École polytechnique de Louvain (EPL)

Towards the automatic design of compartment models for marine transport processes

Dissertation presented by
Renaud DUFAYS

for obtaining the Master's degree in
Mathematical Engineering

Supervisors
Eric DELEERSNIJDER, Jean-Charles DELVENNE

Reader
Laurent DELANNAY

Academic year 2016-2017

Abstract

In the context of solving marine problems, compartment models act as a complementary tool to the usual discretization method such as finite difference or finite element method: although they provide less accurate results, they allows for easier interpretation and faster simulation, hence providing the possibility for long-term model runs. However, the compartment model approach suffers from the fact that there is nowadays no automatic procedure to delineate the compartments and express the fluxes between them. An method towards automatic delineation of the compartments. It is based on a community detection method from networks theory, the stability method. The whole procedure is applied on a simple problem, showing satisfying results. Obviously, the problem considered is too simple to draw conclusions on the method proposed, but the encouraging result in this work shows that further work is worthwhile.

Alongside the presentations of the method and of the stability criterion for community detection, a whole chapter is dedicated to the theory of stochastic differential equations, leading to a Lagrangian numerical method for solving a two dimensional advection-diffusion problem. Besides, a generic compartment model is build from the reactive transport equation, and its fundamental properties are shown. The only theoretical background needed to understand this work is to be familiar with the reactive transport equation. All the other theoretical tools are introduced and proved before they are used, making this work a standalone document.

Acknowledgments

Many thanks the professors Éric Deleersnijder and Jean-Charles Delvenne for having accepted to be my supervisors despite unusual conditions. Thank you both for your time and your insightful advices.

I am grateful to Laurent Delannay for having accepted to be the reader of this master's thesis.

Thank you to my parents, for their support during my whole academic path.

Table of Contents

List of Symbols

Unless otherwise stated, bold lower cases indicate column vectors ($\mathbf{x}$) and bold upper cases indicate matrices ($\mathbf{X}$). The entries of a vector are denoted by a single subscript (x_i) and the entries of a matrix are denoted by brackets with a double subscript ($[\mathbf{X}]_{i,j}$). The transposed sign is $^\intercal$.

As a general rule, subscript t indicates a discrete-time quantity evaluated at time t, while parenthesis indicate a continuous-time quantity.

The time derivative of a quantity x is denoted by the usual convention $\partial x/\partial t$, or by $\dot{x}$.

Clustering

n	Number of nodes of the graph.
$\mathbf{A}$	Weighted adjacency matrix.
q_i	Total outgoing weight from node i ($i \in \{1, \ldots, n\}$).
$\mathbf{q}$	Vector of the q_i's.
$\mathbf{Q}$	Matrix with the total outgoing weights on its diagonal: $\mathbf{Q} = \mathrm{diag}(\mathbf{q})$.
$\mathbf{M}$	Transition probability matrix for the discrete-time Markov process.
$\mathbf{P}(t)$	Transition matrix for the continuous-time Markov process evaluated at time t.
$p_{t,i}$	Probability to be in node i at Markov time t ($i \in \{1, \ldots, n\}$).
$\mathbf{p}_t$	Discrete-time probability vector: $\mathbf{p}_t = [p_{t,1}, \ldots, p_{t,n}]$.
$\mathbf{p}(t)$	Continuous-time probability vector.
$\boldsymbol{\pi}$	Stationary distribution vector.
$\boldsymbol{\Pi}$	Matrix with the stationary distributions on its diagonal: $\boldsymbol{\Pi} = \mathrm{diag}(\boldsymbol{\pi})$.
$\mathbf{x}_t$	Random indicator vector at time t.
$\mathbf{C}(\mathbf{x}_{t_0}, \mathbf{x}_{t_0+t})$	Autocovariance matrix of $\mathbf{x}$.
$\mathcal{P}$	A partition.
c	Number of communities.
$\mathcal{C}_i$	The ith community of the partition ($i \in \{1, \ldots, c\}$).
$\mathbf{H}_{\mathcal{P}}$	Indicator matrix of $\mathcal{P}$.
$\mathcal{H}_{\mathcal{P}}(\cdot)$	Operator $\mathbf{H}_{\mathcal{P}}^\intercal(\cdot)\mathbf{H}_{\mathcal{P}}$.
$\mathbf{y}_t$	Community indicator vector at time t.
$\mathbf{R}_t(\mathcal{P})$	Discrete-time clustered autocovariance matrix for partition $\mathcal{P}$ at time t.
$\mathbf{R}(t;\mathcal{P})$	Continuous-time clustered autocovariance matrix for partition $\mathcal{P}$ at time t.
$r_t(\mathcal{P})$	Discrete-time stability of partition $\mathcal{P}$.
$r(t;\mathcal{P})$	Continuous-time stability of partition $\mathcal{P}$.
r_t	Discrete-time stability curve.
$r(t)$	Continuous-time stability curve.
τ	Teleportation probability.
$\lambda_i(\mathbf{q})$	Rate at which continuous-time random walkers leave node i ($i \in \{1, \ldots, c\}$).
$\boldsymbol{\lambda}(\mathbf{q})$	Vector of the $\lambda_i(\mathbf{q})$'s.
$\mathbf{L}$	Laplacian matrix.
$\mathrm{VI}(\mathcal{P}_1, \mathcal{P}_2)$	Variation of information between partitions $\mathcal{P}_1$ and $\mathcal{P}_2$.

Transport model

Adimensional variables are denoted by primes.

C	Concentration function.
$C(\mathbf{x}, t)$	Concentration function with its dependence on position and time written explicitly.
q	Source/sink term, or reactive term.
$\mathbf{u}$	Velocity vector.
$\mathbf{K}$	Diffusivity tensor (symmetric and positive-definite).
$\mathbf{x}$	Position vector. In two dimensions, $\mathbf{x} = (y, z)$.
ρ	Density of the mixture (seawater).
ρ_w	Density of pure water.
Ω	Domain of the problem (time independent).
$\partial\Omega$	Boundary of Ω.
$\lvert\Omega\rvert$	Volume (area in two dimensions) of Ω.
$\hat{\mathbf{n}}$	Outward unit normal vector.
$[\cdot]_*$	Quantity $\cdot$ evaluated at $*$.
$\bar{C}(t)$	Mean concentration over Ω that possibly depends on t.
$\bar{C}$	Mean concentration over Ω, shown to be constant.
$\hat{C}(\mathbf{x}, t)$	Deviation of the concentration with respect to $\bar{C}$.
$\sigma^2(t)$	Variance of the concentration function.

Compartment model

In what follows, $i, j \in \{1, \ldots, N\}$.

N	Number of subdomains (or compartments).
Ω_i	Subdomain i.
$\lvert\Omega_i\rvert$	Volume (area in two dimensions) of Ω_i.
$\Gamma_{i,j}$	Interface between Ω_i and Ω_j.
$\Gamma_{i,e}$	Interface between Ω_i and the environment.
$\lvert\Gamma_{i,j}\rvert$	Area (length in two dimensions) of $\Gamma_{i,j}$.
$C_i(t)$	Concentration function for compartment i.
$q_i(t)$	Net production rate over Ω_i.
$U_{i,j}$	Advective "flux" from compartment i to compartment j.
$V_{i,j}$	Diffusive "flux" from compartment i to compartment j.
$\sigma^2_{box}(t)$	Box variance.
$\mathbf{c}$	Vector of the compartment's concentrations: $\mathbf{c} = [C_1(t), \ldots, C_N(t)]^\intercal$.
$\mathbf{q}$	Vector of the q_i's: $\mathbf{q} = [q_1(t), \ldots, q_N(t)]^\intercal$.
$\boldsymbol{\omega}$	Vector of the $\lvert\Omega_i\rvert$'s: $\boldsymbol{\omega} = [\lvert\Omega_1\rvert, \ldots, \lvert\Omega_N\rvert]^\intercal$.
$\boldsymbol{\Omega}$	Matrix with $\boldsymbol{\omega}$ on its diagonal: $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\omega})$.
$\mathbf{A}$	Interaction matrix.
$\mathbf{A}_{\Delta t}$	Discrete interaction matrix for time step Δt.
$P_i(t)$	Number of tracer's particles in compartment i at time t.
P	Mass of one tracer's particle.

SDEs

$\delta(t)$	Dirac delta function.
$\xi(t)$	White noise.
$W(t)$	Wiener process.
$(\mathbf{I})$	Indicate an Itô SDE or integral.
$(\mathbf{bI})$	Indicate a backward Itô SDE or integral.
$\langle\cdot\rangle$	Expectation.
p	Probability density function.
$p(x,t;y,s)$	Probability density function with its dependence on position and time, and the initial condition written explicitly.
X_i	Numerical approximation of $x(t_i)$.
ΔW_i	$W(t_i) - W(t_{i-1})$.
Δt_i	$t_i - t_{i-1}$.

From clusters to compartments

$n_{cell,\,y}$	Number of grid cells in the y-direction.
$n_{cell,\,z}$	Number of grid cells in the z-direction.
N_{cell}	Total number of grid cells.
$m_{i,j}(T)$	Probability that a particle ends up in grid cell j after a time T if it was initially in grid cell i.
$\mathbf{M}(T)$	Adjacency matrix of the graph at time T, which is also the transition probability matrix.
$P_{i\to j}(T)$	Number of particles in grid cell j at time T, which were initially in grid cell i.
P_0	Number of particles released in each grid cell.

Introduction

The understanding of geophysical and environmental fluid flows has shown an increasing interest over the past decades. Climate change, and the related issue of pollution have raised the need for accurate simulations allowing to understand and predict the time-space evolution of the concentration of pollutants in the environment. In this regard, the fate of constituents dissolved in a fluid mixture is commonly modelled by means of reactive-transport equations. Those are coupled partial differential equations taking into account the influence of *reactions* as well as of the advective and diffusive *transport phenomena* on the concentration of the constituent in the mixture. Accurate algorithms have been developed by engineers to numerically solve such equations. Finite differences, finite volume and finite elements methods are amongst the most commonly used discretization methods. However, they rely on grids containing thousands and in some instances millions of grid points so that CPU time can become prohibitive in cases where long time simulations are needed. Furthermore, although those methods provide pretty accurate results, these often suffer from a lack of *interpretation.* This raises the need for simpler models with a few number of variables that would allow for an easier interpretation of the results as well as for fast long time simulations. Obviously, one cannot expect such models to provide results that would be as accurate as the ones furnished by the previously mentioned methods. Hence, both approaches are complementary.

One way to build such coarser models is to partition the domain of interest into a relatively small number of subdomains, called *compartments* or *boxes* over which the state variables are assumed *homogeneous.* The fluxes between the compartments are then expressed by simple laws, leading to models that are relatively easy to interpret. This procedure leads to models called *compartment models* or equivalently *box models.* The number of subdomains and their shapes vary widely from one problem to another. Some schematic representations of compartment models found in the literature are shown in figure 1.

