1 The stability criterion for graph communities

The partition of a graph into communities (or clusters) as been widely studies those last two decades. Clustering comes indeed pretty handy to gain insight into the underlying structure of a system represented by a network. In some casesbuild a simplified functional description based on the clusters. Many partitioning methods have been proposed, each relying on a particular measure to quantify the quality of a community structure. Such methods include normalized cut, (α,ϵ) clustering or modularity and its variants and extensions. See for instance [1] for a 2010 survey. In this work, we choose the stability approach, which is based on the statistical properties of a dynamical process taking place on the network. This approach was initially presented in [2] and further expended in [3] and [4].

The stability method presents a number of advantages. First, it does not require the number of communities to be specified beforehand, ensuring a natural partitioning of the graph. Second, it is flexible in the sense that it does not seek a *unique* optimal partition. Instead, it reveals several community structures, each appearing to be the most relevant at particular values of the Markov time: at a given time scale, natural clusters corresponds to sets of states from which escape is unlikely within that time scale. The stability method provides thus a dynamical interpretation of the partitioning problem. The Markov time acts as an intrinsic resolution parameter, as will be developed shortly. Finally, it is probably the most unifying approach since many of the standard partitioning measures find an interpretation through the stability framework.

In order to compute stability partitions in the next of this work, we make use of Michael Schaub's free software *PartitionStability*. This C++ implementation of the stability method with a MATLAB© interface is available at `https://github.com/michaelschaub/PartitionStability`. It relies on the Louvain algorithm [5] to optimize the stability quality function. This heuristic algorithm has been initially developed for modularity optimisation. However one can show that stability can be written as the *modularity* of a time-dependent network evolving under the Markov process [3]. Hence, the Louvain method can almost straightforwardly be applied to stability optimisation.

This section is devoted to the explanation of the stability measure, and how to find good clusterings using stability analysis. This theoretical part is intended to cover everything that is needed to make a proper, informed use of the stability toolbox. The stability measure has initially been presented for dicrete times in [2]. We follow the same approach here : discrete-time stability is developed in the first part of this section; it is then extended to continuous time in a second part; finally, a few tools to analyze the robustness of a partition are presented in the third part of the section.

1.1 Discrete-time stability as an autocovorariance

The stability criterion is based on the two-way relationship between graphs and Markov chains: On one hand, any graph has an associated Markov chain where the states are the nodes of the graph and the transitions probabilities between states are given by the weights of the edges. On the other hand, any Markov chain can be represented by a graph whose edges are weighted according to the transition probabilities. Concretely, consider a graph of N nodes whose $N \times N$ weighted adjacency matrix is denoted $\mathbf{A}$. Let $\mathbf{k} = \mathbf{A1}$; k_i is thus the total weight of the outgoing edges from node i. Let $\mathbf{K} = \mathrm{diag}(\mathbf{k})$. Then, by normalizing the rows of $\mathbf{A}$ we get the matrix $\mathbf{M} = \mathbf{K}^{-1}\mathbf{A}$, the transition probability matrix. $\mathbf{M}$ is row-stochastic (or right-stochastic) and $[\mathbf{M}]_{ij}$ is the probability to go from node i to node j. Consider a particle moving in the network according to the transition probabilities in $\mathbf{M}$. Now let $\mathbf{p}_t$ be the $1 \times N$ probability vector at

Markov time t, namely that $p_{t,i}$ is the probability that the particle is located in node i at time t. The dynamics of the discrete-time Markov process are given by :

$$\mathbf{p}_{t+1} = \mathbf{p}_t \mathbf{K}^{-1}\mathbf{A} = \mathbf{p}_t \mathbf{M}. \tag{1}$$

Now, suppose that the Markov chain is ergodic, i.e. that it is possible to go from every state to every state and that the Markov process is aperiodic. The ergodicity assumption implies that any initial state will asymptotically reach the same stationary solution. Let $\boldsymbol{\pi}$ be that stationary distribution, given by $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{M}$, and $\boldsymbol{\Pi} = \mathrm{diag}(\boldsymbol{\pi})$. Now, let $\mathbf{x}_t$ be the N-dimensional random indicator vector describing the position of a particle undergoing the above dynamics : $x_{t,i} = 1$ if the particle is located in node i at time t, and 0 otherwise. At stationarity, the *autocovariance matrix* of $\mathbf{x}$ is

$$\begin{aligned}
\mathbf{C}(\mathbf{x}_\tau, \mathbf{x}_{\tau+t}) &\triangleq \mathbb{E}\left[(\mathbf{x}_\tau - \mathbb{E}[\mathbf{x}_\tau])^\intercal(\mathbf{x}_{\tau+t} - \mathbb{E}[\mathbf{x}_{\tau+t}])\right] && (2)\\
&= \mathbb{E}\left[(\mathbf{x}_\tau - \boldsymbol{\pi})^\intercal(\mathbf{x}_{\tau+t} - \boldsymbol{\pi})\right] && (3)\\
&= \mathbb{E}\left[\mathbf{x}_\tau^\intercal \mathbf{x}_{\tau+t}\right] - \mathbb{E}[\mathbf{x}_\tau^\intercal]\boldsymbol{\pi} - \boldsymbol{\pi}^\intercal \mathbb{E}\left[\mathbf{x}_{\tau+t}\right] + \boldsymbol{\pi}^\intercal\boldsymbol{\pi} && (4)\\
&= \boldsymbol{\Pi}\mathbf{M}^t - \boldsymbol{\pi}^\intercal\boldsymbol{\pi}, && (5)
\end{aligned}$$

where the fact that $\mathbf{C}(\mathbf{x}_\tau, \mathbf{x}_{\tau+t})$ only depends on the time difference t at stationarity is readily verified. Here, $^\intercal$ is the transposed sign and $\mathbf{M}^t$ is $\mathbf{M}$ at the power t. $[\mathbf{C}(\mathbf{x}_\tau, \mathbf{x}_{\tau+t})]_{ij}$ is interpreted as the correlation between $\mathbf{x}_{\tau,\,i}$ and $\mathbf{x}_{\tau+t,\,j}$. The independence on the initial time τ implies that it can indifferently be chosen equal to 0.

Suppose now a partition $\mathcal{P}$; we note $\mathbf{H}_\mathcal{P}$ the indicator matrix of $\mathcal{P}$. If c is the number of communities in $\mathcal{P}$, $\mathbf{H}_\mathcal{P}$ is a binary $N \times c$ matrix such that

$$[\mathbf{H}_\mathcal{P}]_{ik} = \begin{cases} 1 & \text{if node } i \text{ is in community } k, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Let us define $\mathcal{H}_\mathcal{P} : \mathbb{R}^{N\times N} \to \mathbb{R}^{c\times c} : \mathbf{B} \mapsto \mathcal{H}(\mathbf{B}) = \mathbf{H}_\mathcal{P}^\intercal \mathbf{B}\mathbf{H}_\mathcal{P}$. Let $\mathbf{X}$ be any $N \times N$ matrix, then $\mathbf{Y} = \mathcal{H}_\mathcal{P}(\mathbf{X})$ is a $c \times c$ matrix such that $[\mathbf{Y}]_{kl} = \sum_{i\in\mathcal{C}_k,\, j\in\mathcal{C}_l}[\mathbf{X}]_{ij}$, where $\mathcal{C}_k$ and $\mathcal{C}_l$ denote communites k and l of partition $\mathcal{P}$. One could thus say that operator $\mathcal{H}_\mathcal{P}$ returns the *clustered version* of any $N \times N$ matrix, namely the matrix where the contributions of every nodes belonging to the same community are gathered by summing them. Finally, let $\mathbf{y}_t = \mathbf{H}_\mathcal{P}^\intercal \mathbf{x}_t$ denote the c-dimensional community indicator vector: $\mathbf{y}_{t,\,k}$ is equal to 1 if the particle is in community k at time t and zero otherwise. Using those notations and the interpretation of $\mathcal{H}_\mathcal{P}$, the *clustered autocovariance matrix* for partition $\mathcal{P}$ at time t is defined as

