite size effect

From the above equations, one also finds the critical eigenvalue  $\Lambda_C$  [Eq: 2.4] for given  $\epsilon$ :

$$\Lambda_c(\epsilon) \equiv x_{min}(\sigma_C, \epsilon) = \frac{g_v + \sigma_c f_u}{2\epsilon\sigma_c} = (\dots) = \frac{1}{\epsilon} \sqrt{\frac{\det[J]}{\sigma_c}}$$

When  $\sigma > \sigma_C$  a range of  $\Lambda$  values corresponding to unstable modes appears, centered around  $\Lambda_c$ . However, the eigenvalue spectrum of the laplacian matrix is not continuous: instability will be seen only if the network laplacian possesses at least one eigenvalue  $\Lambda_n$  that falls inside this range. In particular, the eigenvalue spectrum of the laplacian is bounded above for a graph of finite size: then, for  $\epsilon$  sufficiently low, the critical eigenvalue could lay beyond the largest laplacian eigenvalue,  $\Lambda_N$ . In this case, no patterns would be seen. This finite-size effect is analogous to that found in a continuous medium.

### Dispersion relation $\lambda(\Lambda; \sigma, \epsilon)$

Finally, the growing rate of the unstable mode,  $\lambda_n = \lambda(\Lambda_n)$  is calculated from the characteristic polynomial of  $M(\Lambda_n, \sigma, \epsilon)$ :

$$p(\lambda) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) = \lambda^2 - \text{tr}[M(\Lambda, \sigma, \epsilon)] \cdot \lambda + \det[M(\Lambda, \sigma, \epsilon)]$$

$$\begin{aligned} \lambda_{\pm} &= \frac{1}{2} \left[ \text{tr}[M] \pm \sqrt{\text{tr}[M]^2 - 4 \det[M]} \right] = \frac{1}{2} \left[ -|\text{tr}[M]| \pm \sqrt{|\text{tr}[M]|^2 - 4 \det[M]} \right] \\ &= \frac{1}{2} \left[ [f_u + g_v - (1 + \sigma)\epsilon\Lambda] \pm \sqrt{4f_v g_u + [f_u - g_v - (1 - \sigma)\epsilon\Lambda]^2} \right] \end{aligned}$$

In the instability region,  $\text{tr}[M] < 0$  and  $\det[M] < 0$ , then eigenvalues are real and distinct. Also, the lower branch is always negative so we are not interested in it.

# 3 | Task 28: Voter Model

---

## 3.1 | Task Description

---

The voter model is perhaps the simplest and most studied model of cooperative behaviour. While its behaviour on regular lattices of arbitrary dimension  $d$  is known in detail, it is much more difficult to get a comprehensive general picture of its behaviour on a complex network, where there is interplay between many factors, such as degree distribution, correlations, effective dimensionality and level of disorder[10]. The aim of this task is to describe the voter model and attempt to reproduce its behaviour on some class of networks. Specifically, I chose to focus on scale free networks, for which some analytical results have been found [11].

## 3.2 | Mathematical Model

---

In the voter model, each node of the network is in one of two possible states  $\sigma_i \in \{\pm 1\}$  (say, for instance, spin up or down). The evolution starts with some random configuration of up and down spins. The update rule is defined as follows:

- i) choose one node at random,
  - ii) choose one of its neighbors at random,
  - iii) ascribe to this node the current state of its neighbour.
- Node-update rule*

Actually, there is an alternative rule which may look equivalent:

- i) choose one link at random,
  - ii) choose one of its ends at random,
  - iii) ascribe to this node end the current state of the other end.
- Link-update rule*

At each time step, this update rule is applied  $N$  times if  $N$  is the network size, so that *on average* each node is updated once. The two rules are equivalent for regular lattices, but not in the case of complex networks [sucheki\_analytical]. The voter model defines a markovian stochastic process. There are two absorbing states: all spins up and all spins down (they are called absorbing because when the system reaches one of these states, by the update rules it remains there forever). If we identify network nodes as people and spin up and down with two possible opinion, the absorbing states represent unanimous consensus. There is always a chance that a finite size network reaches consensus. In fact, **the mean time required to reach consensus in a finite network is always finite**, whether it is a regular lattice or a complex network. The mean is intended here an ensemble mean, computed over the set of all evolution histories and all initial conditions. However, conceptually, there are different scenarios when infinite size networks are considered instead.