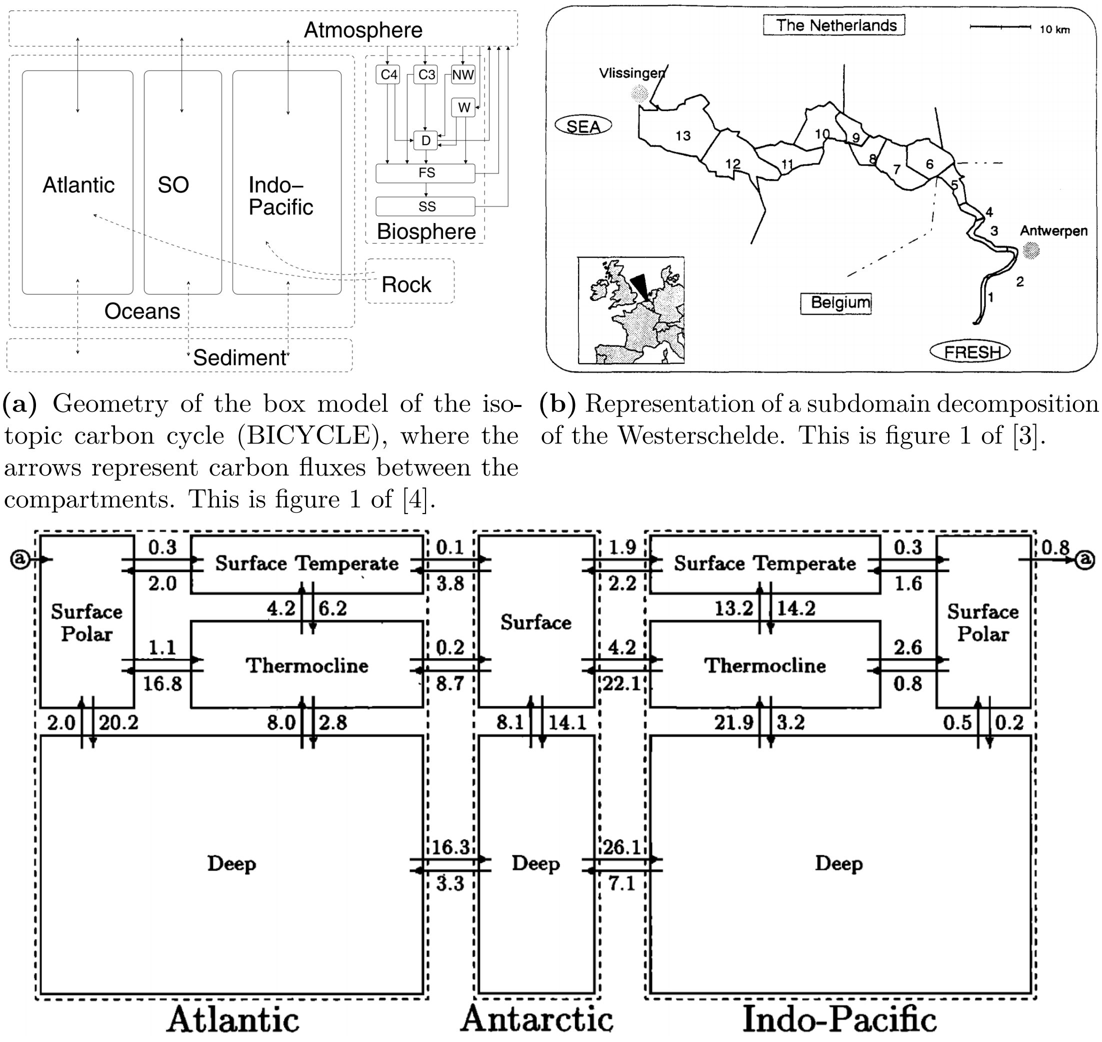
Box models have been widely used in studies involving marine systems as well at local scale [1–3] as at global scale [4, 5]. The choice of the subdomains and the specification of the fluxes exchanged between them rely on ad-hoc or empirical methods, based on the known hydrodynamics over the domain considered. It seems that no automatic procedure exists to define a relevant partitioning of the domain. The goal of this work is thus to fill that gap by proposing a method based on the tools of *network science* to automatically delineate relevant compartments.

Network science may be defined as the study of graphs (or networks) and their use to model various (real-life or not) problems. Informally, a graph is a mathematical representation of a set of objects (the *nodes* or *vertices* of the graph), and the links between pairs of them (the *edges* of the graph). The field of application of network science is extremely large, as illustrated by this quote from Newman [6]: "*Many objects* *of interest in the physical, biological, and social sciences can be thought of as networks and [...] thinking of them in this way can often lead to new and useful insights.*"

Often, graphs exhibit a *community structure*: it is the case if the vertices can be organized into groups such that there are many interactions between the nodes within a group, and few interaction between the nodes of different groups. Such groups of vertices are called *communities* or *clusters*. An example of a three communities partitioning of a simple graph is shown in figure 2. The art of revealing the community structure of a graph has a long history in network science, and a variety of *community detection algorithms* and *heuristics* has been developed, see [7] for a 2010 survey. Clustering methods have shown to be useful for a wide applications such as social and biological networks [8], biochemical networks [9–11], or informations networks such as the world wide web [12]. What makes communities particularly appealing is that they often correspond to *functional units* such as cycles or pathways in metabolic networks [10, 11, 13] or collections of pages on a single topic on the web [12]. In the context of geophysical modeling, clustering methods have been used in works providing suggestions for the optimal placement of marine protected areas [14–16]. In this work, we aim to show that communities may correspond to relevant compartments in the case of advection-diffusion flow networks. To this end, we focus on one relatively recent method for community detection based on the *stability* measure [17–19].

A complete procedure is proposed to build a compartment model from any problem whose velocity field and diffusivity tensor are known. First, the Lagrangian equations describing individual particles trajectories are derived from the transport model, and consistent numerical methods are proposed. This allows to compute the transition probability matrix at the desired times, which is the information needed to apply the stability clustering method. Finally, subdomains are delineated from the communities found by the clustering algorithm, and the exchange coefficients between the compartments are estimated numerically, completing the construction of the box model. This approach may be viewed as a special application of the theory of model order reduction [20]. The whole procedure is applied on a simple, two dimensional test problem, and the resulting box model is assessed.

The first part of the work gathers all the theoretical tools needed for that procedure. Chapter 1 is devoted to community detection and in particular to the *stability* measure and the related clustering algorithm. The reactive transport equation and the study of its properties is the topic of chapter 2, whereas compartment models and their properties are studied in chapter 3. Then, the link between the transport model and the Lagrangian equations describing the position of an individual particle is shown in chapter 4, leading to the theory of stochastic differential equations and of the consistent numerical methods to solve them. The second part of this work consists in applying the procedure on a simple problem. An idealized, two-dimensional overturning circulation model is presented in 6 and is then used the build the test problem in chapter 7. The complete procedure is applied on that test problem and assessed in the same chapter.



(a) Geometry of the box model of the isotopic carbon cycle (BICYCLE), where the arrows represent carbon fluxes between the compartments. This is figure 1 of [4].

(b) Representation of a subdomain decomposition of the Westerschelde. This is figure 1 of [3].

(c) Geometry and water fluxes of a compartmental model for the World Ocean. The fluxes are expressed in sverdrups (1 Sv = 10^6 m^3/s). This is figure 3 of [5].

Figure 1 – Schematic representations of different compartment models.

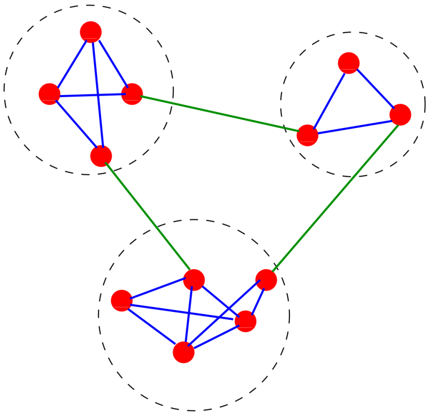


Figure 2 – Example of a three-communities partitioning on a simple network.

Chapter 1

The stability criterion for community detection

The partition of a graph into communities (or clusters) has been widely studied 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 cases one can even build a simplified functional description of the system 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. The reader may refer to [7] for a 2010 survey of the different clustering methods. 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 [17] and further expended in [19] and [18].

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 [21] to optimize the stability quality function. This heuristic algorithm has been initially developed for modularity optimization. However one can show that stability can be written as the *modularity* of a time-dependent network evolving under the Markov process [19]. Hence, the Louvain method can almost straightforwardly be applied to stability optimization.

This chapter is devoted to the explanation of the stability measure, and how to find good clusterings using stability analysis. It acts as a theoretical part intended to cover everything that is needed to make a proper, informed use of the stability toolbox. Notice that the stability measure has initially been presented for discrete times in [17]. We follow the same approach here: discrete-time stability is developed in the first section of this chapter; it is then extended to continuous time in a second section; finally, a few tools to analyze the robustness of a partition are presented in the third section of the chapter.