$$\begin{aligned}
\mathbf{R}_t(\mathcal{P}) &= \mathcal{H}_\mathcal{P}\left(\mathbf{C}(\mathbf{x}_\tau, \mathbf{x}_{\tau+t})\right) && (7)\\
&= \mathbf{C}(\mathbf{y}_\tau, \mathbf{y}_{\tau+t}) && (8)\\
&= \mathbf{H}_\mathcal{P}^\intercal(\boldsymbol{\Pi}\mathbf{M}^t - \boldsymbol{\pi}^\intercal\boldsymbol{\pi})\mathbf{H}_\mathcal{P}. && (9)
\end{aligned}$$

Notice that $\mathbf{R}_t$ depends only on the topology of the graph and on the partition. If the graph has well defined communities given by $\mathcal{P}$ *over a given time scale*, we expect that the particle is more likely to remain within the starting community overt that time scale. This implies that the values of $\mathbf{y}_{0,\,i}$ and $\mathbf{y}_{t,\,i}$ are positively correlated for t in that time scale, which in turn implies large diagonal elements in $\mathbf{R}_t(\mathcal{P})$ and hence a large trace of $\mathbf{R}_t(\mathcal{P})$. The elements of $\mathbf{R}_t(\mathcal{P})$ are interpreted as follows in terms of the random walk of a particle : $[\mathbf{R}_t(\mathcal{P})]_{kl}$ is the probability that a particle is in community $\mathcal{C}_l$ after t discrete time-steps if it has started in $\mathcal{C}_k$ minus the probability that two independant random walkers are in $\mathcal{C}_k$ and $\mathcal{C}_l$, evaluated at stationarity. A good partition is such that there is a high likelyhood of remaining in the starting community over a given time scale. The definition of the stability of a *clustering* $\mathcal{P}$ follows naturally:

$$r_t(\mathcal{P}) = \min_{0\le s\le t} \sum_{i=1}^{c} [\mathbf{R}_s]_{ii} = \min_{0\le s\le t} \mathrm{trace}(\mathbf{R}_s). \tag{10}$$

Note that taking the minimum for all times up to t implies that the stability of the clustering at time t is large only if it is large for all times preceding t. This allows to assign a low stability to partitions where there is a high probability of leaving the community and coming back to it later. According to [4], this minimisation is unnecessary in most cases and we have $r_t(\mathcal{P}) \approx \text{trace}(\mathbf{R}_t)$. Nevertheless, taking the minimization ensures maximum generality and allows for example to deal with almost bipartite graphs where $\text{trace}(\mathbf{R}_s)$ can be oscillatory. The definitions above stands all for a given partition $\mathcal{P}$. As we are interested in the optimal clustering in the sense of stability, we search the partition that maximize the stability. Clearly, the optimal partition might be different for each Markov time t. Computing the optimal partition for each Markov time gives the *stability curve of the graph* :

$$r_t = \max_{\mathcal{P}} r_t(\mathcal{P}). \tag{11}$$

We understand thus how Markov time acts as an intrinsic resolution parameter: as Markov time grows, the number of communities is expected to decrease, since there are more possibilities for a random walker to escape a community when the time window increases. Hence, communities get bigger (or coarser) with Markov time increasing. Interestingly, one can prove that in the case of *undirected* networks, stability at time 1 is equivalent to the well-known *configuration modularity* measure. But this equivalence does not hold for *directed* networks and therefore does not concern the present work.

At this stage, an important remark has to be made about the assumption of ergodicity. The verification of this assumption is often far from being obvious, especially in the case of big undirected networks. The trick in that case is to introduce "à la Google" random teleportations.[1] Let τ be the *teleportation probability*. Then, if a random walker is located on a node with at least one outlink (which is always the case for the networks that we will consider), it follows one of the outlinks with probability $1-\tau$. Otherwise, the node is called a *dangling node* and the random walker is teleported with a uniform probability to another random node. The corresponding perturbation of the transition probability matrix is, in the most general case:

$$\widetilde{\mathbf{M}} = (1-\tau)\mathbf{M} + \frac{1}{N}[(1-\tau)\mathbf{d} + \tau\mathbf{1}]\mathbf{1}^{\mathsf{T}}, \tag{12}$$

where N is the number of nodes, $\mathbf{d}$ is a binary $N \times 1$ vector whose entries are equal to 1 if the corresponding node is a dangling node and 0 otherwise, and $\mathbf{1}$ is the $N \times 1$ unity vector. In the case that we will consider in the next section, $\mathbf{d}$ is the zero vector. This perturbation is known to make the dynamics ergodic, ensuring the existence and uniqueness of the stationary solution $\boldsymbol{\pi}$.

1.2 Extension to continuous time

From a general viewpoint, the discrete process can be interpreted as an approximation of its continuous counterpart : whereas the state of the discrete-time random walker can only change at unit-time intervals, the continuous-time random walkers undergoes a waiting time between each change of state which is itself a random variable. More precisely, it is a continuous memoryless random variable distributed exponentially. Obviously, the transition probabilities from one node to the other are the same for both discrete- and continuous-time processes, only the time at which the jump occurs may vary. The continuous-time process corresponding to (1) is governed

[1] In the original PageRank proposed by S. Brin and L. Page in 1998 (ref. [6]), this consist essentially in applying a perturbation to the transition probability matrix between web pages in order to ensure that at least one row of the matrix is positive, which implies the convergence of the Power Method. If we note the teleportation probability τ, the perturbation can be interpreted as follows: a web surfer follows a link in his current page with probability $1-\tau$ and jumps to an arbitrary web page with probability τ.

by the following dynamics :

$$\dot{\mathbf{p}} = \mathbf{p}\,\mathrm{diag}\,\{\boldsymbol{\lambda}(\mathbf{k})\}\,\mathbf{K}^{-1}\mathbf{A} - \mathbf{p}\,\mathrm{diag}\,\{\boldsymbol{\lambda}(\mathbf{k})\} = -\mathbf{pL}, \tag{13}$$

where $\lambda_i(\mathbf{k})$ is the rate at which random walkers leave node i, and $\mathbf{L} = \mathrm{diag}\{\boldsymbol{\lambda}(\mathbf{k})\}[-\mathbf{K}^{-1}\mathbf{A} + \mathbf{I}]$. Two particular cases of this process are implemented by the stability software and are thus examined here, depending on the choice of $\boldsymbol{\lambda}(\mathbf{k})$: the so-called *normalised Laplacian dynamics* and *standard (combinatorial) Laplacian dynamics*. Their names comes from the similarity that arise between $\mathbf{L}$ and the normalied/standard Laplacian matrix. Each of those two dynamics represent best different physical processes. The former correspond to the choice $\boldsymbol{\lambda}_{norm}(\mathbf{k}) = \mathbf{1}$. Hence, the expected waiting time is 1 at every node, and $\mathbf{L} = -\mathbf{K}^{-1}\mathbf{A} + \mathbf{I} = -\mathbf{M} + \mathbf{I}$. The latter corresponds to $\boldsymbol{\lambda}_{combi}(\mathbf{k}) = \mathbf{k}/\langle\mathbf{k}\rangle$. In that case, $\mathbf{L} = (-\mathbf{A} + \mathbf{K})/\langle\mathbf{k}\rangle$ and the average waiting time at node i is $\langle\mathbf{k}\rangle/k_i$. Hence, the expected waiting time at a given node is smaller (resp. larger) than 1 if the total weight of the outgoing edges from that node is larger (resp. smaller) than the average total weight of the outgoing edges on the network. However, the expected waiting time over the whole network is $\langle\langle\mathbf{k}\rangle/\mathbf{k}\rangle = 1$. The corresponding governing equations are respectively