The mean time  $\langle \tau \rangle$  to reach consensus in a finite, regular lattice of  $N$  nodes and dimension

$d$  is known to scale as:

$$\langle \tau \rangle (N) \sim \begin{cases} N^2 & \text{if } d = 1 \\ N \cdot \ln N & \text{if } d = 2 \\ N & \text{if } d > 2 \end{cases}$$

For networks, one could guess that  $\langle \tau \rangle (N) \sim N$  as happens for high dimensional lattices. However this is not true in general. For instance, [11] found analytically that, for uncorrelated, scale-free networks with degree distribution  $P(k) \sim k^{-\gamma}$  and *node-update* rule:

$$\langle \tau \rangle (N) \sim \begin{cases} N^\alpha, \alpha < 1 & \text{if } \gamma < 3 \\ \frac{N}{\ln N} & \text{if } \gamma = 3 \\ N & \text{if } \gamma > 3 \end{cases}$$

Additionally, they found out numerically that also scale-free networks with degree-degree correlations align well with these formulas. The authors of [10] numerically found for Barabasi-Albert (BA) networks a scaling  $\tau \sim N^{0.88}$ , which is compatible with the theoretical prediction  $\tau \sim \frac{N}{\ln N}$ . Furthermore, they found that when the *edge-update* rule is used instead, a scaling  $\tau \sim N$  is found, which is the same as that of high dimensional lattices. A standard order parameter used to measure the ordering process of the voter dynamic is the *average interface density*  $\rho$ , defined as the density of edges connecting nodes with different states:

$$\rho(\sigma) := \frac{\sum_{i,j} [A_{i,j} \cdot \mathbb{1}(\sigma_i, \sigma_j)]}{\sum_{i,j} A_{i,j}}, \quad \text{where } \mathbb{1}(\sigma_i, \sigma_j) := \frac{1 - \sigma_i \sigma_j}{2} = \begin{cases} 1 & \text{if } \sigma_i \neq \sigma_j \\ 0 & \text{if } \sigma_i = \sigma_j \end{cases} \quad (3.1)$$

The interface density  $\rho$  measures the level of disorder present in the network, and is zero when consensus is reached.

### 3.3 | Numerical Simulations

For my simulation, I tried to verify the scaling law  $\tau(N) \sim N \cdot \ln N$  on BA networks.

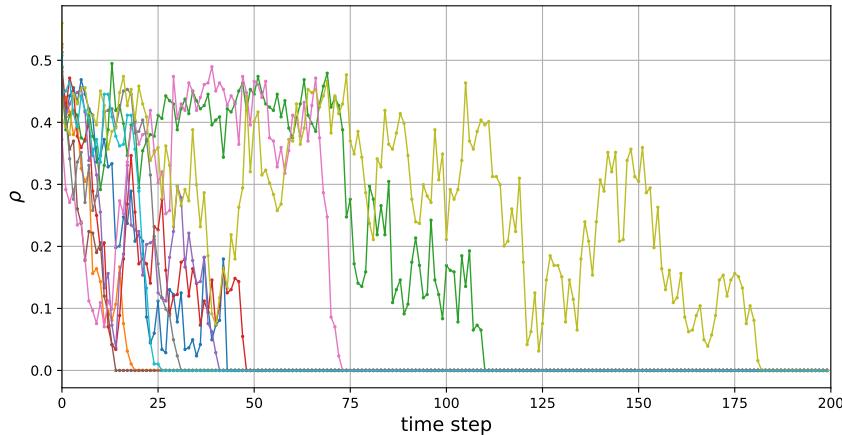


Figure 3.1: Evolution of the voter model on Barabasi-Albert networks with  $N = 100$  and  $\langle k \rangle = 8$ . Each line is a trajectory on a different network instance. The initial state of each node is sampled with equal probability as  $+1$  or  $-1$ .

