Complex Networks project

Study of a configuration model

Victor Delapalme
M2 PCS

Task 1 : Generating a graph

1) Generating an even **degree sequence** ($p_1 = 1-\pi$, $p_4 = \pi$) :

$$(1, 1, 4, 1, 4, ..., 4, 1)$$

2) Generate **stubs** :

$$(1, 2, 3, 3, 3, 3, 4, 5, 5, 5, 5, ..., N-1, N)$$

3) Randomly **shuffle** stubs and **pair** them two by two

$$(45, 13, 2, 18, ..., 19, 19, ..., 45, 13, 27, 12, ...)$$

4) If there is a **self edge** or a **double edge**, go back to 3)

Task 1 : Generating a graph : Result

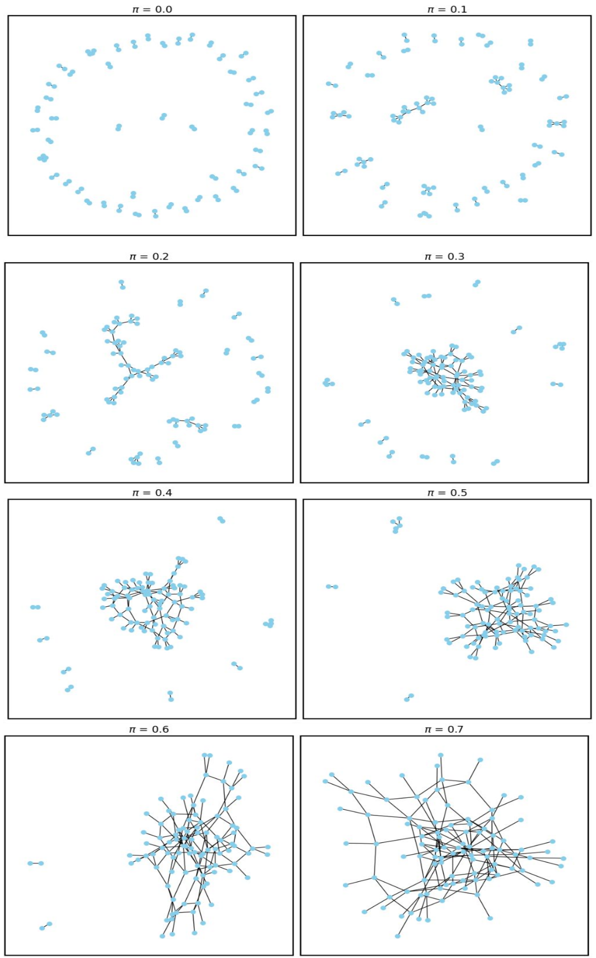
- Set of **edges** :

  E = { (45, 13), (2, 18), ... , (27, 12) }

- **Neighbor** dictionary :

  { 1 : [53], 2 : [18], 3 : [19, 28, 10, 42], ... }

Figure taken from :
https://github.com/EliaBronzo/Complex_network_project

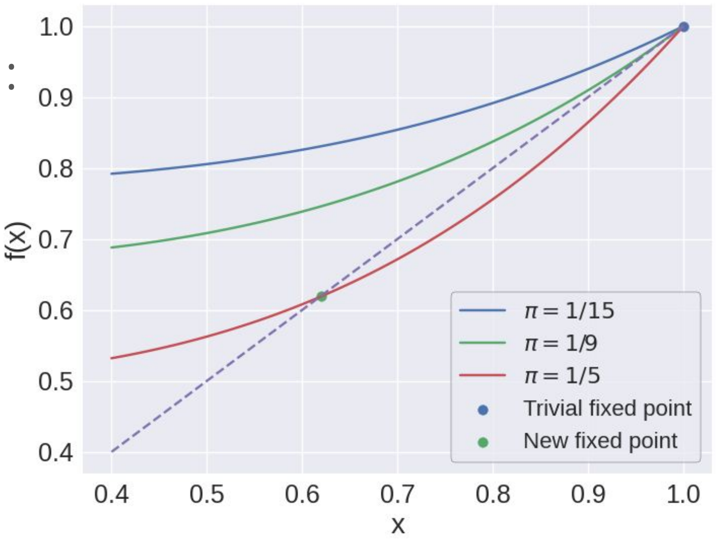


Task 2 : Emergence of a GCC : Intuition

1) Choose a **random edge** :

2) Choose randomly one of the **end vertex** :
   - It has **0** other neighbors with probability $\mathbf{q_1} = 1-\pi/(1+3\pi)$
   - It has **3** other neighbors with probability $\mathbf{q_4} = 4\pi/(1+3\pi)$

$$\Rightarrow \gamma = q_1 + q_4\gamma^3$$



3) Stable solution appears for $\boldsymbol{\pi_c} = \mathbf{1/9}$

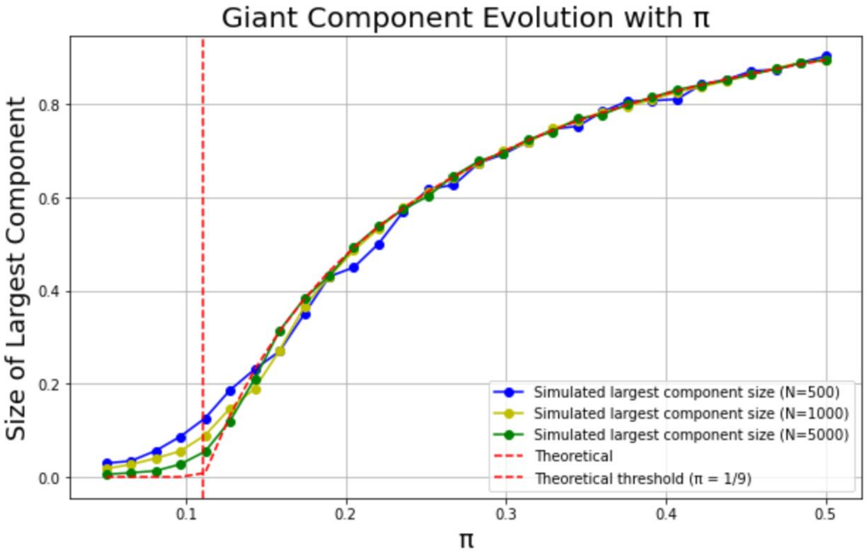
Task 2 : Emergence of a GCC : Results

- Theoretical prediction :