$$\dot{\mathbf{p}} = \mathbf{p}\mathbf{K}^{-1}\mathbf{A} - \mathbf{p} = \mathbf{p}\mathbf{M} - \mathbf{p} \tag{14}$$

for the normalized Laplacian and

$$\dot{\mathbf{p}} = \mathbf{p}\frac{\mathbf{A}}{\langle\mathbf{k}\rangle} - \mathbf{p}\frac{\mathbf{K}}{\langle\mathbf{k}\rangle} \tag{15}$$

for the combinatorial Laplacian.

The clustered autocovariance matrix for partition $\mathcal{P}$ at time t is easily generalised to

$$\mathbf{R}(t;\mathcal{P}) = \mathbf{H}_{\mathcal{P}}^{\intercal}(\boldsymbol{\Pi}\mathbf{P}(t) - \boldsymbol{\pi}^{\intercal}\boldsymbol{\pi})\mathbf{H}_{\mathcal{P}}, \tag{16}$$

where $\mathbf{P}(t)$ is the the transition matrix of the process at time t: $\mathbf{P}(t) = \mathrm{e}^{-t\mathbf{L}}$. The continuous-time definition of the stability of a partition $\mathcal{P}$ follows almost straightforwardly :

$$r(t;\mathcal{P}) = \mathrm{trace}\,[\mathbf{R}(t;\mathcal{P})]\,. \tag{17}$$

Notice that it is not necessary to minimise over the time interval $[0, t]$: indeed, it can be shown that $\mathrm{trace}\,[\mathbf{R}(t;\mathcal{P})]$ is monotonically decreasing with time. The interpretation in terms of a random walk is similar to the discrete case : let $P(\mathcal{C}, t)$ be the probability that a random walker is in community $\mathcal{C}$ at time t if it was initially in $\mathcal{C}$, when the system is at stationarity. Discounting the probability of such an event to take place by chance at stationarity and summing over all communities of $\mathcal{P}$ leads to the definition of the stability of the partition $\mathcal{P}$:

$$r(t;\mathcal{P}) = \sum_{\mathcal{C}\in\mathcal{P}} P(\mathcal{C}, t) - P(\mathcal{C}, \infty). \tag{18}$$

By ergodicity, the memory of the initial condition is lost at infinity and $P(\mathcal{C}, \infty)$ is thus equal to the probability that two independant walkers are in $\mathcal{C}$ at stationarity. Equation (18) tells us that only the communities in which a random walker is likely to stay brings a positive contribution to stability, where *likely to stay* means that the probability for a walker to be in its initial community at time t is larger than the probability of that event occuring by chance at stationarity. The stability curve of the graph can now be expressed as a continuous function of t :

$$r(t) = \max_{\mathcal{P}} r(t;\mathcal{P}). \tag{19}$$

1.3 Assessing the robustness of a partition

We present here two mechanisms commonly used to assess the relevance of a particular partition. One simple way is to consider that a robust partition should not be alterred by a small modification of the quality function. Such a modification could be for example a perturbation of the Markov time t at which the partition has been found. From this point of view, robust partitions correspond to *plateaux* in the community curve of the graph. In other words, robust partitions should be persistent over a wide interval of Markov time.

The second indicator of the robustness of a partition that we will take into account in this work follows from considering that a robust partition is one that is persistent to small modifications of the optimization algorithm. The central tool to quantify this approach of the robustness of a partition is the *normalized variation of information* [7], which is a popular way to compare two partitions. Let $p(\mathcal{C})$ be the probability for a node to be in community $\mathcal{C}$, i.e. $p(\mathcal{C}) = n_{\mathcal{C}}/N$ where $n_{\mathcal{C}}$ is the number of nodes in community $\mathcal{C}$. The variation of information between partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ is defined by :

$$\mathrm{VI}(\mathcal{P}_1, \mathcal{P}_2) := \frac{H(\mathcal{P}_1, \mathcal{P}_2) - H(\mathcal{P}_1) - H(\mathcal{P}_2)}{\log(N)} = \frac{H(\mathcal{P}_1|\mathcal{P}_2) + H(\mathcal{P}_2|\mathcal{P}_1)}{\log(N)}, \tag{20}$$

where $\log(N)$ is a normalization factor; $H(\mathcal{P}) = -\sum_{\mathcal{C}} p(\mathcal{C}) \log[p(\mathcal{C})]$ is the Shannon entropy; $H(\mathcal{P}_1, \mathcal{P}_2)$ is the Shannon entropy of the joint probability $p(\mathcal{C}_1, \mathcal{C}_2)$ that a node belongs to both a community $\mathcal{C}_1$ of $\mathcal{P}_1$ and a community $\mathcal{C}_2$ of $\mathcal{P}_2$. This yields $p(\mathcal{C}_1, \mathcal{C}_2) = n_{\mathcal{C}_1 \cap \mathcal{C}_2}/N$, and $H(\mathcal{P}_1, \mathcal{P}_2) = -\sum_{\mathcal{C}_1 \in \mathcal{P}_1} \sum_{\mathcal{C}_2 \in \mathcal{P}_2} p(\mathcal{C}_1, \mathcal{C}_2) \log[p(\mathcal{C}_1, \mathcal{C}_2)]$; and $H(\mathcal{P}_1|\mathcal{P}_2)$ is the conditional Shannon entropy of partition $\mathcal{P}_1$ given $\mathcal{P}_2$, which is defined in a standard way from the joint distribution: $p(\mathcal{C}_1|\mathcal{C}_2) = p(\mathcal{C}_1, \mathcal{C}_2)/p(\mathcal{C}_2) = n_{\mathcal{C}_1 \cap \mathcal{C}_2}/n_{\mathcal{C}_2}$, and the expression of $H(\mathcal{P}_1|\mathcal{P}_2)$ follows straightforwardly. The latter can be interpreted as the additional information needed to describe $\mathcal{P}_1$ once $\mathcal{P}_2$ is known. This measure of the difference bewteen two partition is then used as follows: for each Markov time, an ensemble of Louvain optimisations of stability are performed, starting from different random initial node ordering. Remember that the problem being $\mathcal{NP}$-hard, we rely on a heuristic algorithm — the Louvain method — that finds a good partition for a given Markov time, but not necessarily the optimal partition. Hence the partition found may differ if a different initial condition is provided. The normalized variation of information allows then to quantify how different the optimised partitions are. Therefore, a low variation of information indicates optimised partitions that are very similar to each others, hence that a small modification of the algorithm barely alter the partition. From the point of view of the field of dynamical system, robust partitions have thus an attractor with a large basin of attraction for the optimisation method.

2 Clustering of the overturner problem

This section presents the results of applying a stability-based community detection algorithm on the overturner problem. First, we explain how the method is applied and then we present the results for a given set of data's.