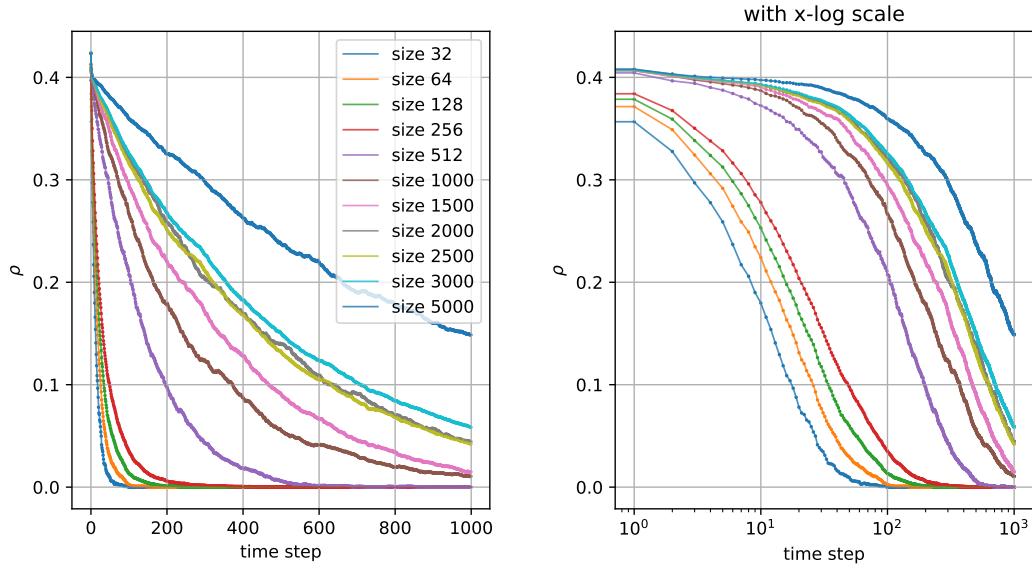


Figure 3.2: Evolution of the average interface density [Eq. 3.1] in Barabasi Albert networks of mean degree  $\langle k \rangle = 6$  different sizes. Data is averaged over 500 realizations.