$$\gamma = 1 - (1-\pi)\left(\frac{1-\sqrt{\pi}}{2\sqrt{\pi}}\right) - \pi\left(\frac{1-\sqrt{\pi}}{2\sqrt{\pi}}\right)^4$$

- Numerical simulation : Search for connected component Breadth First Search (**BFS**), with different initialisations, until you have found the largest one.

Task 2 : Emergence of a GCC : Results



- We see that the transition gets **sharper** for larger N

Figure made by Loumi Gatouillat

Task 3 : Emergence of a giant q-core : Intuition

- Definition : **Maximal** subgraph of **minimal** degree **q**

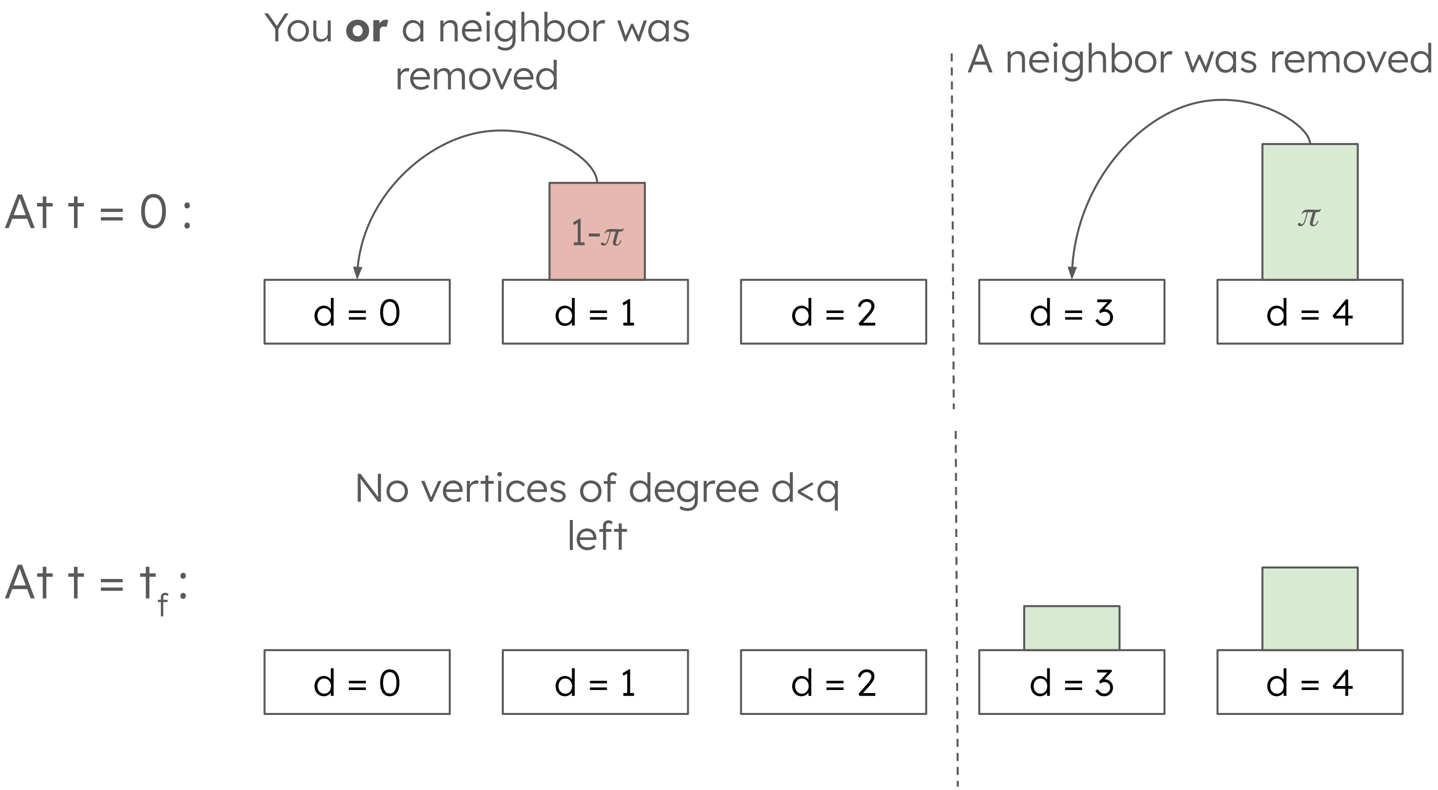
- **How** to find it ?