1.1 Discrete-time stability as an autocovariance

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 $\mathbf{q} = \mathbf{A}\mathbf{1}$; q_i is thus the total weight of the outgoing edges from node i. Let $\mathbf{Q} = \operatorname{diag}(\mathbf{q})$. Then, by normalizing the rows of $\mathbf{A}$ we get the matrix $\mathbf{M} = \mathbf{Q}^{-1}\mathbf{A}$, 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 :

$$\mathbf{p}_{t+1} = \mathbf{p}_t \mathbf{K}^{-1}\mathbf{A} = \mathbf{p}_t \mathbf{M}. \tag{1.1}$$

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, obtained by solving $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{M}$, and $\boldsymbol{\Pi} = \operatorname{diag}(\boldsymbol{\pi})$. 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

$$\begin{aligned}
\mathbf{C}(\mathbf{x}_{t_0}, \mathbf{x}_{t_0+t}) &\triangleq \mathbb{E}\left[(\mathbf{x}_{t_0} - \mathbb{E}[\mathbf{x}_{t_0}])^\intercal(\mathbf{x}_{t_0+t} - \mathbb{E}[\mathbf{x}_{t_0+t}])\right] && (1.2a)\\
&= \mathbb{E}\left[(\mathbf{x}_{t_0} - \boldsymbol{\pi})^\intercal(\mathbf{x}_{t_0+t} - \boldsymbol{\pi})\right] && (1.2b)\\
&= \mathbb{E}\left[\mathbf{x}_{t_0}^\intercal \mathbf{x}_{t_0+t}\right] - \mathbb{E}[\mathbf{x}_{t_0}^\intercal]\boldsymbol{\pi} - \boldsymbol{\pi}^\intercal \mathbb{E}\left[\mathbf{x}_{t_0+t}\right] + \boldsymbol{\pi}^\intercal\boldsymbol{\pi} && (1.2c)\\
&= \boldsymbol{\Pi}\mathbf{M}^t - \boldsymbol{\pi}^\intercal\boldsymbol{\pi}, && (1.2d)
\end{aligned}$$

where the fact that $\mathbf{C}(\mathbf{x}_{t_0}, \mathbf{x}_{t_0+t})$ 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. $[\mathbf{C}(\mathbf{x}_{t_0}, \mathbf{x}_{t_0+t})]_{ij}$ is interpreted as the correlation between $\mathbf{x}_{t_0,i}$ and $\mathbf{x}_{t_0+t,j}$. The independence on the initial time t_0 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

$$[\mathbf{H}_\mathcal{P}]_{ik} = \begin{cases} 1 & \text{if node } i \text{ is in community } k, \\ 0 & \text{otherwise.} \end{cases} \tag{1.3}$$

Let us define $\mathcal{H}_\mathcal{P} : \mathbb{R}^{n\times n} \to \mathbb{R}^{c\times c} : \mathbf{B} \mapsto \mathcal{H}_\mathcal{P}(\mathbf{B}) = \mathbf{H}_\mathcal{P}^\intercal \mathbf{B}\mathbf{H}_\mathcal{P}$. 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

$$[\mathbf{Y}]_{kl} = \sum_{i\in\mathcal{C}_k}\sum_{j\in\mathcal{C}_l}[\mathbf{X}]_{ij}, \tag{1.4}$$

where $\mathcal{C}_k$ and $\mathcal{C}_l$ denote communities 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

$$\begin{aligned}\mathbf{R}_t(\mathcal{P}) &= \mathcal{H}_\mathcal{P}\left(\mathbf{C}(\mathbf{x}_{t_0}, \mathbf{x}_{t_0+t})\right) && (1.5a)\\ &= \mathbf{C}(\mathbf{y}_{t_0}, \mathbf{y}_{t_0+t}) && (1.5b)\\ &= \mathbf{H}_\mathcal{P}^\intercal(\mathbf{\Pi}\mathbf{M}^t - \boldsymbol{\pi}^\intercal\boldsymbol{\pi})\mathbf{H}_\mathcal{P}. && (1.5c)\end{aligned}$$

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 over 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 independent random walkers are in $\mathcal{C}_k$ and $\mathcal{C}_l$, evaluated at stationarity. A good partition is such that there is a high likelihood of remaining in the starting community over a given time scale. The definition of the stability of a *clustering* $\mathcal{P}$ follows naturally:

$$r_t(\mathcal{P}) = \min_{0\le s\le t} \sum_{i=1}^{c} [\mathbf{R}_s]_{ii} = \min_{0\le s\le t} \operatorname{trace}(\mathbf{R}_s). \tag{1.6}$$

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 [18], this minimization is unnecessary in most cases and we have $r_t(\mathcal{P}) \approx \operatorname{trace}(\mathbf{R}_t)$. Nevertheless, taking the minimization ensures maximum generality and allows for example to deal with almost bipartite graphs where $\operatorname{trace}(\mathbf{R}_s)$ can be oscillatory.

All the definitions introduced until now are for a given partition $\mathcal{P}$. But what we ultimately want to compute is the optimal partition in the sense of stability, hence the one that maximizes the stability measure. Clearly, the optimal partition might be different for each Markov time t. Computing the optimal clustering for each Markov time gives the *stability curve of the graph* :

$$r_t = \max_{\mathcal{P}} r_t(\mathcal{P}). \tag{1.7}$$

Now we understand 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) as Markov time increases. 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:

$$\widetilde{\mathbf{M}} = (1-\tau)\mathbf{M} + \frac{1}{n}[(1-\tau)\mathbf{d} + \tau\mathbf{1}]\mathbf{1}^{\intercal}, \tag{1.8}$$

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 undergo a waiting time between each change of state which is itself a random variable. More precisely, the waiting time 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 (1.1) is governed by the following dynamics:

$$\dot{\mathbf{p}} = \mathbf{p}\,\mathrm{diag}\left\{\boldsymbol{\lambda}(\mathbf{q})\right\}\mathbf{Q}^{-1}\mathbf{A} - \mathbf{p}\,\mathrm{diag}\left\{\boldsymbol{\lambda}(\mathbf{q})\right\} = -\mathbf{p}\mathbf{L}, \tag{1.9}$$

where $\lambda_i(\mathbf{q})$ is the rate at which random walkers leave node i, and

$$\mathbf{L} = \mathrm{diag}\{\boldsymbol{\lambda}(\mathbf{q})\}[-\mathbf{Q}^{-1}\mathbf{A} + \mathbf{I}]. \tag{1.10}$$

Two particular cases of this process are implemented by the stability software and are thus examined here, depending on the choice of $\boldsymbol{\lambda}(\mathbf{q})$: the so-called *normalized Laplacian dynamics* and *standard (combinatorial) Laplacian dynamics*. Their names come from the similarity that arises between $\mathbf{L}$ and the normalized/standard Laplacian matrix. Each of those two dynamics represent best different physical processes. The former corresponds to the choice $\boldsymbol{\lambda}_{norm}(\mathbf{q}) = \mathbf{1}$. Hence, the expected waiting time is 1 at every node, and $\mathbf{L} = -\mathbf{Q}^{-1}\mathbf{A} + \mathbf{I} = -\mathbf{M} + \mathbf{I}$. The latter corresponds to $\boldsymbol{\lambda}_{combi}(\mathbf{q}) = \mathbf{q}/\langle\mathbf{q}\rangle$. In that case, $\mathbf{L} = (-\mathbf{A} + \mathbf{Q})/\langle\mathbf{q}\rangle$ and the average waiting time at node i is $\langle\mathbf{q}\rangle/q_i$. 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 $\langle\langle\mathbf{q}\rangle/\mathbf{q}\rangle = 1$. The corresponding governing equations are respectively

$$\dot{\mathbf{p}} = \mathbf{p}\mathbf{Q}^{-1}\mathbf{A} - \mathbf{p} = \mathbf{p}\mathbf{M} - \mathbf{p} \tag{1.11}$$

[1] In the original PageRank proposed by S. Brin and L. Page in 1998 (ref. [22]), this consists 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.

for the normalized Laplacian and

$$\dot{\mathbf{p}} = \mathbf{p}\frac{\mathbf{A}}{\langle \mathbf{q} \rangle} - \mathbf{p}\frac{\mathbf{Q}}{\langle \mathbf{q} \rangle} \tag{1.12}$$

for the combinatorial Laplacian.

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

$$\mathbf{R}(t;\mathcal{P}) = \mathbf{H}_{\mathcal{P}}^{\mathsf{T}}(\mathbf{\Pi}\mathbf{P}(t) - \boldsymbol{\pi}^{\mathsf{T}}\boldsymbol{\pi})\mathbf{H}_{\mathcal{P}}, \tag{1.13}$$

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:

$$r(t;\mathcal{P}) = \operatorname{trace}\left[\mathbf{R}(t;\mathcal{P})\right]. \tag{1.14}$$

Notice that it is not necessary to minimize over the time interval $[0,t]$: indeed, it can be shown that 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 $\mathcal{P}$:

$$r(t;\mathcal{P}) = \sum_{\mathcal{C}\in\mathcal{P}} P(\mathcal{C},t) - P(\mathcal{C},\infty). \tag{1.15}$$

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 independent walkers are in $\mathcal{C}$ at stationarity. Equation (1.15) tells us that only the communities in which a random walker is likely to stay bring 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 occurring by chance at stationarity. The stability curve of the graph can now be expressed as a continuous function of t:

$$r(t) = \max_{\mathcal{P}} r(t;\mathcal{P}). \tag{1.16}$$

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 altered 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* [23], 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 as

$$\mathrm{VI}(\mathcal{P}_1, \mathcal{P}_2) := \frac{H(\mathcal{P}_1, \mathcal{P}_2) - H(\mathcal{P}_1) - H(\mathcal{P}_2)}{\log(n)} = \frac{H(\mathcal{P}_1|\mathcal{P}_2) + H(\mathcal{P}_2|\mathcal{P}_1)}{\log(n)}, \tag{1.17}$$

where $\log(n)$ is a normalization factor;

$$H(\mathcal{P}) := -\sum_{\mathcal{C}\in\mathcal{P}} p(\mathcal{C}) \log[p(\mathcal{C})] \tag{1.18}$$

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 both to a community $\mathcal{C}_1$ of $\mathcal{P}_1$ and to a community $\mathcal{C}_2$ of $\mathcal{P}_2$. We have

$$p(\mathcal{C}_1, \mathcal{C}_2) = \frac{n_{\mathcal{C}_1\cap\mathcal{C}_2}}{n}, \tag{1.19}$$

and

$$H(\mathcal{P}_1, \mathcal{P}_2) := -\sum_{\mathcal{C}_1\in\mathcal{P}_1} \sum_{\mathcal{C}_2\in\mathcal{P}_2} p(\mathcal{C}_1, \mathcal{C}_2) \log[p(\mathcal{C}_1, \mathcal{C}_2)]. \tag{1.20}$$

Similarly, $H(\mathcal{P}_1|\mathcal{P}_2)$ 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 between two partitions is then used as follows: for each Markov time, an ensemble of Louvain optimizations of stability are performed, starting from different random initial node ordering.[2] The normalized variation of information allows then to quantify how different the optimized partitions are. Therefore, a low variation of information indicates optimized partitions that are very similar to each others, and thus 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 optimization method.

[2] 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.

Chapter 2

The transport model

2.1 The reactive transport equation

The evolution of the concentration of a particular constituent (or tracer) in an aquatic environment is described by the *reactive transport equation*:

$$\frac{\partial C}{\partial t} = q - \nabla \cdot (\mathbf{u}C - \mathbf{K}\nabla C)\,. \tag{2.1}$$

In this equation, C is the concentration of the constituent in water, expressed in $\mathrm{kg/m^d}$, where d is the dimension of the problem; t is the time, expressed in seconds; q is the local net production (i.e. production - destruction) rate of the constituent, expressed in $\mathrm{kg/(m^d s)}$; $\mathbf{u}$ is the velocity vector whose units are m/s; and $\mathbf{K}$ is the *diffusivity tensor*, whose entries are expressed in $\mathrm{m^2/s}$. Without loss of generality, we can assume $\mathbf{K}$ to be symmetric. This is essentially because the impact of the anti-symmetric part of $\mathbf{K}$, if any, may be viewed as additional advection. More details may be found in appendix A of [24]. Of course, the symmetric tensor $\mathbf{K}$ must then be positive-definite in order to represent truly diffusive processes, namely phenomena which tend, at any time and location, to homogenize the concentration of any constituent.