2.1 Description of the method

In order to apply a clustering algorithm on the overturner problem, we have to define how the model can be considered as a graph. To this end, the domain is decomposed into $n_{box,\,y} \times n_{box,\,z}$ boxes. We note $N_{box} = n_{box,\,y} n_{box,\,z}$ the total number of boxes. Figure 1 represents an example

of such a domain decomposition with $n_{box,\,y} = 15$ and $n_{box,\,z} = 10$. For any time T, the corresponding directed graph is build as follows : each node represents a box, and the weight of the edge between nodes i and j is the probability $m_{ij}(T)$ that a particle ends up in box j after a time T if it was initially in box i. If $m_{ij}(T) = 0$, one can equivalently consider that there is no edge between nodes i and j. Since the problem is stationnary, $m_{ij}(T)$ depends only on the elapsed time T, not on the initial time. Hence, the initial time can indifferently be considered as being zero. The adjacency matrix $\mathbf{M}(T)$ of the graph is build from the weights $m_{ij}(T)$: $[\mathbf{M}(T)]_{ij} = m_{ij}(T)$. For any time T, $\mathbf{M}(T)$ is row-stochastic, i.e. $\mathbf{M}(T)\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the N_{box}-dimensional unit column vector. The latter has a straightforward physical interpretation: every particle remains in the domain.

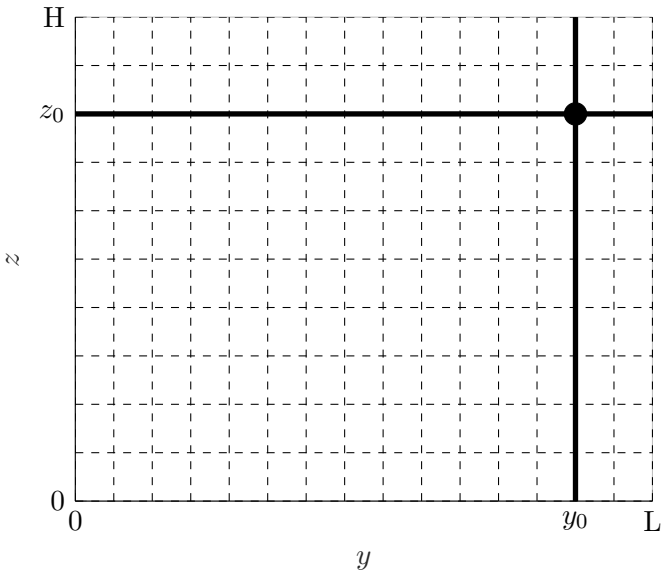


Figure 1 – Illustration of the decomposition of the domain into boxes with $n_{box,\,y} = 15$ and $n_{box,\,z} = 10$.

To estimate the probabilities $m_{ij}(T)$, the program is runned for a time T with each box containing initially J uniformly distributed particles. $m_{ij}(T)$ is then numerically estimated as the number of particles having started in box i and ending up in box j, divided by J. This *box counting* method has been extensively used to estimate the concentration in studies using random walk modeling, see e.g. [8]. Nevertheless, this method suffers some drawbacks; the most important of them are pointed in [9], but we recall them here for the sake of completeness. The estimated transition probability depends on the choice of the boxes, in particular of their size and their center. Moreover, the number of boxes cannot be chosen to be too large; otherwise the estimated concentration tends to become very irregular or noisy. Finally, the resolution of the estimated concentration is limited to the size of the boxes, as it cannot be described in a box more precisely than a constant. But it is the perfect method for our problem since the volume average over such boxes (the nodes) is precisely what we want. Note however that other methods exist for estimating the concentration, that might be better suited for other studies. For example, the *kernel estimation* method allows to reduce drastically the number of particles, and does not suffer from the resolution limit inherent to the box counting method. This method is briefly presented in [9]. Classical references are [10] and [11].

2.2 Use of the stability software

We present here briefly how the *PartitionStability* software is used to compute the partitions. Every concept appearing here has been presented in section 1. The `stability` function is simply

called as follows :

```
[S,N,VI,C] = stability(M,Markov_T,'directed','plot','teleport',0.01);
```

Here, `M` is the matrix $\mathbf{M}(T)$ at the desired time T; `Markov_T` is the vector containing every Markov times at which the optimal stability partition has to be computed (ideally, the sampling should be exponential); the `'directed'` option specifies that we consider a directed graph; `'plot'` asks the program to plot the stability, number of communities and variation of information as a function of the Markov time; and `'teleport',0.01` allows to specify the value of the teleportation probability τ to 0.01, the default value being 0.15. The choice of 0.01 is motivated by the fact that we believe the graph to be ergodic, even if we cannot prove it. Note that the program allows to choose which type of laplacian should be used to calculate the stability. However, the question does not arise here since both laplacians are equivalent in our case. Indeed, the total outgoing weight is the same at every node and is precisely equal to the number of particles J released in each box. Hence, $k_i = J$ for every node i and $\langle \mathbf{k} \rangle = J$, so that $\boldsymbol{\lambda}_{combi}(\mathbf{k}) = \mathbf{k}/\langle \mathbf{k} \rangle = \mathbf{1} = \boldsymbol{\lambda}_{norm}(\mathbf{k})$. We let thus the program run with the default normalized Laplacian, since it does not make any difference in our case. The output arguments `S`, `N`, `VI` and `C` contain respectively the stability, the number of communities, the variation of information, and the optimal partition for each Markov time contained in `Markov_T`. If the latter is of size n, then `S`, `N` and `VI` are n-dimensional vectors and `C` is a $N_{box} \times n$ matrix. At the jth Markov time, communities are labeled by consecutive integers between 0 and `N(j)`-1 such that `C(i,j)` $= k$ means that node i belongs to community k at Markov time `Markov_T(j)`.

2.3 Results

Now we present some results on a particular discretization of the overturner problem. The box decomposition of the domain is the one shown in figure 1, and $J = 10\,000$ particles are released in each box. More precesily, a box is decomposed into a 100×100 subgrid, and one particle is initially located at every point of the subgrid, so that the particles are initially uniformly distributed within the box. The transition probability matrices $\mathbf{M}(T)$ are generated for different values of T. We show here the results for $T = 1$, 10, 50 and 100 years. The vector of the Markov times `Markov_T` is sampled exponentially from 0.1 to 100 : $\log_{10}$(`Markov_T`) $= [-1, -0.98, \ldots, 1.98, 2]$. Notice that the physical meaning of the Markov time changes with T : a Markov time step of 1 is equal to a physical time step of T. Hence, for $a > 0$, if for $\mathbf{M}(T)$ we find some communities in the range of Markov times $[t_{M_1}, t_{M_2}]$, then for $\mathbf{M}(aT)$ we expect to find similar communities in the range of Markov times $\frac{1}{a}[t_{M_1}, t_{M_2}]$.

Figures 2, 4, 6 and 8 show the stability curves, the number of communities and the variation of information as functions of the Markov time for $T = 1$, 10, 50 and 100 years respectively. As discussed in section 1.3, robust partitions correspond to plateaux in the community curve of the graphs. By using this criterion, partitions of 6, 5, 4, 3 and 2 communities are found at different time scales. Those partitions are summarized in table 1 along with the physical time range at which they reveal themselves. Figure 3, 5, 7 and 9 shows the most robust clusterings for $T = 1$, 10, 50 and 100 respectively. From table 1, we observe that some similar partitions happen to be the most relevant at different time scales when we modify T. For example, for $T = 1$, 6 communities are found in the time range 9 - 12 years (figure 3a). A similar clustering is found for $T = 10$ and $T = 50$ but in the time ranges 24 - 48 years and 36-48 years respectively (figures 5a and 7a).