  → **Remove iteratively** vertices with less than **q** neighbors, until **none are left**

- Analytically : Rewrite this process as a set of **coupled differential equations** for the number of vertices of degree d = 0, 1, 2, 3, 4

$$\partial_t[(1-t)p_d(t)] = -\frac{\chi_d p_d(t)}{\overline{\chi_d}} + \frac{\overline{d\chi_d}}{c(t)\overline{\chi_d}}\left[(d+1)p_{d+1} - dp_d\right]$$

Task 3 : Emergence of a giant q-core : Intuition



Task 3 : Emergence of a giant q-core : Results

- $1 - t_f$ = Size of the q-core
- Discontinuous jump to **…** at ***π* = 2/3**
- Again, transition will get **sharper** with larger N

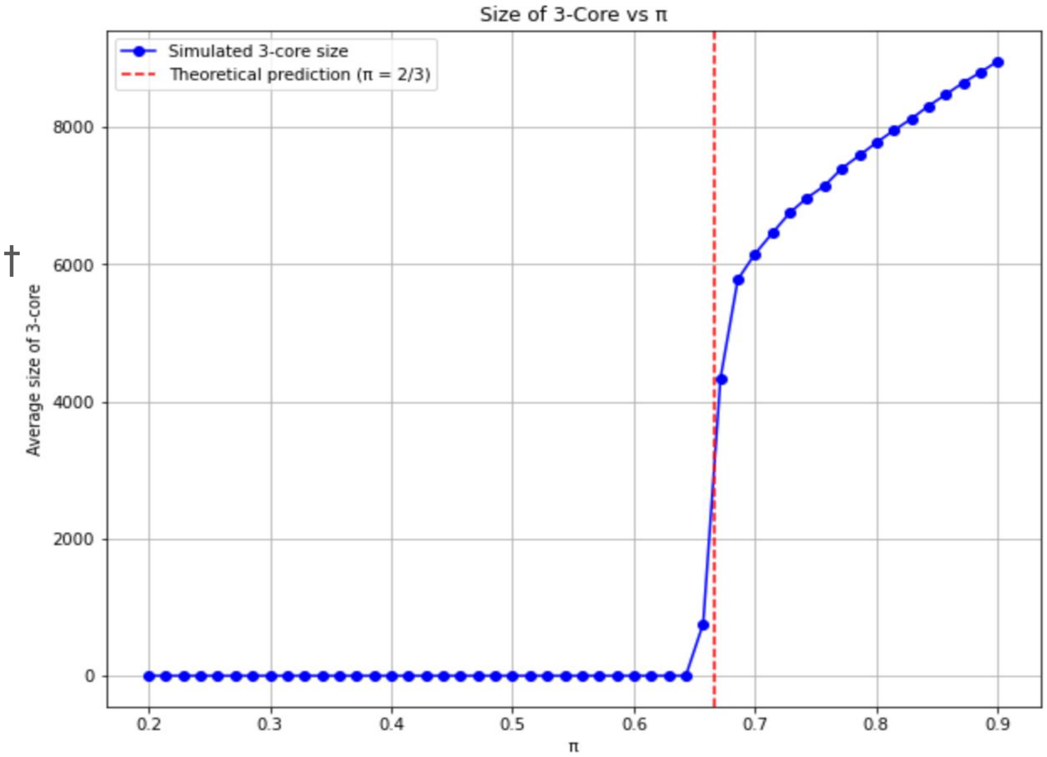


Figure made by Loumi Gatouillat

Task 4 : Ising model on a graph : MCMC

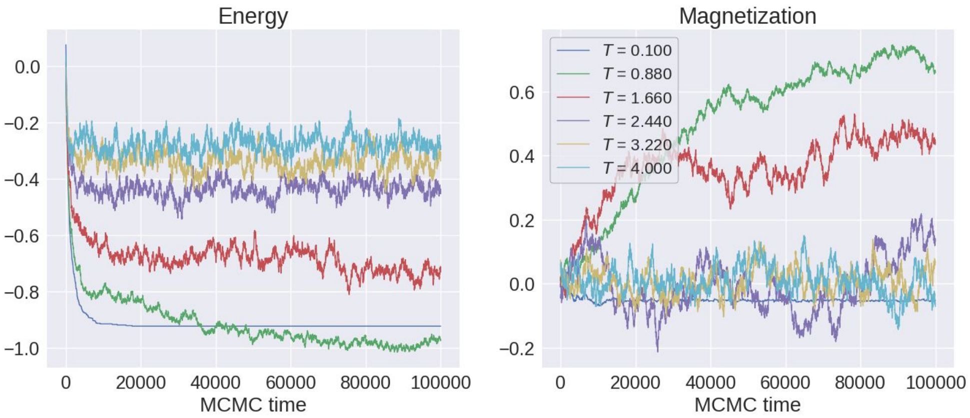
Metropolis Hastings algo :

1) **Propose** a spin flip

2) Compute the (local) **energy difference ΔE**

3) **Accept** with proba p = Max($e^{-\beta \Delta E}$, 1)