In equation (2.1), $\nabla \cdot (\mathbf{u}C)$ is the *advective* term, $\nabla \cdot (\mathbf{K}\nabla C)$ is the *diffusive* term, and q is often called the *source/sink* term. When $q = 0$, equation (2.1) becomes the *advection-diffusion* equation:

$$\frac{\partial C}{\partial t} = -\nabla \cdot (\mathbf{u}C - \mathbf{K}\nabla C)\,. \tag{2.2}$$

When $q = 0$, the tracer is said to be *passive.*

The reactive transport equation is omnipresent in this work for several reasons. First, it is a fundamental equation in our purpose to develop consistent compartment models, as we will see in chapter 3. Furthermore, we have seen in chapter 1 that the stability method is based on the knowledge of the *transition probability matrix.* Computing that matrix in the case of geophysical flows cannot be done unless we understand the dynamics of the flow: we will see in chapter 4 that efficient numerical methods allowing to track the fate of individual tracer's particles can be devised from equation (2.1). The transition probability matrix can then easily be approximated from the particles trajectories. Finally, the particles trajectories can also be used to evaluate the concentration, and hence to assess the validity of the compartment models that will be developed further in this work.

2.2 The continuity equation and the Boussinesq approximation

In general, the (possibly time-dependent) velocity field in equation (2.1) is unknown and has to be solved from the Navier-Stokes equations. In this work, we will restrict ourselves to problems where the velocity field is known and stationary. The *continuity equation* (local mass conservation)

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.3}$$

must be satisfied everywhere on the domain. In equation (2.3), ρ is the density of the (sea)water mixture. In order to simplify the continuity equation, we make the very common *Boussinesq approximation.* In the aquatic environment, water is, by far, the dominant constituent. The density of seawater is thus close to that of pure water, ρ_w. The latter depends on the temperature and pressure, but the variations are often very small: this consideration is at the basis of the Boussinesq approximation. Let $\bar{\rho}$ and $\Delta\rho$ be appropriate reference values of the density and the order of magnitude of its variation. The key assumption in the *Boussinesq approximation* is that

$$\frac{\Delta\rho}{\bar{\rho}} \ll 1. \tag{2.4}$$

To assess the impact of this assumption on the continuity equation, we consider its dimensionless form. Let U, T and X be relevant velocity-, time- and space-scales. We use those parameters to scale the flow variables, leading to the following dimensionless variables (denoted by primes):

$$\rho' = \frac{\rho - \bar{\rho}}{\Delta\rho}, \quad \mathbf{u}' = \frac{\mathbf{u}}{U}, \quad t' = \frac{t}{T}, \quad \text{and} \quad \mathbf{x}' = \frac{\mathbf{x}}{X}, \tag{2.5}$$

where $\mathbf{x} = (y, z)$. The dimensionless version of the continuity equation (2.3) reads then:

$$\frac{\Delta\rho}{T}\frac{\partial \rho'}{\partial t'} + \frac{U\Delta\rho}{X}\mathbf{u}' \cdot \nabla'\rho' + \frac{U(\bar{\rho} + \rho'\Delta\rho)}{X}\nabla' \cdot \mathbf{u}' = 0. \tag{2.6}$$

Multiplying both sides by $X/(U\bar{\rho})$ yields :

$$\frac{X}{UT}\frac{\Delta\rho}{\bar{\rho}}\frac{\partial \rho'}{\partial t'} + \frac{\Delta\rho}{\bar{\rho}}\mathbf{u}' \cdot \nabla'\rho' + \left(1 + \frac{\Delta\rho}{\bar{\rho}}\rho'\right)\nabla' \cdot \mathbf{u}' = 0. \tag{2.7}$$

By taking (2.4) into account, this equation simplifies to $\nabla' \cdot \mathbf{u}' = 0$, or equivalently in dimensional variables

$$\nabla \cdot \mathbf{u} = 0. \tag{2.8}$$

2.3 Properties of the solution of the reactive transport equation

In this section, we show the most important properties related to the reactive transport equation. To this end, let us first introduce some notations and terminology. Let Ω denote the (time independent) domain of interest and $\partial\Omega$ its boundary. Let $\hat{\mathbf{n}}$ denote the unit vector, normal to $\partial\Omega$ and oriented towards the exterior of the domain. The volume of the domain is

$$|\Omega| = \int_\Omega \mathrm{d}\Omega. \tag{2.9}$$

The mean tracer concentration over the domain is

$$\bar{C}(t) = \frac{1}{|\Omega|}\int_\Omega C(\mathbf{x},t)\mathrm{d}\Omega, \tag{2.10}$$

and the variance of the concentration over Ω is expressed as

$$\sigma^2(t) = \frac{1}{|\Omega|}\int_\Omega \left(\hat{C}(\mathbf{x},t)\right)^2 \mathrm{d}\Omega, \tag{2.11}$$

where

$$\hat{C}(\mathbf{x},t) := C(\mathbf{x},t) - \bar{C} \tag{2.12}$$

denotes the deviation of the concentration with respect to $\bar{C}$. The variance is a measure of the concentration inhomogeneity. The mean net production rate is denoted

$$\bar{q}(t) = \frac{1}{|\Omega|}\int_\Omega q \ \mathrm{d}\Omega. \tag{2.13}$$

The flux that enters the domain through the boundary $\partial\Omega$ is

$$\Phi = -\int_{\partial\Omega} (\mathbf{u}C - \mathbf{K}\cdot\nabla C)\cdot\hat{\mathbf{n}} \ \mathrm{d}(\partial\Omega). \tag{2.14}$$

An *isolated* domain is such that there is no exchange with the environment, and hence no net flux of the fluid crossing the boundary:

$$[\mathbf{u}\cdot\hat{\mathbf{n}}]_{\mathbf{x}\in\partial\Omega} = 0 \quad \text{and} \quad [(\mathbf{K}\cdot\nabla C)\cdot\hat{\mathbf{n}}]_{\mathbf{x}\in\partial\Omega} = 0. \tag{2.15}$$

Therefore, $\Phi = 0$ if the domain is isolated.

Property 2.3.1. *The mean tracer concentration satisfies the following relation*

$$\frac{d\bar{C}(t)}{dt} = \bar{q} + \frac{\Phi}{|\Omega|}. \tag{2.16}$$

Proof. The time derivative of the mean concentration is expressed as

$$\frac{d\bar{C}(t)}{dt} = \frac{1}{|\Omega|}\frac{d}{dt}\int_\Omega C(\mathbf{x},t)\mathrm{d}\Omega = \frac{1}{|\Omega|}\int_\Omega \frac{\partial}{\partial t}C(\mathbf{x},t)\mathrm{d}\Omega. \tag{2.17}$$

Injecting (2.1) into (2.17), yields successively

$$\begin{aligned}\frac{d\bar{C}(t)}{dt} &= \frac{1}{|\Omega|}\int_\Omega q \ \mathrm{d}\Omega + \frac{1}{|\Omega|}\int_\Omega -\nabla\cdot(\mathbf{u}C - \mathbf{K}\nabla C)\,\mathrm{d}\Omega && \text{(2.18a)}\\ &= \bar{q} - \frac{1}{|\Omega|}\int_{\partial\Omega}(\mathbf{u}C - \mathbf{K}\nabla C)\cdot\hat{\mathbf{n}} \ \mathrm{d}(\partial\Omega) && \text{(2.18b)}\\ &= \bar{q} + \frac{\Phi}{|\Omega|}, && \text{(2.18c)}\end{aligned}$$

where we have used the divergence theorem. □

Corollary 2.3.1. *If the domain is isolated and the tracer is passive, the mean tracer concentration is constant.*

Proof. This is a straightforward consequence of property 2.3.1 with $\Phi = 0$ (isolated domain) and $\bar{q} = 0$ (passive tracer). □

Property 2.3.2. *For a passive tracer in an isolated domain, if the initial concentration is constant, then the concentration remains constant.*

Proof. Let $C_0 \geq 0$ and $C(\mathbf{x}, 0) = C_0$. At time $t = 0$, equation (2.1) becomes

$$\frac{\partial C}{\partial t} = q - C_0 \nabla \cdot \mathbf{u}. \tag{2.19}$$

But $q = 0$ since the tracer is passive, and the Boussinesq approximation implies that $\mathbf{u}$ is divergence-free (equation (6.2)). Hence

$$\frac{\partial C}{\partial t} = 0, \tag{2.20}$$

and the concentration remains constant. □

Property 2.3.3. *For a passive tracer in an isolated domain, the variance of the concentration decreases monotonously with time until the concentration is everywhere equal to the mean concentration. In other words, the tracer concentration gets more and more homogeneous with time until the equilibrium state is reached, which happens when the concentration is constant and uniform everywhere on Ω.*

Proof. By corollary 2.3.1, the mean concentration $\bar{C}$ is constant. Let $\hat{C}(\mathbf{x}, t)$ be defined as in (2.12). It is easy to show that $\bar{C}$ satisfies the advection diffusion equation:

$$0 = \frac{\partial \bar{C}}{\partial t} = -\nabla \cdot \left(\mathbf{u}\bar{C} - \mathbf{K}\nabla\bar{C} \right) = -\bar{C}\nabla \cdot (\mathbf{u} - 0) = 0, \tag{2.21}$$

and thus

$$\frac{\partial \hat{C}}{\partial t} = -\nabla \cdot \left(\mathbf{u}\hat{C} - \mathbf{K}\nabla\hat{C} \right). \tag{2.22}$$

Multiplying equation (2.22) by $\hat{C}$ yields, after some calculations

$$\frac{\partial \hat{C}^2}{\partial t} = -\nabla \cdot \left(\hat{C}^2 \mathbf{u} \right) + 2\nabla \cdot \left(\hat{C}\mathbf{K}\nabla\hat{C} \right) - 2\nabla\hat{C}^\intercal \mathbf{K} \nabla \hat{C}. \tag{2.23}$$

Now, notice that

$$\int_\Omega \frac{\partial \hat{C}^2}{\partial t} \mathrm{d}\Omega = \frac{d}{dt} \int_\Omega \left(\hat{C}(\mathbf{x}, t) \right)^2 \mathrm{d}\Omega = |\Omega| \frac{d\sigma^2}{dt}. \tag{2.24}$$

Combining (2.23) and (2.24) yields

$$|\Omega| \frac{d\sigma^2}{dt} = -\int_\Omega \nabla \cdot \left(\hat{C}^2 \mathbf{u} \right) \mathrm{d}\Omega + 2 \int_\Omega \nabla \cdot \left(\hat{C}\mathbf{K}\nabla\hat{C} \right) \mathrm{d}\Omega - 2 \int_\Omega \nabla\hat{C}^\intercal \mathbf{K}\nabla\hat{C} \mathrm{d}\Omega. \tag{2.25}$$