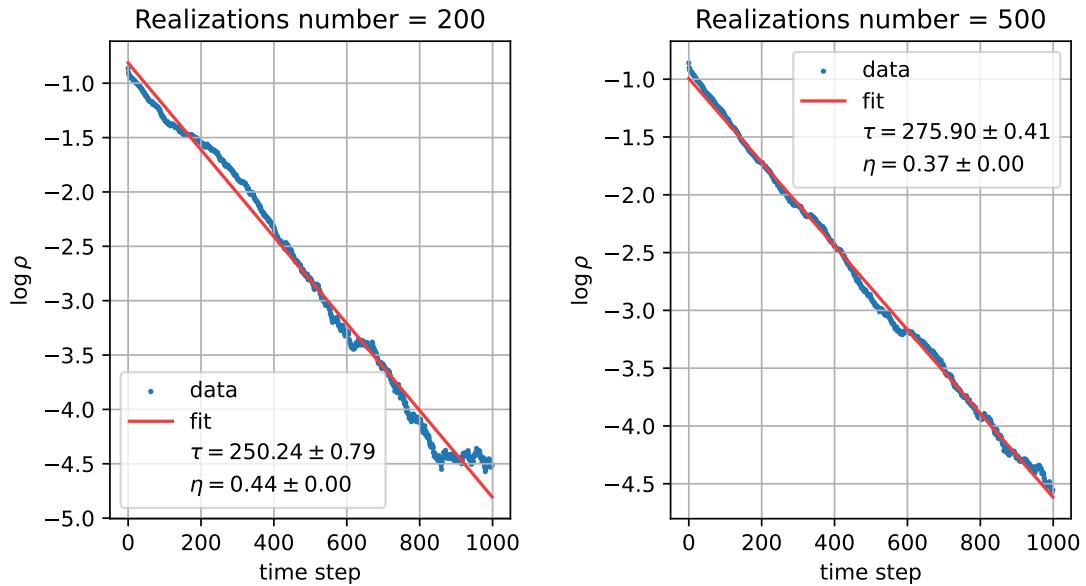


Figure 3.3: Comparison between exponential fit parameters estimated from 200 (left) and 500 (right) different realization on BA networks with same size  $N = 1000$ .

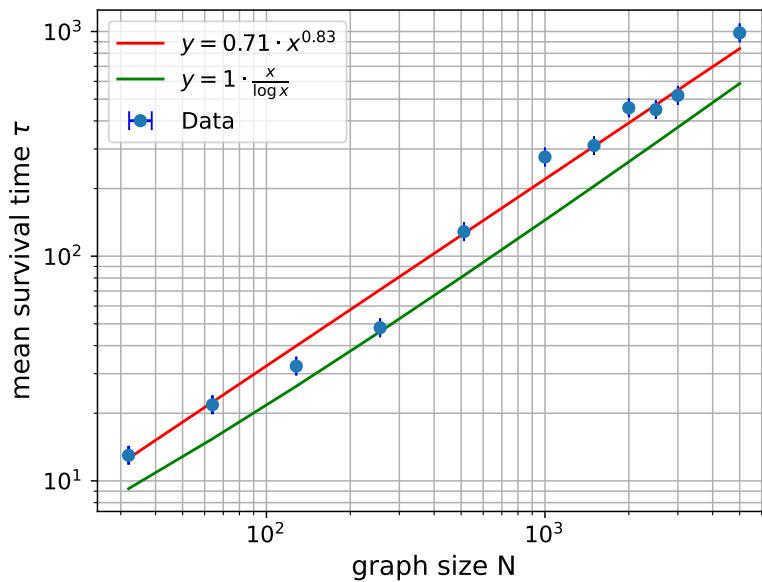


Figure 3.4: Average survival times for Barabasi Albert networks of different sizes, as results from the exponential fit of the curves in [Fig: 3.2]. Data is fitted with a power law  $y = \alpha \cdot x^\gamma$  with free parameters  $\alpha, \gamma$  with the method of least squares For comparison, also the theoretical expectation  $\tau \sim \frac{N}{\log N}$  is plotted (green line). One can see that the slope of the two curves are very similar.

## 4 | Task 46: EU transportation network II

---

### 4.1 | Task Description

---

The aim of this task is to reconstruct the railway networks of European countries from the raw geographical data provided in the open database [12, *EuroGlobalMap*, 2019 release]. The requested output for each network is two files: one containing the edge list and one containing the node metadata *[nodeLabel, latitude, longitude, country\_name, country\_ISO3]*.

### 4.2 | Data extraction

---

The choice of what exactly the nodes of the network should represent is not specified (a possibility is to identify nodes with cities, but one could also opt for administrative district or even larger regions). I decided to build networks where **nodes** correspond to **cities** where a railway station is present, and two nodes are connected through an **edge** if they are **consecutive stops** of some rail. Accomplishing this task was not straightforward. The best way I found involves first creating a preliminary network, where nodes do not have any specific physical significance, and, subsequently, extracting the desired network as a subset of it.