Example for N = 1000 and π = 0.4

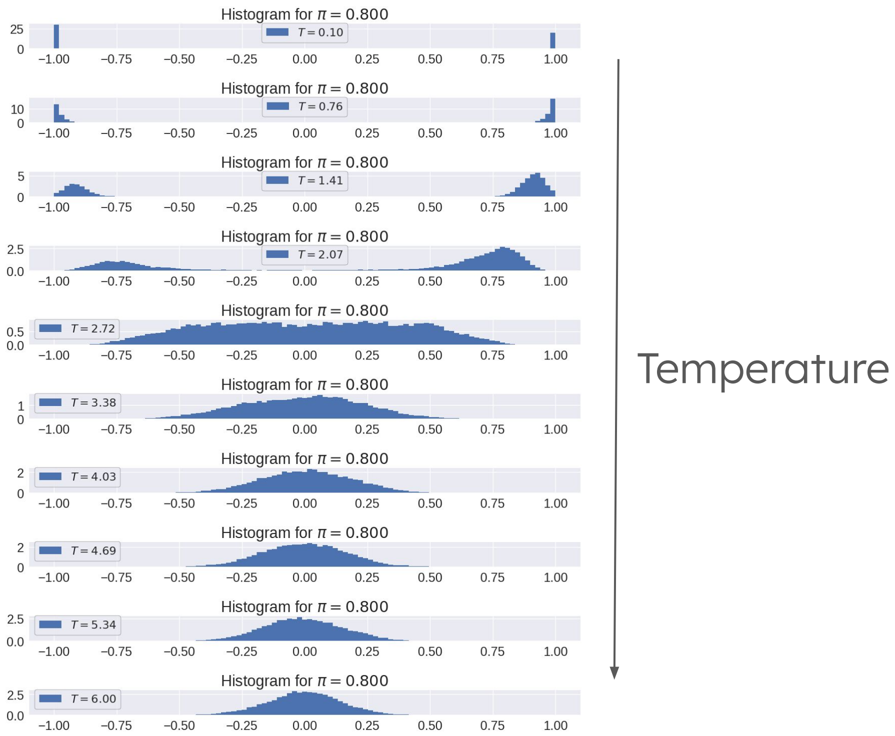


→ We can spot the **phase transition** by looking a the **magnetization**

Task 4 : Ising model on a graph : MCMC results

Histogram of **magnetization** :

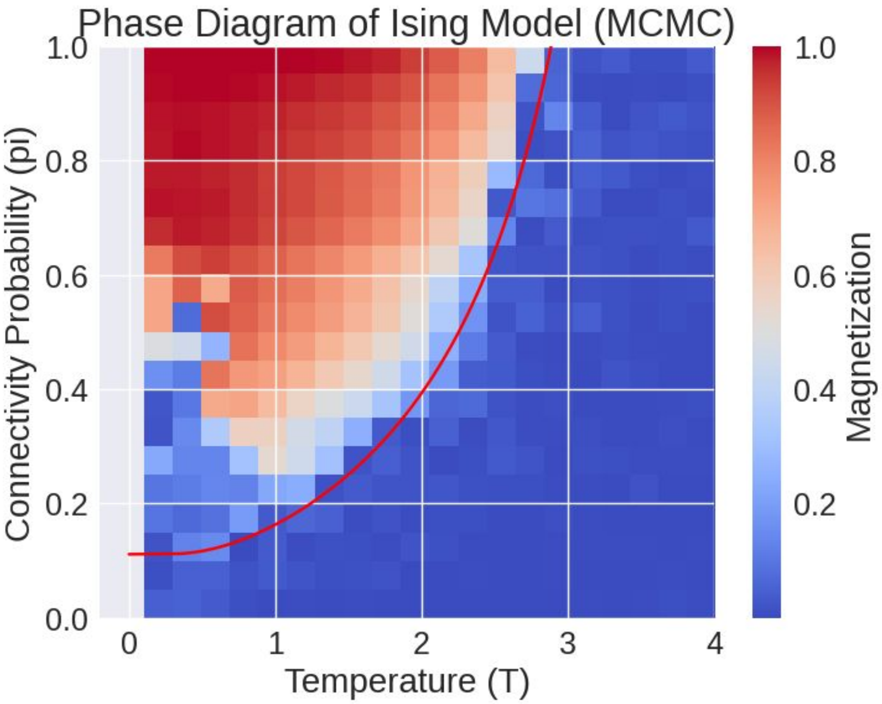
- **Bimodal** distribution at low T
  ⇒ **Ferromagnetic** phase
- **Gaussian** at high T
  ⇒ **Paramagnetic** phase
- In between : **broad** distribution



Task 4 : Ising model on a graph : MCMC results

Phase diagram using MCMC :

- Two phases : **PM** and **FM**
- Transition is **not sharp** because N = 1000 is finite
- At **low T**, MCMC struggles to stabilize, even after 4×10^5 steps



Task 4 : Ising model on a graph : Belief-Propagation

Idea : Compute the partition function iteratively using :

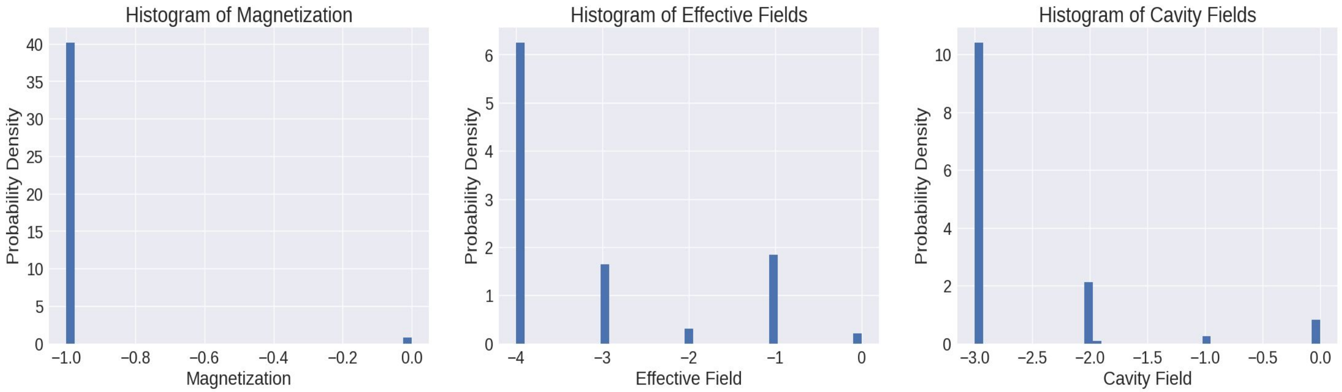
- Cavity fields (messages) $$h_{j|i} = \frac{1}{2\beta} \sum_{k \in \partial j - i} \ln \frac{\cosh(\beta(h_{k|j} + 1))}{\cosh(\beta(h_{k|j} - 1))}$$
- Effective fields $$h_i = \frac{1}{2\beta} \sum_{j \in \partial i} \ln \frac{\cosh(\beta(h_{j|i} + 1))}{\cosh(\beta(h_{j|i} - 1))}$$