Using the divergence theorem and equations (2.15), we get

$$|\Omega| \frac{d\sigma^2}{dt} = \underbrace{-\int_{\partial\Omega} \hat{C}^2 (\mathbf{u} \cdot \hat{\mathbf{n}}) \mathrm{d}(\partial\Omega)}_{=0} + 2 \underbrace{\int_{\partial\Omega} \hat{C} (\mathbf{K}\nabla\hat{C} \cdot \hat{\mathbf{n}}) \mathrm{d}(\partial\Omega)}_{=0} - 2 \int_\Omega \nabla\hat{C}^\intercal \mathbf{K}\nabla\hat{C} \mathrm{d}\Omega. \tag{2.26}$$

Finally,

$$\frac{d\sigma^2}{dt} = -\frac{2}{|\Omega|} \int_\Omega \nabla\hat{C}^\intercal \mathbf{K}\nabla\hat{C} \mathrm{d}\Omega. \tag{2.27}$$

Since $\mathbf{K}$ is positive definite, the variance of the concentration will decrease until $\nabla\hat{C} = 0$, hence until C reaches a constant value everywhere on Ω. The only possibility is $\bar{C}$, and thus we have shown that

$$\lim_{t \to \infty} C(\mathbf{x}, t) = \bar{C}. \tag{2.28}$$

□

Chapter 3

Structure of a compartment model for tracer transport

A relevant compartment model should reproduce the properties covered in section 2.3 for the corresponding continuous model. In this chapter, a compartment model structure that fulfills that requirement is proposed, and the corresponding matrix form is derived. This chapter is widely inspired from [25] and [26].

3.1 Formulation of a compartment model

Recall that the evolution of the concentration $C(t, \mathbf{x})$ of a tracer in a domain Ω obeys the following equation:

$$\frac{\partial C}{\partial t} = q - \nabla \cdot (\mathbf{u}C - \mathbf{K} \cdot \nabla C), \tag{3.1}$$

where $\mathbf{u}$ is the velocity field and q is the source or sink term, i.e. the rate at which the tracer is produced or destroyed. In the case of a passive tracer, $q = 0$. Under the Boussinesq approximation, the continuity equation simplifies to

$$\nabla \cdot \mathbf{u} = 0, \tag{3.2}$$

namely the velocity field is divergence-free. As already stated, the diffusivity tensor $\mathbf{K}$ is symmetric and positive definite.

In order to derive a compartment model from (3.1), Ω must be partitioned into N subdomains $\Omega_1, \Omega_2, \ldots, \Omega_N$ called the *compartments*. Mathematically, the fact that these subdomains form a partition of Ω means that

$$\bigcup_{i \in \{1,\ldots,N\}} \Omega_i = \Omega \qquad \text{and} \qquad \Omega_i \cap \Omega_j = \varnothing \text{ if } i \neq j. \tag{3.3}$$

In this section, unless otherwise stated, the subscripts i and j are implicitly assumed to be in $\{1, \ldots, N\}$. The interface between the subdomains Ω_i and Ω_j is denoted $\Gamma_{i,j}$ and the interface between Ω_i and the environment is denoted $\Gamma_{i,e}$. Obviously, $\Gamma_{i,j} = \Gamma_{j,i}$ and $\cup_{i \in \{1,\ldots,N\}} \Gamma_{i,e} = \partial\Omega$. In a compartment model, only the averages over the compartments are considered. The mean tracer concentration over compartment i is

$$C_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} C(t, \mathbf{x}) \mathrm{d}\Omega_i, \tag{3.4}$$

and the net production rate over the subdomain i is

$$q_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} q(t, \mathbf{x}) \mathrm{d}\Omega_i. \tag{3.5}$$

The equation governing the evolution of $C_i(t)$, the average concentration over compartment i, is obtained by integrating (3.1) over Ω_i. This yields, using the divergence theorem:

$$|\Omega_i| \frac{dC_i}{dt} = |\Omega_i| q_i - \sum_{\substack{j=1 \\ j \neq i}}^{N} \underbrace{\int_{\Gamma_{i,j}} (\mathbf{u}C - \mathbf{K} \cdot \nabla C) \cdot \mathbf{n}_{i,j} \mathrm{d}\Gamma_{i,j}}_{:=\phi_{i,j}} - \underbrace{\int_{\Gamma_{i,e}} (\mathbf{u}C - \mathbf{K} \cdot \nabla C) \cdot \mathbf{n}_{i,e} \mathrm{d}\Gamma_{i,e}}_{:=\phi_{i,e}}, \tag{3.6}$$

where $\phi_{i,j}$ is interpreted as the tracer flux from compartment i to compartment j, and $\phi_{i,e}$ is the flux of tracer leaving compartment i towards the environment. In the framework of a compartment model, only the mean concentration in each compartment are available, making it impossible to evaluate the integrals in the right-hand side exactly. The flux $\phi_{i,j}$ can be split into an advective and a diffusive part:

$$\phi_{i,j} = \underbrace{\int_{\Gamma_{i,j}} (\mathbf{u}C) \cdot \mathbf{n}_{i,j} \mathrm{d}\Gamma_{i,j}}_{:=\phi^A_{i,j} \text{ (advective part)}} + \underbrace{\int_{\Gamma_{i,j}} (-\mathbf{K} \cdot \nabla C) \cdot \mathbf{n}_{i,j} \mathrm{d}\Gamma_{i,j}}_{:=\phi^D_{i,j} \text{ (diffusive part)}}. \tag{3.7}$$

A natural approximation of the advective flux $\phi^A_{i,j}$ is obtained as

$$\phi^A_{i,j} \approx \frac{C_i + C_j}{2} \int_{\Gamma_{i,j}} \mathbf{u} \cdot \mathbf{n}_{i,j} \mathrm{d}\Gamma_{i,j} = \frac{C_i + C_j}{2} |\Gamma_{i,j}| u_{i,j}, \tag{3.8}$$

where a characteristic speed $u_{i,j}$ has been introduced such that

$$|\Gamma_{i,j}| u_{i,j} = \int_{\Gamma_{i,j}} \mathbf{u} \cdot \mathbf{n}_{i,j} \mathrm{d}\Gamma_{i,j}. \tag{3.9}$$

Since $\phi^A_{i,j} = -\phi^A_{j,i}$ and $\Gamma_{i,j} = \Gamma_{j,i}$, the characteristic speed must satisfy

$$u_{i,j} = -u_{j,i}. \tag{3.10}$$

The definition (3.9) satisfies that condition. The diffusive flux $\phi^D_{i,j}$ involves the gradient of the concentration at the interface. A possible approximation is given by

$$\phi^D_{i,j} = -|\Gamma_{i,j}| k_{i,j} \frac{C_j - C_i}{l_{i,j}}, \tag{3.11}$$

where $k_{i,j} > 0$ is a characteristic diffusivity and $l_{i,j} > 0$ is a characteristic length. Obviously, $\phi^D_{i,j} = -\phi^D_{j,i}$ hence $k_{i,j} = k_{j,i}$ and $l_{i,j} = l_{j,i}$. In [25], *Deleersnijder* proposes to simplify those notations by introducing advective and diffusive "fluxes" $U_{i,j}$ and $V_{i,j}$:

$$U_{i,j} = |\Gamma_{i,j}| u_{i,j} \quad \text{and} \quad V_{i,j} = \frac{|\Gamma_{i,j}| k_{i,j}}{l_{i,j}}. \tag{3.12}$$

By the previously mentioned properties, it is obvious that

$$U_{i,j} = -U_{j,i} \tag{3.13}$$

and that

$$V_{i,j} = V_{j,i} > 0. \tag{3.14}$$

Using those notations, the flux from compartment i to compartment j is approximated by

$$\phi_{i,j} \approx U_{i,j}\frac{C_i + C_j}{2} - V_{i,j}(C_j - C_i), \tag{3.15}$$

and the equation governing the evolution of the concentration in compartment i is obtained as

$$\Omega_i \frac{dC_i}{dt} = \Omega_i q_i - \sum_{\substack{j=1\\ j\neq i}}^{N} \left[U_{i,j}\frac{C_i + C_j}{2} - V_{i,j}(C_j - C_i)\right] - \phi_{i,e}. \tag{3.16}$$

Let us fix the convention $U_{i,i} = 0$ and $V_{i,i} = 0$ so that the summation subscript $j \neq i$ becomes unnecessary. Note that by (3.12), (3.9) and the divergence theorem,

$$\sum_{j=1}^{N} U_{i,j} + U_{i,e} = \sum_{\substack{j=1\\ j\neq i}}^{N} \int_{\Gamma_{i,j}} \mathbf{u}\cdot\mathbf{n}_{i,j}\mathrm{d}\Gamma_{i,j} + \int_{\Gamma_{i,e}} \mathbf{u}\cdot\mathbf{n}_{i,e}\mathrm{d}\Gamma_{i,e} = \int_{\Omega_i} (\nabla\cdot\mathbf{u})\mathrm{d}\Omega_i = 0, \tag{3.17}$$

the counterpart to the continuity equation.