I consulted *EGM19\_DataSpecification.pdf*, Annex C - Definition of Features and Attributes to understand what files I needed to look for in the database, Python library GeoPandas to open the shapefiles and NetworkX to create and manipulate graph objects. Station's data encoded in *Point* objects containing the station's latitude and longitude, while railway data is encoded in *LineString* objects, which are lists of points that connected together form a segmented line. Besides, geometrical data comes with a number of attributes. These include :'ICC', the 2- character country IS03 code (es. IT, for Italy), 'NAMA1', the station name, and 'EXS', the existence category of the rail (e.g. abandoned, operational, under construction...).

As mentioned, the desired networks are extracted as a subset of a preliminary network. The latter is built upon the data contained in file *RailrdL.shp* with the following steps:

1. open the shapefile *RailrdC.shp/.shx/.dbf* with Geopandas and extract relevant attributes
2. initialize an empty graph with NetworkX
3. iterate through the dataframe rows. For each LineString object, assign each of its point-like components to a node of the graph. Draw an edge between each pair of successive components.

Figure [Fig: 4.1] shows what the resulting network looks like. At this stage, the nodes have no particular physical meaning: they are just the starting and ending points of the straight segments that form the rail line.

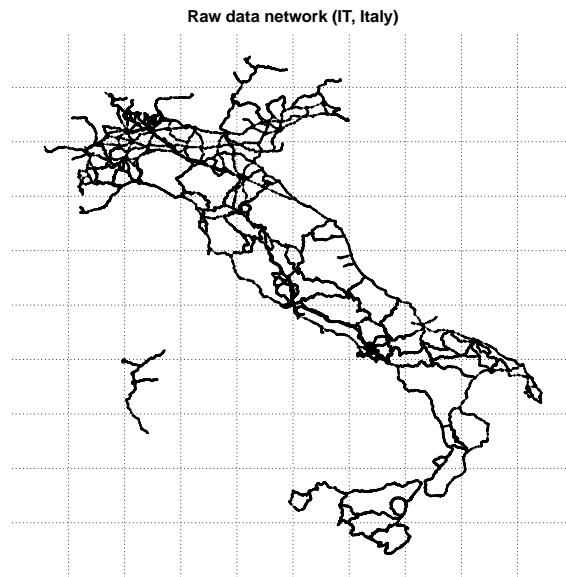


Figure 4.1

The following step consists in selecting only the nodes which corresponds to actual city stations, and rewire the links. To achieve this:

1. city stations data is loaded from file *RailrdC.shp/.shx/.dbf*,
2. *Node labeling*: two new attributes are assigned to nodes of the preliminary network: "label" and "is\_near\_city". These fields are both "empty" by default. A loop checks if the preliminary network nodes' coordinates match any of the stations coordinates within a given threshold distance  $d$ . Nodes in the preliminary network are very densely packed, and in general more than one node will satisfy the threshold condition. The best match is assigned attribute "label = station\_name", while the others that also satisfy threshold are assigned "is\_near\_city = station\_name".
3. a new empty network is initialized and the nodes of the preliminary network with "label"  $\neq$  "empty" (i.e. all the best matches) are added to it,
4. *Edge creation*: a maximum radius parameter  $R$  is defined. The presence of an edge is checked only for the pair of nodes within distance  $< 2R$ . The other pairs are assumed to not be connected: this introduces a possible error source but was necessary to obtain feasible computations.
5. For each node pair in the new graph, the shortest paths between them in the *old* graph is calculated. If a shortest path exists where all intermediate nodes have both fields "label" and "is\_near\_city"  $\equiv$  "empty" (i.e. there exists a rail which connects the cities with no intermediate stops), these nodes are connected with an edge in the new graph.

Notice that two threshold parameters  $d$  and  $R$  were added in this procedure. They were tuned by trial and error, and depended upon the country considered. The introduction of the auxiliary attribute "is\_near\_city" may seem unnecessary, but was motivated by data inspection [see Appendix, figure 2.6]. In fact, rail bifurcations often start slightly before or after entering a city station. In reality, the train must pass through the city station even if it takes the bifurcation. Without the "is\_near\_city" field, the final network contained more edges than it should. Results for Italy are shown in [Figure: 4.2], while all other networks

visualizations are in GitHub repository.