⇒ Magnetization : $m_i = \tanh \beta h_i$

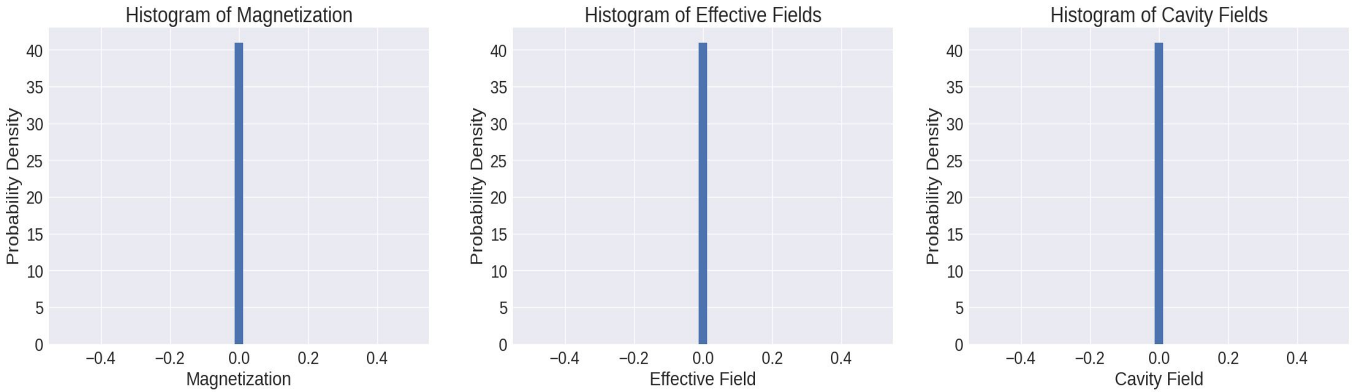
⚠️ **Only** exact on a **tree**

Task 4 : Ising model on a graph : BP results

T = 0.1



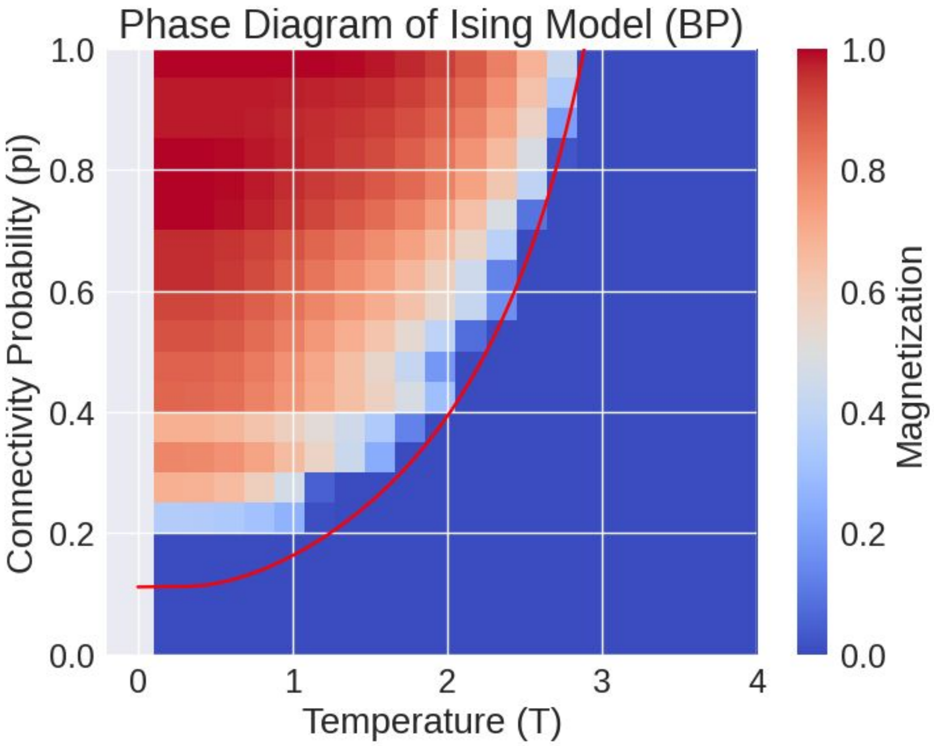
T = 5



Task 4 : Ising model on a graph : BP results

Phase diagram using BP :

- **Similar results** as MCMC, but much **faster !**
- **Physical results**, even on a graph with **loops**



Task 4 : Ising model on a graph : Population Dynamics

→ **Self consistent integral equation** for P(h) and P_{cav}(h) :

$$P(h) = \sum_{d=0}^{\infty} p_d \int \prod_{l=1}^{d} \mathrm{d}h_l P_{cav}(h_l) \delta\left(h - \frac{1}{2\beta}\sum_{l=1}^{d} \ln \frac{\cosh(\beta(h_l+1))}{\cosh(\beta(h_l-1))}\right)$$

- **Delta solution** is always a solution, but **only stable** for :

$$\beta > \beta_c(\pi) = \operatorname{arctanh}\left(\frac{1+3\pi}{12\pi}\right)$$