3.2 Properties of the compartment model solution

In section 2.3, important properties of the continuous transport model have been derived. In this section, the counterparts of those properties are established in the case of a compartment model. The domain-averaged concentration can be expressed in terms of the compartments concentrations. Indeed, by equations (2.10) and (3.4):

$$\bar{C}(t) = \frac{1}{|\Omega|}\int_{\Omega} C(\mathbf{x},t)\mathrm{d}\Omega = \frac{1}{|\Omega|}\sum_{i=1}^{N}\int_{\Omega_i} C(\mathbf{x},t)\mathrm{d}\Omega_i = \frac{1}{|\Omega|}\sum_{i=1}^{N}|\Omega_i|C_i(t). \tag{3.18}$$

The variance of the concentration is approximated by the *box variance*

$$\sigma^2_{box}(t) = \frac{1}{|\Omega|}\sum_{i=1}^{N}|\Omega_i|\left[C_i(t) - \bar{C}(t)\right]^2. \tag{3.19}$$

Property 3.2.1. *The mean tracer concentration $\bar{C}$ is not influenced by the internal advective and diffusive fluxes.*

Proof. By (3.18), (3.16), (3.13) and (3.14), we get successively

$$\frac{d\bar{C}}{dt} = \frac{1}{|\Omega|}\sum_{i=1}^{N}|\Omega_i|\frac{dC_i(t)}{dt} \tag{3.20a}$$

$$= \frac{1}{|\Omega|}\sum_{i=1}^{N}\left[|\Omega_i|q_i - \sum_{j=1}^{N}\left[U_{i,j}\frac{C_i + C_j}{2} - V_{i,j}(C_j - C_i)\right] - \phi_{i,e}\right] \tag{3.20b}$$

$$= \frac{1}{|\Omega|}\sum_{i=1}^{N}(|\Omega_i|q_i + \phi_{e,i}). \tag{3.20c}$$

□

Corollary 3.2.1. *For a passive tracer in an isolated domain, the mean tracer concentration within the whole domain $\bar{C}$ is a constant.*

Proof. The proof is straightforward by introducing $q_i = 0$ (passive tracer) and $\phi_{i,e} = 0$ (isolated domain) in equation (3.20c). □

Property 3.2.2. *If the tracer is passive, the domain isolated and the initial concentration is constant, then the concentration stays constant.*

Proof. If the initial concentration is constant, then $C_i(0) = C_0$ for all $i = 1, \dots, N$. At time $t = 0$, equation (3.16) becomes

$$|\Omega_i| \frac{dC_i}{dt} = -C_0 \sum_{j=1}^{N} U_{i,j}, \tag{3.21}$$

which is equal to zero by (3.17). Here we have used that $q_i = 0$ and $\phi_{i,e} = 0$ (and thus $U_{i,e} = 0$) since the tracer is passive and the domain is isolated. □

Property 3.2.3. *For a passive tracer ($q_i = 0$) in an isolated domain ($\phi_{i,e} = 0$), the box variance of the concentration decreases monotonically until $C_i(t) = \bar{C}(0)$ for all $i = 1, \dots, N$ (i.e., until the tracer distribution is uniform and equal to $\bar{C}(0)$ everywhere).*

Proof. Let $\hat{C}_i(t)$ denote the deviation of the concentration in compartment i with respect to the mean concentration over the whole domain:

$$\hat{C}_i(t) := C_i(t) - \bar{C}(t). \tag{3.22}$$

By property 3.2.2, $\bar{C}(t) = \bar{C}(0) = \bar{C}$ since the domain is isolated and the tracer is passive. Hence, we can rewrite equation (3.16) as

$$|\Omega_i| \frac{d\hat{C}_i}{dt} = -\sum_{j=1}^{N} \left[U_{i,j} \frac{\hat{C}_i + \hat{C}_j}{2} - V_{i,j}(\hat{C}_j - \hat{C}_i) \right] - \bar{C} \sum_{j=1}^{N} U_{i,j}, \tag{3.23}$$

which by (3.17) simplifies to

$$|\Omega_i| \frac{d\hat{C}_i}{dt} = -\sum_{j=1}^{N} \left[U_{i,j} \frac{\hat{C}_i + \hat{C}_j}{2} - V_{i,j}(\hat{C}_j - \hat{C}_i) \right]. \tag{3.24}$$

Multiplying (3.24) by $\hat{C}_i(t)$ and summing over $i = 1, \dots, N$ yields after some calculations

$$|\Omega| \frac{d\sigma^2_{box}}{dt} = -\sum_{i=1}^{N} \sum_{j=1}^{N} \left[U_{i,j}(\hat{C}_i + \hat{C}_j)\hat{C}_i - 2V_{i,j}(\hat{C}_j - \hat{C}_i)\hat{C}_i \right]. \tag{3.25}$$

Let us look at both terms in the right-hand side of (3.25) separately. The first term is equal to zero:

$$\sum_{i=1}^{N} \sum_{j=1}^{N} U_{i,j}(\hat{C}_i + \hat{C}_j)\hat{C}_i = \sum_{i=1}^{N} \left(\hat{C}_i\right)^2 \underbrace{\sum_{j=1}^{N} U_{i,j}}_{=0} + \sum_{i=1}^{N} \sum_{j=1}^{N} U_{i,j}\hat{C}_j\hat{C}_i \tag{3.26a}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \underbrace{\frac{U_{i,j} + U_{j,i}}{2}}_{=0} \hat{C}_j\hat{C}_i. \tag{3.26b}$$

The second term is rewritten as

$$2\sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)\hat{C}_i = -2\sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)^2 - 2\sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)\hat{C}_i, \tag{3.27}$$

so that

$$2\sum_{i=1}^{N}\sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)\hat{C}_i = -\sum_{i=1}^{N}\sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)^2. \tag{3.28}$$

Finally,

$$|\Omega|\frac{d\sigma_{box}^2}{dt} = -\sum_{i=1}^{N}\sum_{j=1}^{N} V_{i,j}(\hat{C}_j - \hat{C}_i)^2, \tag{3.29}$$

and thus

$$\frac{d\sigma_{box}^2}{dt} = -\frac{1}{|\Omega|}\sum_{i=1}^{N}\sum_{j=1}^{N} V_{i,j}(C_j - C_i)^2, \tag{3.30}$$

which proves the claim. Interestingly, the advective part of the fluxes does not contribute to the homogenization of the concentration; only the diffusive part does. □

3.3 Matrix formulation

The generic compartment model described in the previous sections can be written in matrix form. For simplicity and because we will restrict ourselves to such systems in the next, we consider the case of an isolated system, i.e. $\phi_{i,e} = 0$. Let

$$\mathbf{c} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{pmatrix}, \quad \boldsymbol{\omega} = \begin{pmatrix} |\Omega_1| \\ |\Omega_2| \\ \vdots \\ |\Omega_N| \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\Omega} = \text{diag}(\boldsymbol{\omega}). \tag{3.31}$$

The evolution of the concentrations in the compartments is given by

$$\boldsymbol{\Omega}\dot{\mathbf{c}} = \boldsymbol{\Omega}\mathbf{q} + \mathbf{A}\mathbf{c}, \tag{3.32}$$

where $\mathbf{A}$ is the *interaction matrix*, which describes the advective and diffusive fluxes between the subdomains. Using the parameters introduced previously, $\mathbf{A}$ can be expressed as

$$\mathbf{A} = \begin{pmatrix} -\sum_{j=1}^{N}\frac{1}{2}U_{1,j} + V_{1,j} & -\frac{1}{2}U_{1,2} + V_{1,2} & \cdots & -\frac{1}{2}U_{1,N} + V_{1,N} \\ -\frac{1}{2}U_{2,1} + V_{2,1} & -\sum_{j=1}^{N}\frac{1}{2}U_{2,j} + V_{2,j} & \cdots & -\frac{1}{2}U_{2,N} + V_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{2}U_{N,1} + V_{N,1} & -\frac{1}{2}U_{N,2} + V_{N,2} & \cdots & -\sum_{j=1}^{N}\frac{1}{2}U_{N,j} + V_{N,j} \end{pmatrix}. \tag{3.33}$$

In matrix formulation, the mean concentration is expressed as

$$\bar{C} = \frac{1}{|\Omega|}\mathbf{1}^{\intercal}\boldsymbol{\Omega}\mathbf{c}, \tag{3.34}$$

where $\mathbf{1}$ is the N-dimensional unit column vector. Notice that $|\Omega| = \mathbf{1}^{\intercal}\boldsymbol{\Omega}\mathbf{1}$. In the next of this section, we show how properties 3.2.1, 3.2.2 and 3.2.3 from section 3.2 are expressed in matrix formulation.

Property 3.3.1. *Each column of the interaction matrix sums to zero:*

$$\mathbf{1}^{\intercal}\mathbf{A} = 0. \tag{3.35}$$

Proof. From equations (3.32) and (3.34),

$$|\Omega|\dot{\bar{C}} = \mathbf{1}^\mathsf{T}\mathbf{\Omega}\dot{\mathbf{c}} = \mathbf{1}^\mathsf{T}\mathbf{\Omega}\mathbf{q} + \mathbf{1}^\mathsf{T}\mathbf{A}\mathbf{c}. \tag{3.36}$$

This equation is valid for an isolated system. If furthermore the tracer is passive ($\mathbf{q} = 0$), we know by corollary 3.2.1 that $\bar{C}$ must be constant. Hence

$$|\Omega|\dot{\bar{C}} = \mathbf{1}^\mathsf{T}\mathbf{A}\mathbf{c} = 0. \tag{3.37}$$

Equation (3.37) must be valid for any concentration vector $\mathbf{c}$, thus $\mathbf{1}^\mathsf{T}\mathbf{A} = 0$. □

Property 3.3.2. *Each row of the interaction matrix sums to zero:*

$$\mathbf{A}\mathbf{1} = 0. \tag{3.38}$$

Proof. Let $C_0 > 0$ be a constant and $\mathbf{c} = C_0\mathbf{1}$. Introducing the latter in (3.32) with $\mathbf{q} = \mathbf{0}$ yields

$$0 = C_0\mathbf{A}\mathbf{1}. \tag{3.39}$$

□

Property 3.3.3. *The interaction matrix is negative definite.*

Proof. Let $\mathbf{q} = 0$. Since we consider an isolated domain, $\bar{C}$ is a constant. Consider $\hat{\mathbf{c}} = \mathbf{c} - \bar{C}\mathbf{1}$, the deviation of the concentration with respect to $\bar{C}$. Since under those conditions, (3.32) is linear and homogeneous in $\mathbf{c}$, it applies to any perturbation of $\mathbf{c}$ by a constant, and thus to $\hat{\mathbf{c}}$:

$$\mathbf{\Omega}\dot{\hat{\mathbf{c}}} = \mathbf{A}\hat{\mathbf{c}}. \tag{3.40}$$

The box variance is expressed as

$$\sigma^2_{box} = \frac{1}{|\Omega|}\hat{\mathbf{c}}^\mathsf{T}\mathbf{\Omega}\hat{\mathbf{c}}. \tag{3.41}$$

A time-differentiation of (3.41) yields

$$\frac{d\sigma^2_{box}}{dt} = \frac{2}{|\Omega|}\hat{\mathbf{c}}^\mathsf{T}\mathbf{\Omega}\dot{\hat{\mathbf{c}}}, \tag{3.42}$$

and finally by (3.40)

$$\frac{|\Omega|}{2}\frac{d\sigma^2_{box}}{dt} = \hat{\mathbf{c}}^\mathsf{T}\mathbf{A}\hat{\mathbf{c}}. \tag{3.43}$$

By property 3.2.3, the latter must be negative (or zero if $\hat{\mathbf{c}} = \mathbf{0}$), which shows that $\mathbf{A}$ is negative definite. □

3.4 Discrete-time compartment model

It is also possible to build a discrete-time compartment model, and this could even be more relevant than a continuous-time one in the context of numerical implementation. Consider equation (3.32). For a given time step $\Delta t > 0$, we would like to build a similar expression that would link $\mathbf{c}(t + \Delta t)$ to $\mathbf{c}(t)$. From the theory of ordinary differential equations [27], we know that the general solution to that equation is

$$\mathbf{c}(t) = \mathrm{e}^{\mathbf{\Omega}^{-1}\mathbf{A}(t-t_0)}\mathbf{c}(t_0) + \int_{t_0}^{t} \mathrm{e}^{\mathbf{\Omega}^{-1}\mathbf{A}(t-s)}\mathbf{q}(s)\mathrm{d}s. \tag{3.44}$$