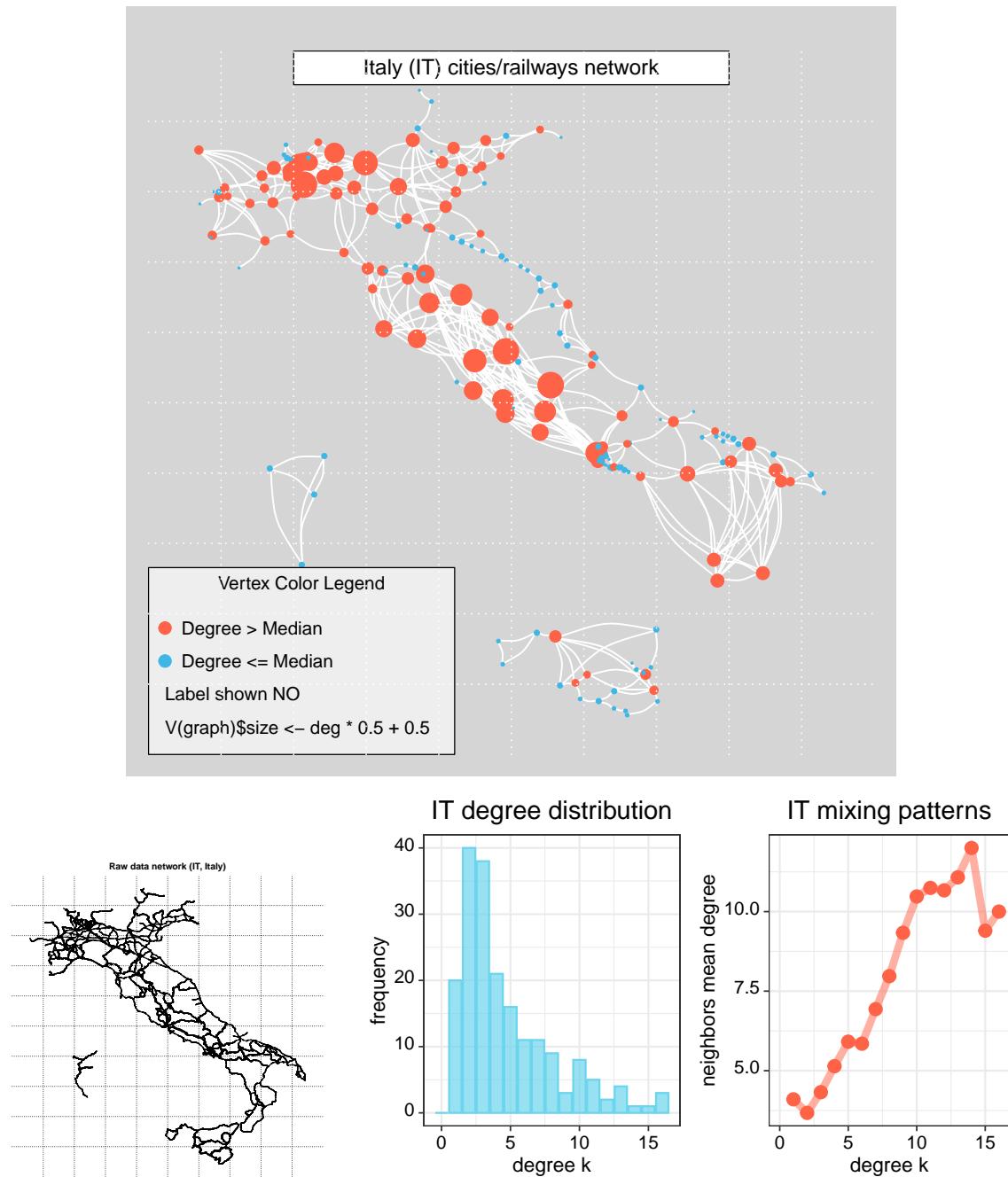


Figure 4.2: IT (Italy) cities/ railways networks. Notice for instance the high density of edges in the center-left zone of the graph: it is a consequence of the fact that the stations "Roma Termini" and "Roma Tiburtina" are missing from file "RailrdC.shp" ! Many stations result directly connected where in reality they are not because Rome stations are in between.