- Note : **Finite T** phase transition only for $\boldsymbol{\pi > \pi_c}$ = 1/9

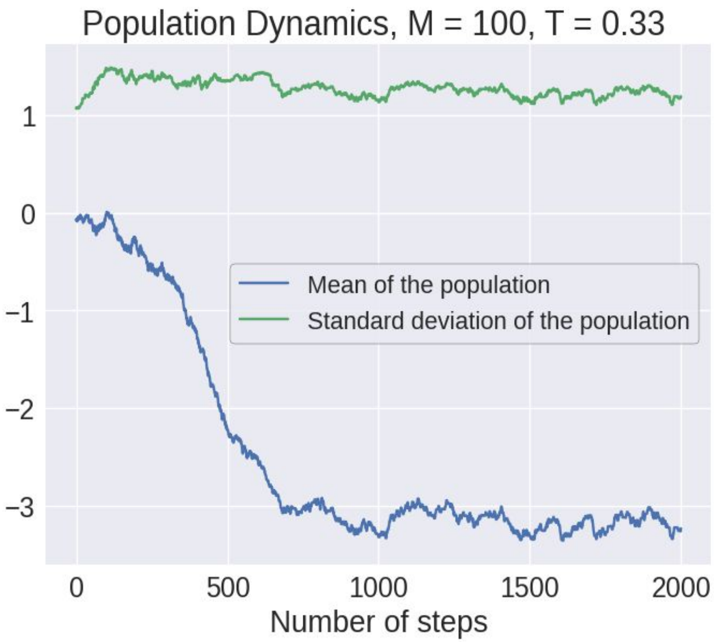
Task 4 : Ising model on a graph : Population Dynamics

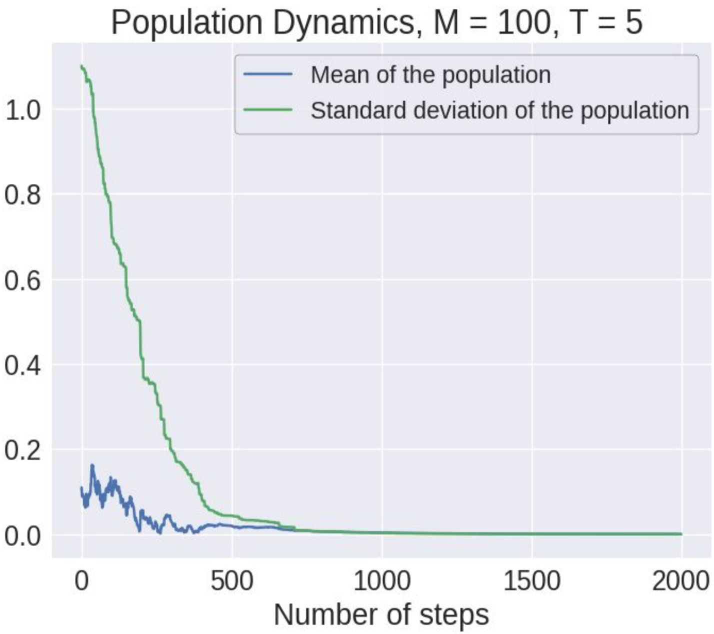
Idea : **Iterative** procedure to solve the **self consistent integral equation** for P(h) and $P_{cav}(h)$

- Generate M cavity fields h_i , i = 1, .., M (the **population**)
- Draw **d** from q_{d+1} and select **d+1 distinct indices** from 1 to M
- **Update** the population using :

$$h_i \leftarrow \frac{1}{2\beta} \sum_{l=i}^{d} \ln \frac{\cosh(\beta(h_{i_l}+1))}{\cosh(\beta(h_{i_l}-1))}$$

Task 4 : Ising model on a graph : PD results

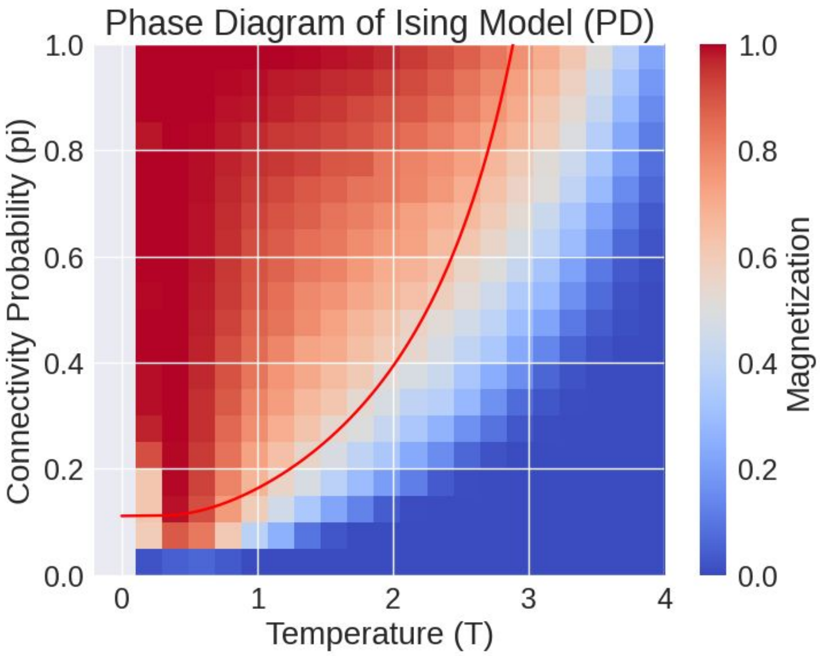




Task 4 : Ising model on a graph : PD results

Phase diagram using PD :

- **Worse results** as MCMC, but **a bit faster**
- **M** has to **scale with N**, hence the **number of steps also**