For a general source term $\mathbf{q}$, it is not possible to write a matrix expression of the form $\mathbf{c}(t+\Delta t) = \mathbf{A}_{\Delta t}\mathbf{c}(t) + \mathbf{B}_{\Delta t}\mathbf{q}(t)$, but it is possible under certain conditions on $\mathbf{q}$. In this work, we won't need such conditions because we make the assumption that there is no source/sink term, i.e. we assume that $\mathbf{q} = 0$. Let $t_0 < t_1 < t_2 < \dots$ be such that $t_{k+1} = t_k + \Delta t$. If $\mathbf{c}(t_k)$ is known, $\mathbf{c}(t_{k+1})$ can be computed using (3.44) with $\mathbf{q} = 0$:

$$\mathbf{c}(t_{k+1}) = \mathrm{e}^{\boldsymbol{\Omega}^{-1}\mathbf{A}\Delta t}\,\mathbf{c}(t_k). \tag{3.45}$$

Doing this, we have found a matrix relation that links $\mathbf{c}(t+\Delta t)$ to $\mathbf{c}(t)$:

$$\mathbf{c}(t+\Delta t) = \mathbf{A}_{\Delta t}\mathbf{c}(t), \tag{3.46}$$

where $\mathbf{A}_{\Delta t} = \mathrm{e}^{\boldsymbol{\Omega}^{-1}\mathbf{A}\Delta t}$ is the *discrete interaction matrix*. Notice that the discrete interaction matrix is closely related to the transition probability matrix introduced in chapter 1. Indeed, developing (3.46) for $C_i(t)$ yields

$$C_i(t+\Delta t) = [\mathbf{A}_{\Delta t}]_{i,1}C_1(t) + [\mathbf{A}_{\Delta t}]_{i,2}C_2(t) + \dots + [\mathbf{A}_{\Delta t}]_{i,N}C_N(t). \tag{3.47}$$

Let $P_i(t)$ denote the number of tracer's particles in compartment i at time t, and let P be the mass of one tracer's particle. Notice that

$$C_i(t) = \frac{P_i(t)P}{|\Omega_i|}. \tag{3.48}$$

Multiplying equation (3.47) by $|\Omega_i|/P$ yields after some manipulations

$$P_i(t) = [\mathbf{A}_{\Delta t}]_{i,1}\frac{|\Omega_i|}{|\Omega_1|}P_1(t) + [\mathbf{A}_{\Delta t}]_{i,2}\frac{|\Omega_i|}{|\Omega_2|}P_2(t) + \dots + [\mathbf{A}_{\Delta t}]_{i,N}\frac{|\Omega_i|}{|\Omega_N|}P_N(t). \tag{3.49}$$

Hence, the factor $\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j}$ can be interpreted as the probability for a tracer's particle to end up in compartment i after a period Δt if it was initially in compartment j. This consideration suggests that the entries of $\mathbf{A}_{\Delta t}$ should be nonnegative: this is indeed an important property of the discrete interaction matrix that shall be proven shortly.

Property 3.4.1. *The entries of the discrete interaction matrix are nonnegative:*

$$[\mathbf{A}_{\Delta t}]_{ij} \geq 0 \textit{ for every } i,j = 1,\dots,N. \tag{3.50}$$

Proof. Let $C_0 > 0$ be a positive constant, and fix $j \in \{1,\dots,N\}$. Suppose that

$$\mathbf{c}(t) = \begin{pmatrix} c_1(t) \\ \vdots \\ c_{j-1}(t) \\ c_j(t) \\ c_{j+1}(t) \\ \vdots \\ c_N(t) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ C_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.51}$$

By equation (3.46),

$$\mathbf{c}(t+\Delta t) = \mathbf{A}_{\Delta t}\mathbf{c}(t) = \begin{pmatrix} [\mathbf{A}_{\Delta t}]_{1,j}C_0 \\ \vdots \\ [\mathbf{A}_{\Delta t}]_{N,j}C_0 \end{pmatrix}. \tag{3.52}$$

Since $\mathbf{c}$ is a concentration, we must have that $\mathbf{c}(t) \geq \mathbf{0}$ at any time t. Therefore, since $C_0 > 0$ we must have that

$$[\mathbf{A}_{\Delta t}]_{i,j} \geq 0 \text{ for every } i = 1, \ldots, N. \tag{3.53}$$

Finally, as we have made no assumption on j, this must be true for any $j \in \{1, \ldots, N\}$, which concludes the proof. □

The interpretation of $\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j}$ in terms of a transition probability of the tracer's particles between compartments suggests another important property of the discrete interaction matrix. Indeed, the tracer's particles can neither disappear nor be created, hence the sum of the probabilities over all possible destinations (i.e. over all compartments since we consider an isolated domain) must be equal to one:

$$\sum_{i=1}^{N} \frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j} = 1 \quad \text{for every } j = 1, \ldots, N. \tag{3.54}$$

This property can be deduced from property 3.2.1:

Property 3.4.2. *The columns of the discrete interaction matrix satisfy the following relation:*

$$\sum_{i=1}^{N} |\Omega_i|[\mathbf{A}_{\Delta t}]_{ij} = |\Omega_j| \quad \textit{for every } j = 1, \ldots, N. \tag{3.55}$$

In matrix form:

$$\boldsymbol{\omega}^\intercal \mathbf{A}_{\Delta t} = \boldsymbol{\omega}^\intercal. \tag{3.56}$$

Proof. The expression of the mean concentration over Ω is the same as in the continuous case:

$$\bar{C} = \frac{1}{|\Omega|}\mathbf{1}^\intercal \boldsymbol{\Omega}\mathbf{c}. \tag{3.57}$$

Since $\bar{C}$ is constant, we must have that

$$|\Omega|\bar{C}(t + \Delta t) = |\Omega|\bar{C}(t) = \mathbf{1}^\intercal \boldsymbol{\Omega}\mathbf{c}(t) = \boldsymbol{\omega}^\intercal \mathbf{c}(t), \tag{3.58}$$

but by equation (3.46) we also have that

$$|\Omega|\bar{C}(t + \Delta t) = \mathbf{1}^\intercal \boldsymbol{\Omega}\mathbf{c}(t + \Delta t) = \mathbf{1}^\intercal \boldsymbol{\Omega}\mathbf{A}_{\Delta t}\mathbf{c}(t) = \boldsymbol{\omega}^\intercal \mathbf{A}_{\Delta t}\mathbf{c}(t). \tag{3.59}$$

Hence, $\boldsymbol{\omega}^\intercal \mathbf{c}(t) = \boldsymbol{\omega}^\intercal \mathbf{A}_{\Delta t}\mathbf{c}(t)$, and since this must be true for every possible value of $\mathbf{c}(t)$, we must have that

$$\boldsymbol{\omega}^\intercal = \boldsymbol{\omega}^\intercal \mathbf{A}_{\Delta t}, \tag{3.60}$$

the desired result. □

Corollary 3.4.2. *If the compartments all have the same size, the discrete interaction matrix is left stochastic:*

$$\mathbf{1}^\intercal \mathbf{A}_{\Delta t} = \mathbf{1}^\intercal. \tag{3.61}$$

Proof. Let $|\Omega_0|$ be the size of the compartments. Then, $\boldsymbol{\omega} = |\Omega_0|\mathbf{1}$ and equation (3.60) reduces to

$$|\Omega_0|\mathbf{1}^\intercal = |\Omega_0|\mathbf{1}^\intercal \mathbf{A}_{\Delta t}. \tag{3.62}$$

□

Property 3.2.2 also has an interpretation in terms of the discrete interaction matrix:

Property 3.4.3. *The discrete interaction matrix is right stochastic:*

$$\mathbf{A}_{\Delta t}\mathbf{1} = \mathbf{1}. \tag{3.63}$$

Proof. Let $C_0 > 0$ be a constant and $\mathbf{c}(t) = C_0\mathbf{1}$. Introducing the latter in (3.46) yields

$$C_0\mathbf{1} = \mathbf{A}_{\Delta t}C_0\mathbf{1} = C_0\mathbf{A}_{\Delta t}\mathbf{1}. \tag{3.64}$$

□

Two last properties allow to bound the entries of the discrete interaction matrix:

Property 3.4.4. *The entries of the discrete interaction matrix are smaller than one:*

$$[\mathbf{A}_{\Delta t}]_{i,j} \leq 1 \textit{ for every } i, j = 1, \ldots, N. \tag{3.65}$$

Proof. We proceed by contradiction. Suppose there exists $i, j \in \{1, \ldots, N\}$ such that $[\mathbf{A}_{\Delta t}]_{i,j} > 1$. By property 3.4.1, all the entries of $\mathbf{A}_{\Delta t}$ are nonnegative so that

$$\sum_{k=1}^{N}[\mathbf{A}_{\Delta t}]_{i,k} > 1, \tag{3.66}$$

in contradiction with property 3.4.3. □

Property 3.4.5. *The entries of the discrete interaction matrix satisfy*

$$\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j} \leq 1 \textit{ for every } i, j = 1, \ldots, N. \tag{3.67}$$

Proof. We proceed again by contradiction. Suppose there exist $i, j \in \{1, \ldots, N\}$ such that $\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j} > 1$. By property 3.4.1, all the entries of $\mathbf{A}_{\Delta t}$ are nonnegative so that

$$\sum_{k=1}^{N}\frac{|\Omega_k|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{k,j} > 1, \tag{3.68}$$

in contradiction with property 3.4.2. □

This last property is in agreement with the interpretation of the factors $\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j}$ as probabilities of transition between the compartments.

Chapter 4

A Lagrangian solver for the two-dimensional advection-diffusion equation

Recall that the final goal of this work is to develop a method to automatically delineate subdomains that would be relevant compartments in a box model for a geophysical flow problem, and then to develop and assess compartment models based on the so calculated subdomains. The idea is to determine the boxes via the community detection method presented in chapter 1. In this work, we focus on *two-dimensional* problems involving *passive* tracers. The evolution of the concentration of such tracers is dictated by the advection-diffusion equation (2.2). In order to apply the community detection method, we need to emphasize the tracer's problem as a graph problem, and we aim to compute the transition probability matrix $\mathbf{M}(t)$ at some wisely chosen times t on that graph. To this end, the solution to the tracer's transport model (2.2) must be computed at different times and for different initial conditions. However, the advection-diffusion equation can in general not be solved analytically: the goal of this chapter is thus to provide all the material that is necessary to develop efficient numerical methods for the tracer's transport problem. A lot of different numerical methods have been developed through the years, each having their pros and cons. In the next paragraphs, we briefly summarize the main families of methods available, with their main advantages and disadvantages. This allows then to make an informed decision about which method to use in this work. Most of the discussion in the following paragraphs is inspired from [28] and [29].