## Appendix

### Supplementary figures

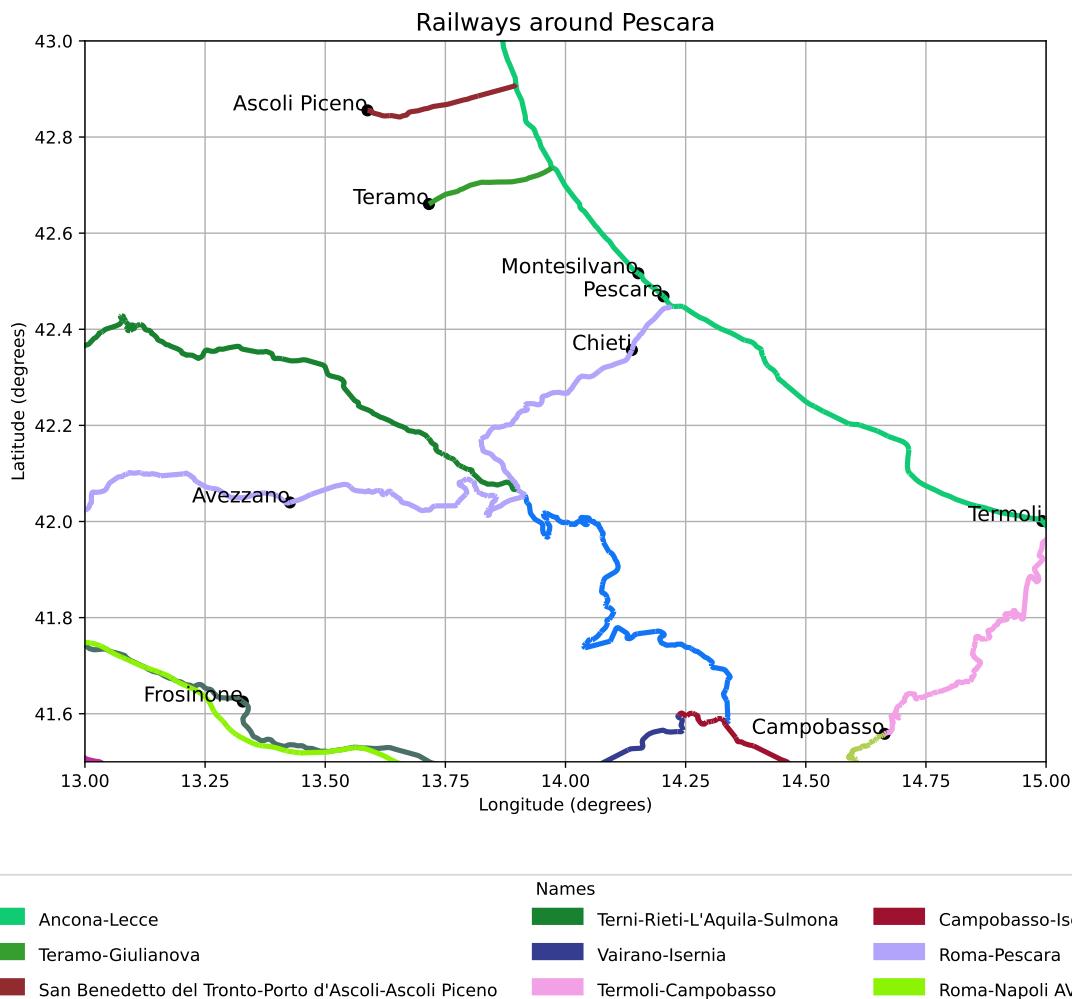


Figure 4.3: An example to justify the need for the "is near city" check: the rail bifurcation "Roma - Pescara" joins the "Ancora - Lecce" rail just before entering Pescara from south. Without the check implementation, edge "Termoli-Chieti" is created whereas in reality it does not exist: a train heading to Chieti must first stop in Pescara.