It is important to notice that the community detection algorithm may fail to detect the right number of communities. Take figure 5d for example : the stability software detects 2 communities. However, the white community consists of two noncontiguous blocks. Intuitively,

particles leaving the lower white block should enter the khaki block first before entering the upper white block. Hence, there should be 3 communities rather than 2 for this partitioning. A way to quantify this intuition is by looking at

$$\mathbf{M}_{\mathcal{P}}(T) = \mathrm{diag}^{-1}(\mathbf{n})\mathbf{H}_{\mathcal{P}}^{\intercal}\mathbf{M}(T)\mathbf{H}_{\mathcal{P}}, \tag{21}$$

where $\mathbf{n}$ is the c-dimensional vector containing the number of blocks in each community. $[\mathbf{M}_{\mathcal{P}}]_{kl}$ is the transition probability from community k to community l. By considering the clustering where the lower and the upper white blocks are separated communities, we get

$$\mathbf{M}_{\mathcal{P}}(10) = \begin{pmatrix} 0.886 & 0.114 & 0.000 \\ 0.052 & 0.895 & 0.053 \\ 0.017 & 0.256 & 0.727 \end{pmatrix}. \tag{22}$$

Here, community 1 is the lower white block, community 2 is the khaki block and community 3 is the upper white block. We observe that $[\mathbf{M}_{\mathcal{P}}]_{13} = 0$ and $[\mathbf{M}_{\mathcal{P}}]_{31} = 0.017$, indicationg very weak links between the lower and the upper white blocks. Hence, they should indeed be considered as separated communities. However, this does not mean that 5d provides then the optimal clustering with 3 communities ! Such a clustering is rather given by figure 5c, and the clustering proposed in figure 5d should simply be disregareded as being nonrelevant.

Now, let us analyze a seemingly relevant community structure. By looking at table 1 together with figures 5b and 7b, we observe that two similar 5-communities clusterings arise in the time range 50 - 63 years when $T = 10$ and $T = 50$. This indicates that those community structures might be more resilient than others. There is only a 2 boxes difference between the two clusterings; we will therefore focus on the clustering found for $T = 10$, namely the one from figure 5b. The communities are numbered from 1 to 5 on the figure. The matrix $\mathbf{M}_{\mathcal{P}}$ for this community structure is

$$\mathbf{M}_{\mathcal{P}}(10) = \begin{pmatrix} 0.907 & 0.024 & 0 & 0.024 & 0.045 \\ 0.073 & 0.827 & 0.043 & 0.057 & 0 \\ 0 & 0.029 & 0.925 & 0.036 & 0.010 \\ 0.039 & 0.041 & 0.089 & 0.776 & 0.055 \\ 0.020 & 0 & 0.048 & 0.022 & 0.910 \end{pmatrix}. \tag{23}$$

Obviously, particles in a community tends to stay in that community. But what are the main interconnections between communities ? By looking at matrix $\mathbf{M}_{\mathcal{P}}$, we observe that particles leaving community 1 goes preferentially to community 5; from community 2, the main tendency is to go to community 1; from 3 to 4 and 2; from 4 to 3 (mainly because of the size of 3) and from 5 to 3. Hence, the dominant tendency is that the particles tend to describes a clockwise cycle in the domain, which is exactly the expected behavior.

J'aimerais aller un peu plus loin dans mes commentaires mais les idées ne se bousculent pas... Et les commentaires que je fais ci-dessus ne nous apprennent rien. Cela fait sans doute beaucoup d'images d'images pour au final pas grand chose.

Table 1 – Summary of the dominant clusterings found by inspection of the transition probability matrix $\mathbf{M}(T)$ for $T = 1$, 10, 50 and 100 years.

T	Time range [year]				
	6 communities	5 communities	4 communities	3 communities	2 communities
1	9 - 12	15 - 26	28 - 36	38 - ...	
10	24 - 48	50 - 63		91 - 316	331 - ...
50	36 - 48	50 - 66		69 - 138	144 - 190
100	58 - 76	79 - 105		120 - 229	240 - 316

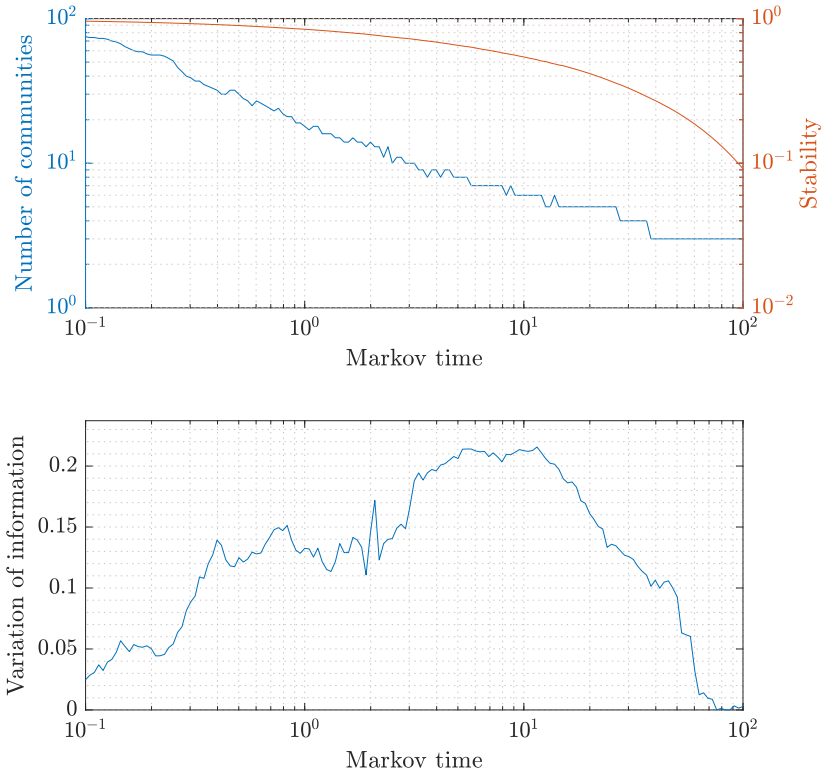
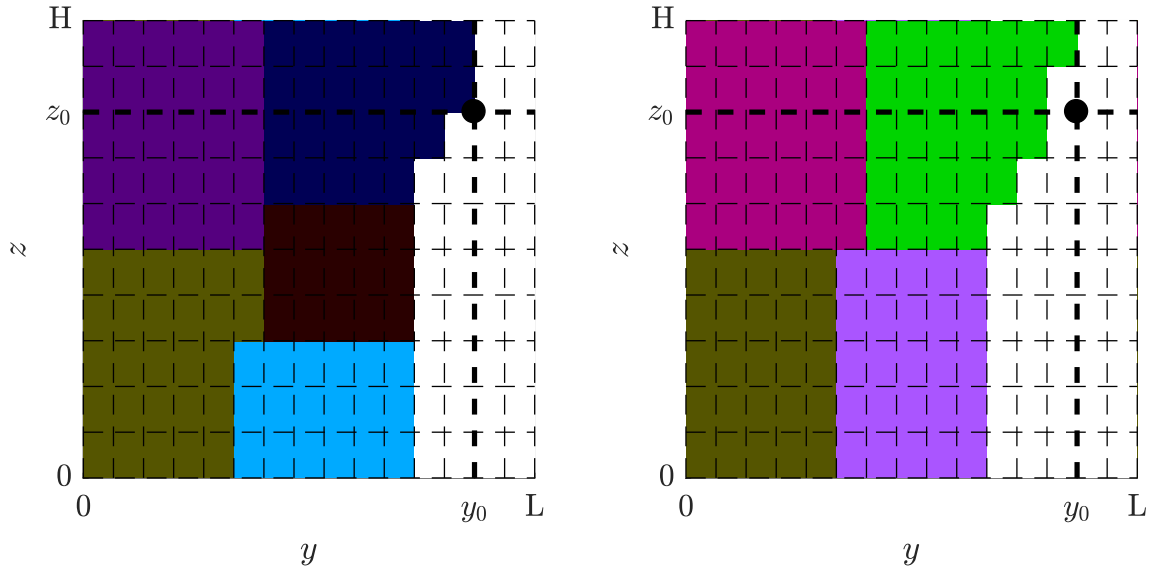
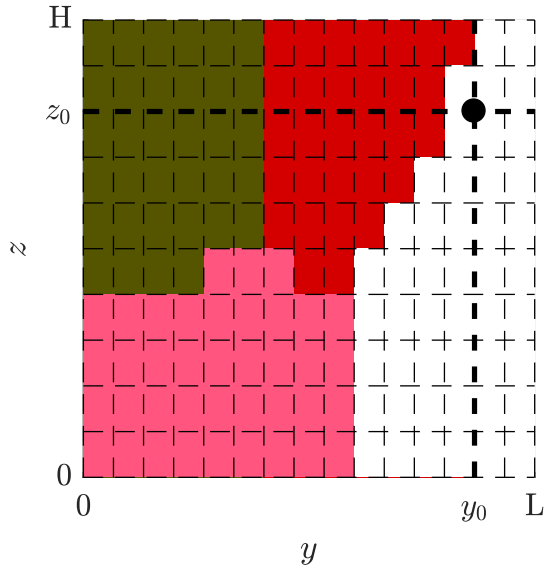