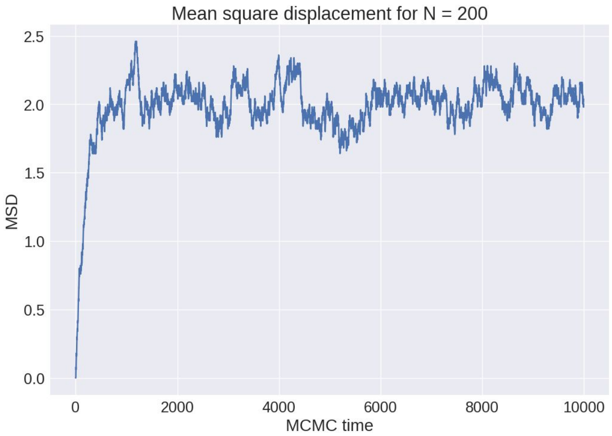


Task 5 : Graph inference

Question : Can we **infer the graph** from a sample of the **spins** ?

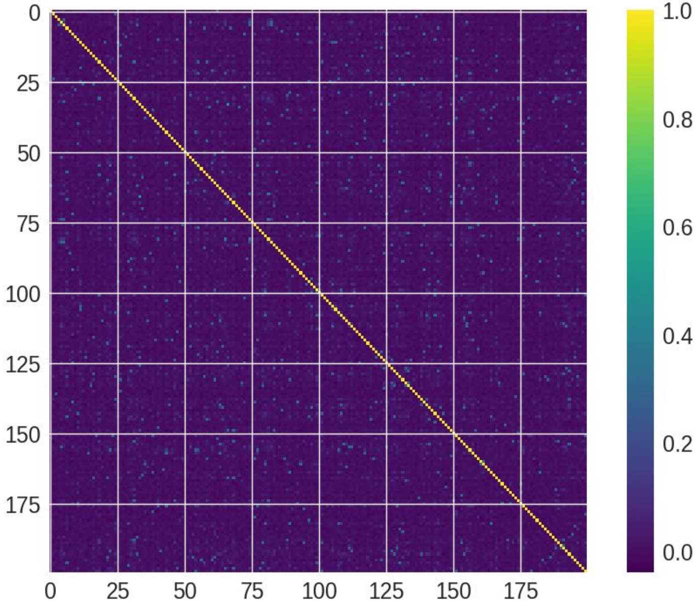
$$\mathcal{D} = \{s_i^m\}_{i=1,\,..,\,N}^{m=1,\,..,\,M} \longrightarrow G = (V, E)$$

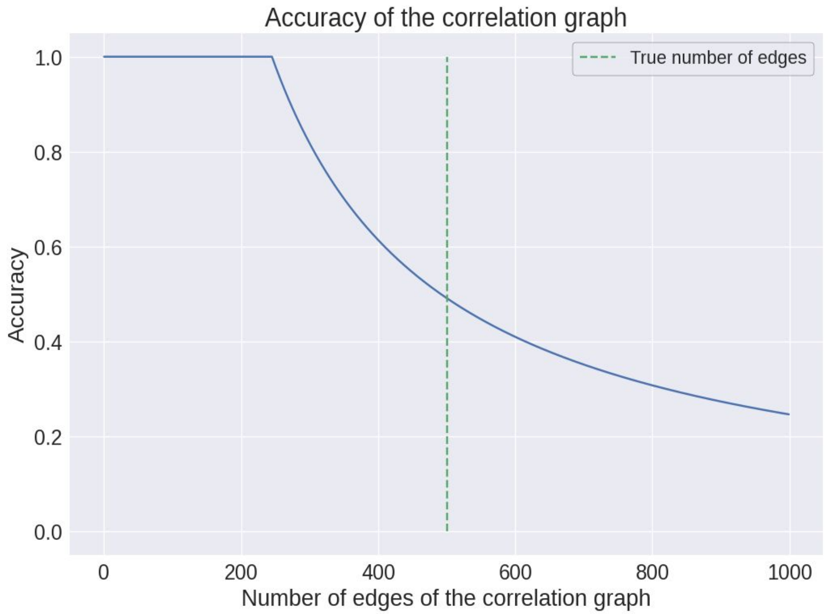
1) Generate data : i.i.d. samples
   Here : N = 200, π = ½, T = 10/3
2) Using correlations
3) Using mean field approximation
4) Comparing the results



Task 5 : Graph inference

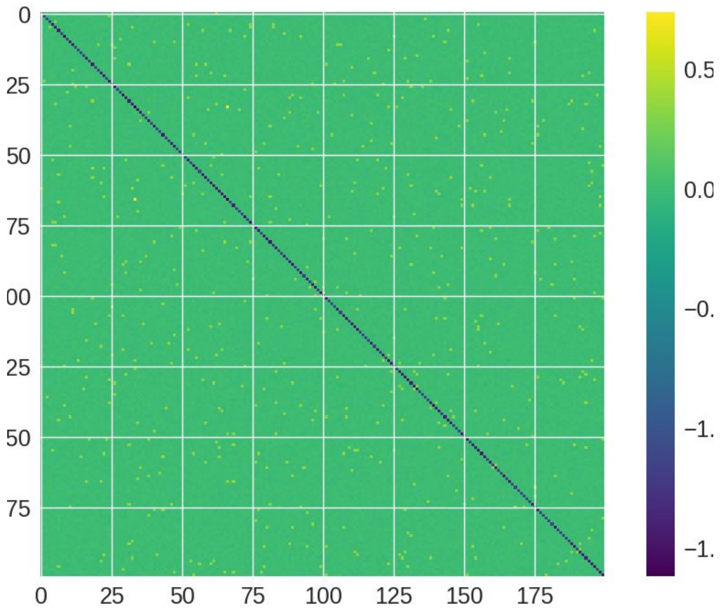
2) Using **correlations** : $c_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$