# 5 | Bibliography

---

## Task 01: Ising Model

---

- [1] S. N. Dorogovtsev and A. V. Goltsev. *Critical phenomena in complex networks*. Reviews of Modern Physics, 2008.
- [2] M. Leone and A. Vazques. “Ferromagnetic ordering in graphs with arbitrary degree distribution”. In: *The European Physical Journal B* (2002). DOI: [10.1140/epjb/e2002-00220-0](https://doi.org/10.1140/epjb/e2002-00220-0).
- [3] A. Aleksiejuk, J. A. Holyst, and D. Stauffer. “Ferromagnetic Phase Transition in Barabasi-Albert Networks”. In: *Physica A* (2002). DOI: [10.48550/arXiv.cond-mat/0112312](https://doi.org/10.48550/arXiv.cond-mat/0112312).
- [4] Daniel Wellingut. *Final report for the course of Complex Networks*. <https://github.com/dw-unipd/PoCN-Project-DW>. [Accessed: 14/07/24].

## Task 18: Turing Patterns

---

- [5] Mikhailov A. Nakao H. “Turing patterns in network-organized activator–inhibitor systems”. In: *Nature Phys* (2010). DOI: [10.1038/nphys1651](https://doi.org/10.1038/nphys1651).
- [6] Miriam Zara. *ComplexNet\_Project*. [https://github.com/miriamzara/ComplexNet\\_project](https://github.com/miriamzara/ComplexNet_project).
- [7] Takashi Miura Shigeru Kondo. “Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation”. In: *Science* (2010). DOI: [10.1126/science.1179047](https://doi.org/10.1126/science.1179047).
- [8] J. D. Murray. *Mathematical Biology II*. 3rd ed. Interdisciplinary Applied Mathematics. Springer New York, NY, 2003. DOI: [10.1007/b98869](https://doi.org/10.1007/b98869).
- [9] Arianna Bianchi. *The Dynamics of Biological Systems*. Mathematics of Planet Earth. Springer Cham, 2019. DOI: [10.1007/978-3-030-22583-4](https://doi.org/10.1007/978-3-030-22583-4).

**A note to the bibliography.** This assignment specifically focuses on Turing patterns in networks. However, for someone unfamiliar with Turing patterns, a few preliminary read may be useful. Article [7], targeted at biologists, offers an intuitive overview of the concept with minimal mathematical details. To better understand the mathematical side, [8, *Chapter 2: Spatial Pattern Formation with Reaction Diffusion Systems*] provides a comprehensive and detailed explanation of Turing Patterns in a continuous medium. Another valuable resource is [9, *Chapter 7: The Turing Model for Biological Pattern Formation*], which, while shorter than Murray’s chapter, still offers insightful observations.

## Task 28: Voter Model

---

- [1] S. N. Dorogovtsev and A. V. Goltsev. *Critical phenomena in complex networks*. Reviews of Modern Physics, 2008.

- [10] K. et al Suchecki. "Voter model dynamics in complex networks: Role of dimensionality, disorder, and degree distribution". In: *Physical Review* (2005). doi: [10.1103/PhysRevE.72.036132](https://doi.org/10.1103/PhysRevE.72.036132).
- [11] V. Sood and S. Redner. "Voter Model on Heterogeneous Graphs". In: *Physical Review Letters* (2005). doi: [10.1103/PhysRevLett.94.178701](https://doi.org/10.1103/PhysRevLett.94.178701).

### Task 46: European transportation network II

---

- [12] EuroGepographics. *EuroGlobalMap*. <https://www.mapsforeurope.org/datasets/euro-global-map>.