Figure 2 – Stability, number of communities and variation of information as a function of the Markov time for $T = 1$ year.

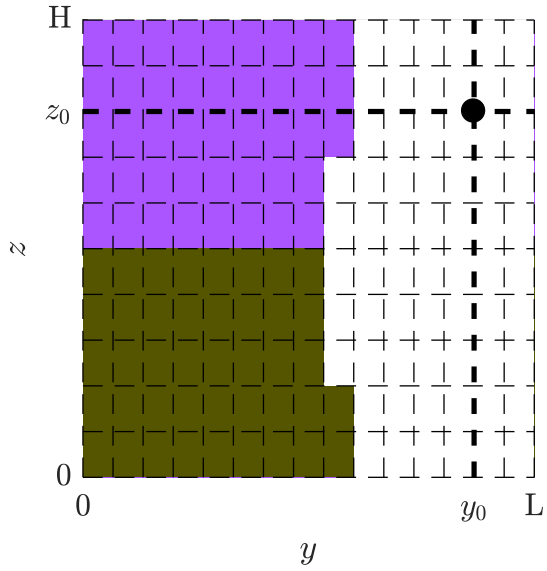


(a) 6 communities.

(b) 5 communities.



(c) 4 communities.



(d) 3 communities.

Figure 3 – The relevant clusterings detected at different time scales for $T = 1$ year.

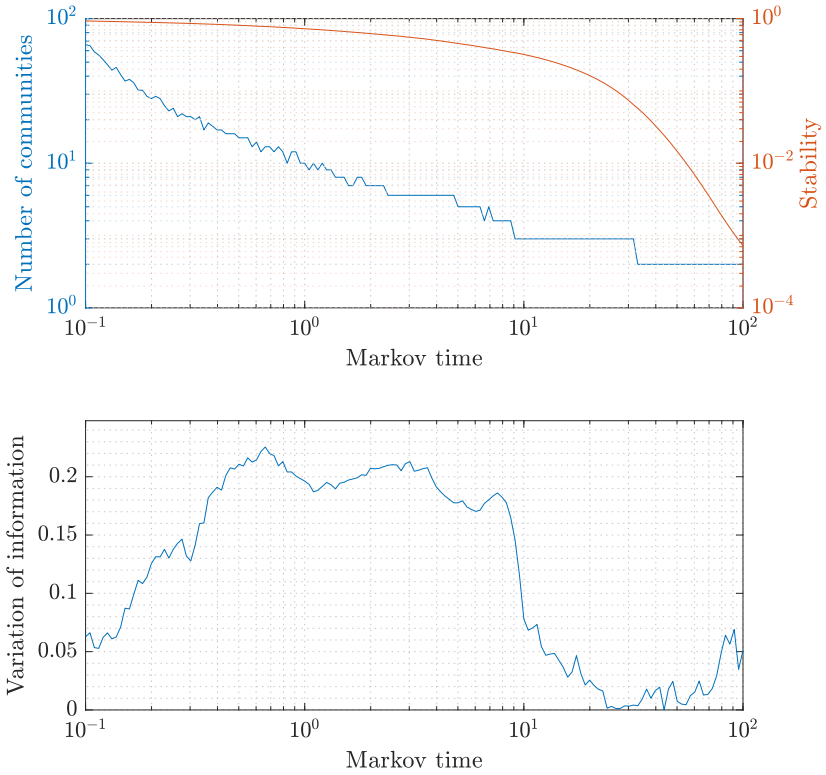
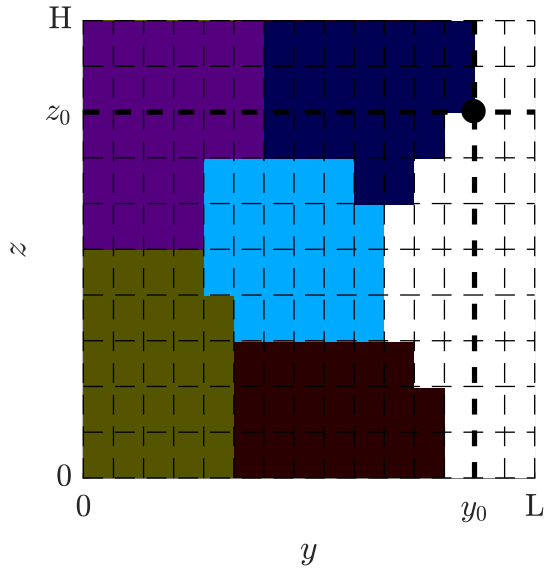
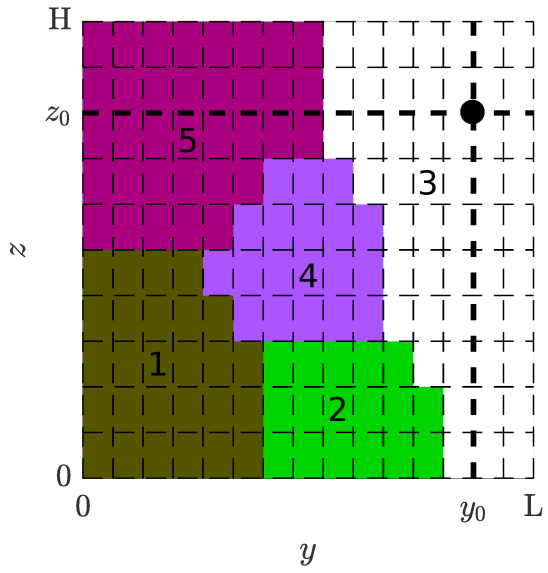


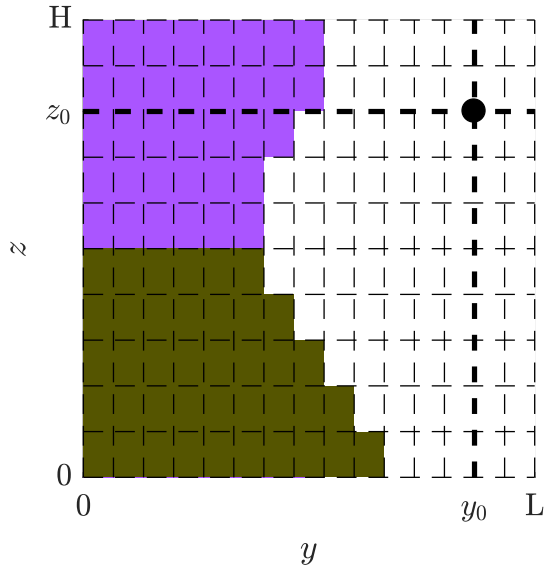
Figure 4 – Stability, number of communities and variation of information as a function of the Markov time for $T = 10$ years.