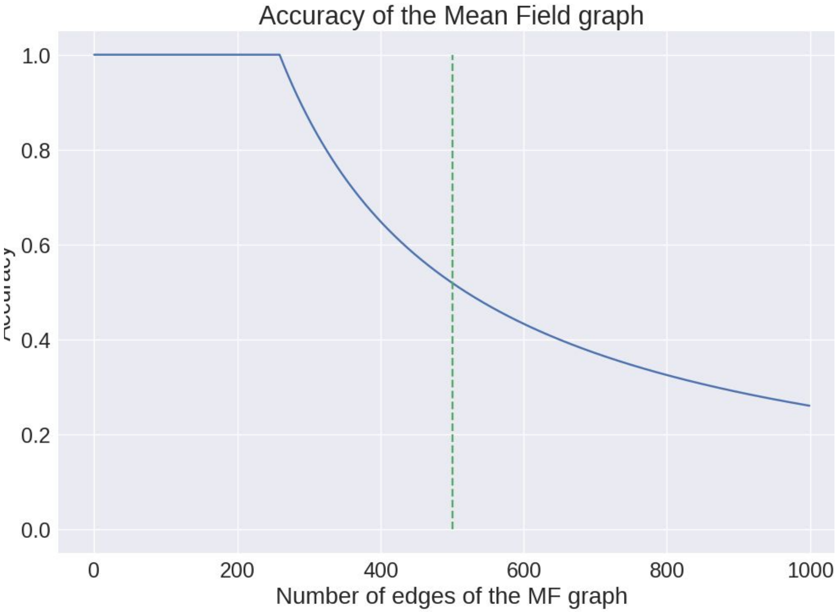




Task 5 : Graph inference

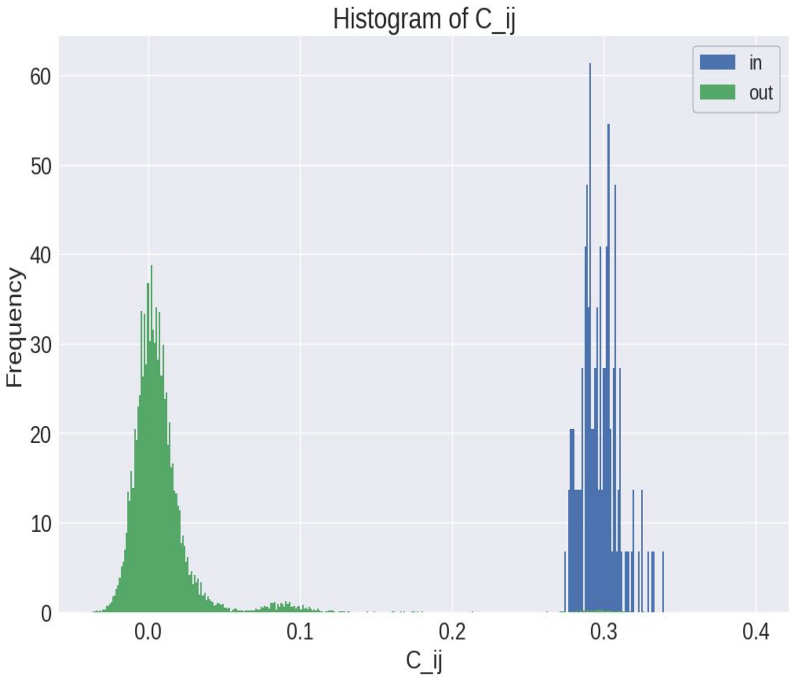
3) Using **MF approximation** : $J_{ij} = -(C^{-1})_{ij}$

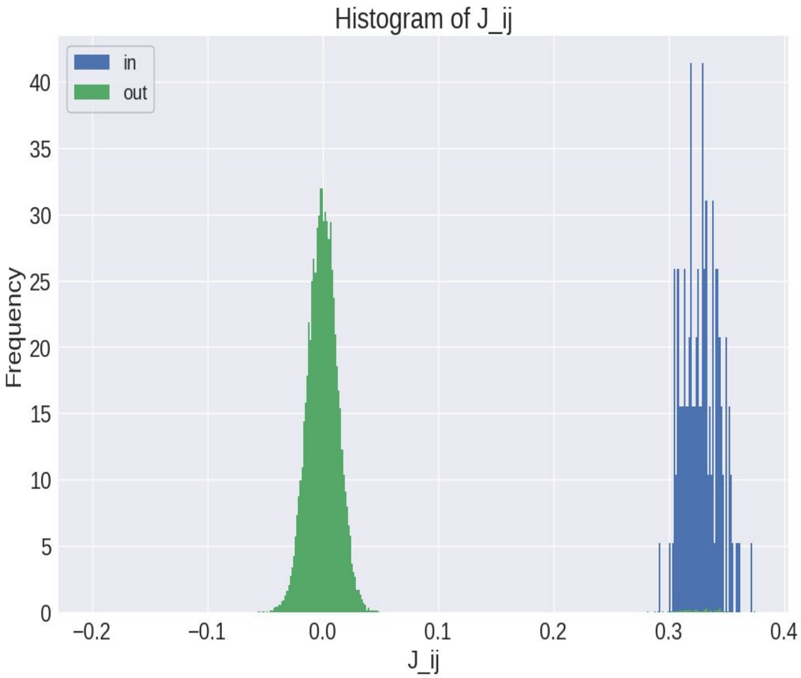




Task 5 : Graph inference

4) Comparison of the two methods :





Thank you for listening !