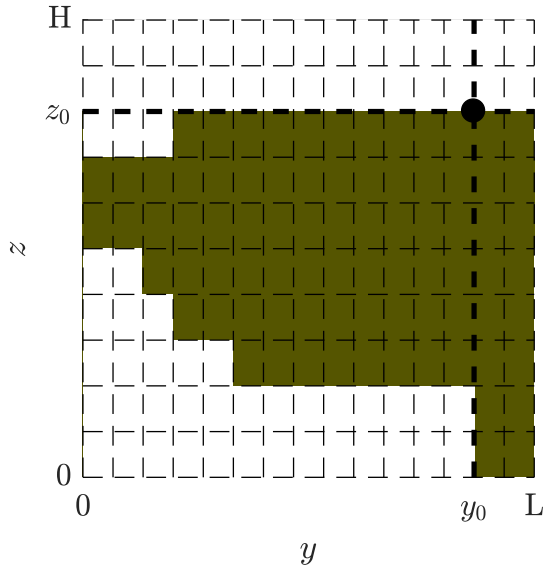
(a) 6 communities.



(b) 5 communities.



(c) 3 communities.



(d) 2 communities detected by the algorithm which should rather be considered as being 3 communities.

Figure 5 – The relevant clusterings detected at different time scales for $T = 10$ years.

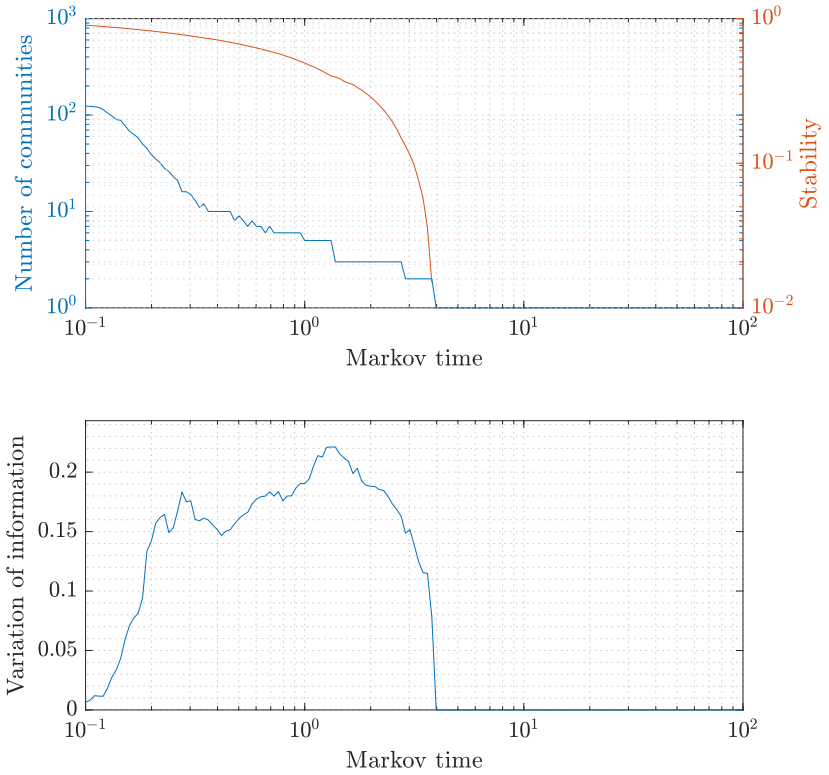
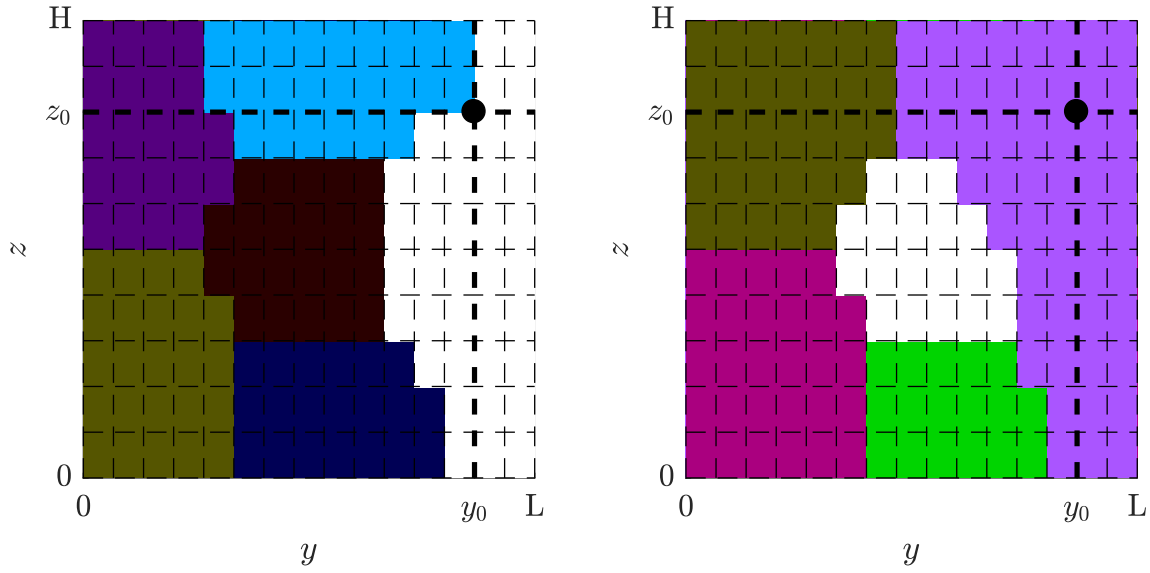
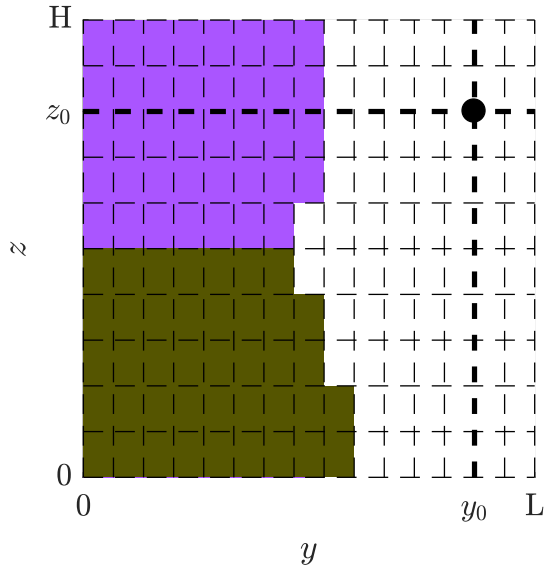


Figure 6 – Stability, number of communities and variation of information as a function of the Markov time for $T = 50$ years.

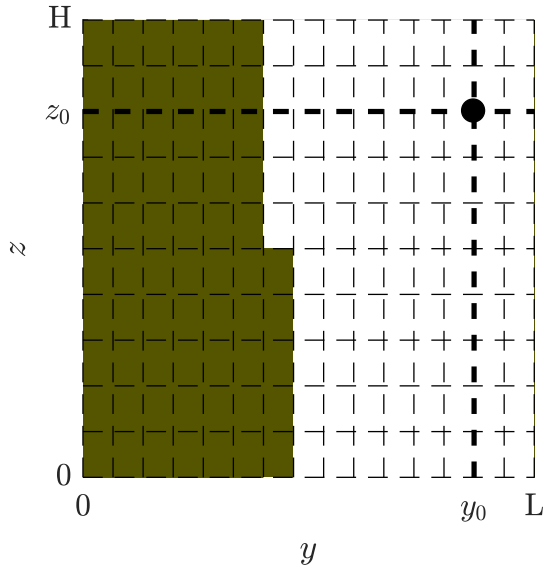


(a) 6 communities.

(b) 5 communities.



(c) 3 communities.



(d) 2 communities.

Figure 7 – The relevant clusterings detected at different time scales for $T = 50$ years.

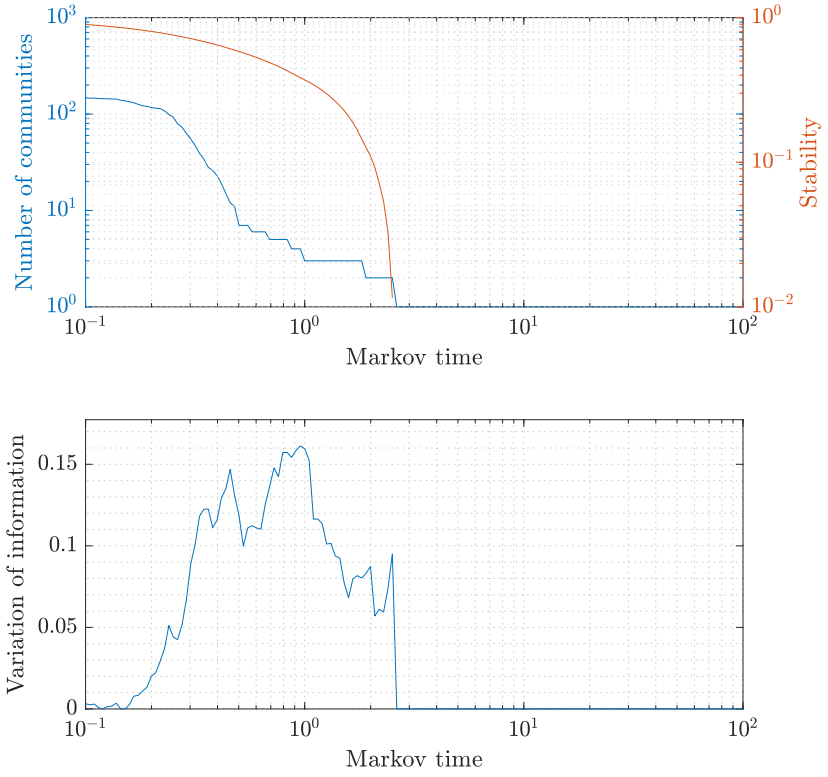
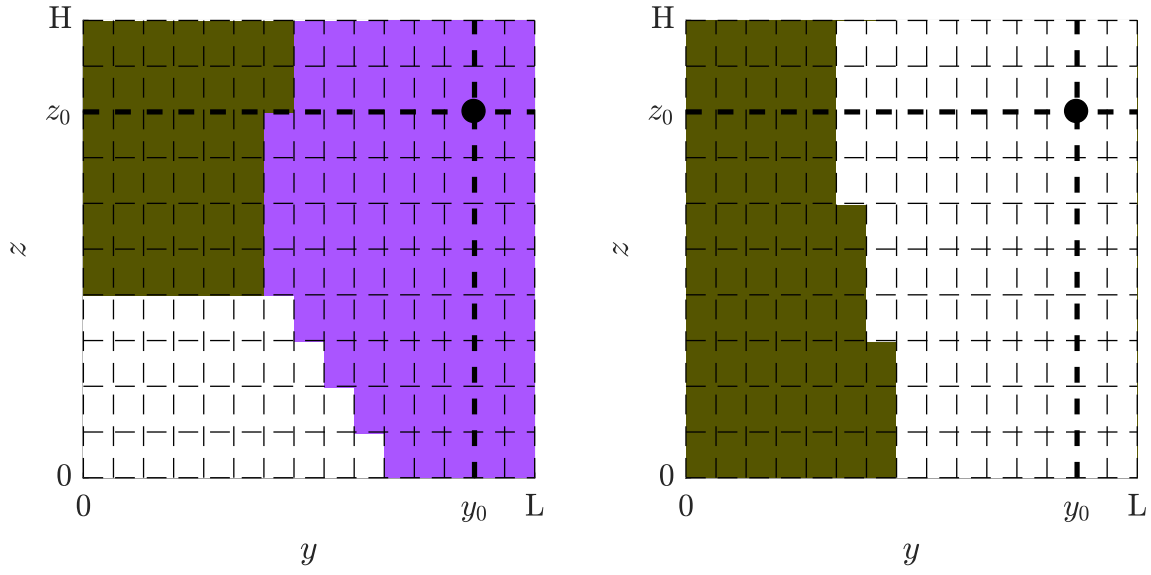


Figure 8 – Stability, number of communities and variation of information as a function of the Markov time for $T = 100$ years.



(a) 3 communities. **(b)** 2 communities.

Figure 9 – The relevant clusterings detected at different time scales for $T = 100$ years.

References

[1] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[2] J-C Delvenne, Sophia N Yaliraki, and Mauricio Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760, 2010.

[3] Renaud Lambiotte, J-C Delvenne, and Mauricio Barahona. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*, 2009.

[4] Jean-Charles Delvenne, Michael T Schaub, Sophia N Yaliraki, and Mauricio Barahona. The stability of a graph partition: A dynamics-based framework for community detection. In *Dynamics On and Of Complex Networks, Volume 2*, pages 221–242. Springer, 2013.

[5] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[6] Sergey Grin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[7] Marina Meilă. Comparing clusterings – an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[8] AM Riddle. The specification of mixing in random walk models for dispersion in the sea. *Continental Shelf Research*, 18(2):441–456, 1998.

[9] Darya Spivakovskaya, Arnold W Heemink, and Eric Deleersnijder. Lagrangian modelling of multi-dimensional advection-diffusion with space-varying diffusivities: theory and idealized test cases. *Ocean Dynamics*, 57(3):189–203, 2007.

[10] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[11] MP Wand and MC Jones. Kernel smoothing. 1995. *Chapman&Hall, London*, 1995.

[12] Eric Deleersnijder. Test cases for advection-diffusion equations with a first-order decay term. 2011. Available at `http://hdl.handle.net/2078.1/155372`.

[13] Renaud Lambiotte, Roberta Sinatra, J-C Delvenne, Tim S Evans, Mauricio Barahona, and Vito Latora. Flow graphs: Interweaving dynamics and structure. *Physical Review E*, 84(1):017102, 2011.