A very popular class of numerical methods is formed by the Eulerian methods, in which the advection-diffusion equation is solved on a fixed spatial grid. This class encompass the finite difference method, finite element method and finite volume method. A second class is formed by the Lagrangian methods, where particles are followed through space at every time step. As we shall see later, the movement of an individual particle is modeled with a stochastic differential equation (SDE) which is consistent with the advection-diffusion equation. The idea is to estimate the concentration by simulating the trajectories of a large number of particles and taking averages. Several averaging methods have been developed to estimate the concentration from the set of particles positions: we shall come back to this further in section 4.1.8. Finally, a third class of mixed Eulerian-Lagrangian methods has been developed, which basically attempts to combine the advantages of both approaches. Such conceptually attractive methods have been widely applied in applications. They are however more complex to implement; in view of the relative simplicity of the two-dimensional tracer's transport problem studied in this work, we will not consider such mixed methods here.

Both Eulerian and Lagrangian methods have their own advantages and disadvantages. The Eulerian methods provide the convenience of a fixed grid and are easy to implement. The main drawbacks are the inherent dispersion errors and artificial oscillations, leading to solutions that may be neither mass conservative nor positive [30], [31]. Basically, the effect of dispersion errors is similar to physical dispersion but it is due to truncation errors. Artificial oscillations are typical from higher order methods designed to reduce dispersion errors. Those downsides could become excessively severe in case of problems involving a sharp concentration front, for instance advection-dominated problems or problems with large gradients on the initial concentration field (typically delta-like initial concentration) [32]. In those cases, numerical dispersion (i.e. dispersion due to dispersion errors) tends to inappropriately smooth out the sharp concentration front, whereas artificial oscillations tends to become more important, leading to serious problems with the positiveness of the solution.

On the other hand, Lagrangian methods ensures that the solution is always mass conservative and nonnegative. They are thus more suited than Eulerian methods to advection-dominated problems and to problems with large concentration gradients, since they do not suffer from dispersion errors and artificial oscillations. Moreover, if the tracer does not occupy the whole domain, the Lagrangian methods may be computationally more efficient than their Eulerian counterpart [33]. Depending on the number of particles used, Lagrangian methods may also require less storage than finite differences or finite element methods. Another advantage of Lagrangian methods is that they make it possible to advect the particles exactly when the velocity field can locally be described by an analytical function [34]. Finally, because each realization of the particle movement is independent from the others, Lagrangian methods are perfect candidates for parallelization. For instance, the MPI library makes it pretty easy and efficient to parallelize a code based on random walk models. Among the drawbacks of particle methods, the lack of a fixed grid may lead to numerical instability and computational difficulties [35]. If flow variables are not known analytically, their interpolation to the particle location could lead to local mass balance errors and solution anomalies [36]. Finally, the number of particles needed to get a smooth solution might be large leading to a large computational time, but this can be compensated by a parallelization of the code.

Considering the above discussion, it seems that a Lagrangian method is more appropriate for our concern. Indeed, in order to build an approximation of the transition probability matrix, the domain Ω is partitioned into grid cells which are the nodes of the graph. For a simulation time T, an entry $[\mathbf{M}(T)]_{ij}$ is the probability that a particle goes from grid cell i to grid cell j in a time T. To estimate the entries of a line i of the matrix, the idea is to run a simulation for a time T and an initial concentration which is uniform in grid cell i and zero in every other grid cell. The initial concentration is thus sharp, a first argument in favor of a Lagrangian method. Furthermore, the concentration is interpreted as a probability, hence positiveness and mass conservation are crucial topics, another point for Lagrangian methods. Finally, the flow variables are known analytically in the problems that will be considered in this work, which considerably reduce the drawbacks of Lagrangian methods pointed out above. For these reasons, we choose to go for a Lagrangian method.

4.1 Preliminaries: the theory of Stochastic Differential Equations

We introduce here the notions of stochastic differential equations (SDE's) and stochastic integrals. Those are the fundamental tools at the basis of the Lagrangian numerical methods. The discussion that comes next is based on several references, including [28] and [37–40]. Other references have been used for local parts of the work; they are then cited when they come in handy. Some results are stated without proofs. In such cases, unless otherwise stated, the reader may refer to [37] for formal proofs.

The idea behind the Lagrangian methods developed in this chapter is to estimate the concentration obeying an advection-diffusion-reaction equation by simulating the trajectories of a large number of particles in the flow. In this work, we restrict ourselves to advection diffusion equations of the form (2.2), which we recall here for the sake of readability:

$$\frac{\partial C}{\partial t} = \nabla \cdot (-\mathbf{u}C + \mathbf{K}\nabla C) \tag{4.1}$$

In the next, equation (4.1) will be referred to as the *transport model*. In order to implement a numerical method tracking the fates of individual particles, an equation describing the fate of such a particle must be derived, and that equation must be consistent with the transport model. Formally, the transport model can be interpreted as a Fokker-Planck equation, namely the partial differential equation governing the time evolution of the probability density function $p(\mathbf{x}, t)$ of the position of a particle. The correspondence is made by interpreting the concentration as the probability density function: $p = C$.

At the microscopic scale, Brownian diffusion is modeled by a stochastic force acting on the particles. This force is interpreted as the resultant of atomic bombardment on the particle. Intuitively, the direction of the force due to atomic bombardment is constantly changing, and at different times the particle is hit more on one side than another, leading to the seemingly random nature of the force, and hence of the motion. Therefore, the differential equation governing the position $\mathbf{x}(t)$ of a particle is stochastic. For example, Langevin proposed in 1908 an equation governing the position of a Brownian particle, which in one dimension can be written in the form :

$$\frac{dx}{dt} = a(x,t) + b(x,t)\xi(t), \tag{4.2}$$

where x is the position of the particle, $a(x,t)$ and $b(x,t)$ are known functions and $\xi(t)$ is the rapidly fluctuating random term. The simplest model is obtained by considering that $\xi(t)$ is a white noise, i.e.

$$\begin{cases} \langle \xi(t) \rangle = 0, & \text{(4.3a)} \\ \langle \xi(t)\xi(t') \rangle = \delta(t-t'), & \text{(4.3b)} \end{cases}$$

where $\langle \cdot \rangle$ denotes expectation. The fact that ξ has zero mean is because any nonzero mean can be absorbed in the term $a(x,t)$. The second condition states that $\xi(t)$ is uncorrelated, namely that the random force acting on a particle at a time is independent of the random forces acting on that particle at any other time. This simple form of the noise is justified by the fact that the correlation time t_c of the bombardment forces (of the order of 10^{-13} s for Brownian motion) is much smaller than the viscous time scale ($t_v \sim 10^{-9}$ s for Brownian motion) which is itself much smaller than the Brownian motion time scale ($t_B \sim 10^{-3}$ s). One must keep in mind that is an idealization of the atomic bombardment force. It is possible to show that

$$\int_0^t \xi(t')\mathrm{d}t' = W(t), \tag{4.4}$$

where $W(t)$ is the *Wiener process*, a *continuous* stochastic process defined by the following characteristics:

$$\begin{cases} W(0) = 0, & (4.5a) \\ W(t_2) - W(t_1) \sim \mathcal{N}(0, t_2 - t_1), & (4.5b) \\ \langle [W(t_4) - W(t_3)][W(t_2) - W(t_1)] \rangle = 0, & (4.5c) \end{cases}$$

where $t_1 < t_2 < t_3 < t_4$. In other words, $W(t)$ is a zero mean gaussian process of variance t which has the property of independent increments. Suppose now that $a(x,t) = a$ and $b(x,t) = b$ are constant. The above relations imply that the solution to (4.2) is

$$x(t) = at + bW(t), \tag{4.6}$$

where we implicitly assumed that $x(0) = 0$. However, one can show that the Wiener process is not differentiable with probability 1. We are thus faced with a paradox here since this implies that $x(t)$ is itself non-differentiable, and hence that the Langevin equation as stated in (4.2) *does not exist mathematically.* In fact, $\xi(t)$ is the derivative of $W(t)$ in the *distributive sense.* From (4.4), it follows directly that

$$\mathrm{d}W(t) \equiv W(t + \mathrm{d}t) - W(t) = \xi(t)\mathrm{d}t, \tag{4.7}$$

but it is incorrect (or at least very misleading) to write $\frac{dW(t)}{dt} = \xi(t)$, since the Wiener process is nowhere differentiable with probability 1, as already stated above.

Hopefully this introductory example shows the need for some preliminary steps in order to rigorously define a Stochastic Differential Equation (SDE), and to formalize the link between SDE's and Fokker-Planck equations. This is precisely the goal of the next pages.

4.1.1 Formal definition of a SDE

In this section, we restrict ourselves to a one-dimensional problem. This allows to make the notations less cumbersome while still introducing all the tools and concepts that are needed in order to understand the two-dimensional Lagrangian models which are at the basis of our numerical resolution of two-dimensional passive tracer's transport problems. Indeed, all the results presented here can almost straightforwardly be extended to several dimensions. Consider again the Langevin equation (4.2). We have shown in the introduction that the equation does not really make sense under that form. What we are now going to show is that the corresponding *integral equation*

$$x(t) = x(t_0) + \int_{t_0}^t a(x(s), s)\mathrm{d}s + \int_{t_0}^t b(x(s), s)\xi(s)\mathrm{d}s \tag{4.8}$$

can be interpreted consistently. The first integral is a standard Lebesgue integral of a function a of the stochastic process $x(t)$. The second integral has to be defined carefully, because of the presence of the white noise. By (4.7), we rewrite it as

$$\int_0^t b(x(s), s)\mathrm{d}W(s), \tag{4.9}$$

which is a kind of stochastic Stieltjes integral (see appendix A.1). Consider the following partition of the interval $[t_0,\, t]$:

$$t_0 \leq t_1 \leq t_2 \leq \cdots \leq t_{n-1} \leq t_n = t, \tag{4.10}$$

and define intermediate points $\tau_i \in [t_{i-1},\, t_i]$. Such a partition will often be used thereafter without introducing it explicitly every time it appears. The stochastic integral of b with respect to the Wiener process $\int_{t_0}^{t} b(x(s), s)\mathrm{d}W(s)$ is defined as a mean-square limit (see appendix A.2) of the partial sum

$$S_n = \sum_{i=1}^{n} b(x(\tau_i), \tau_i)[W(t_i) - W(t_{i-1})]. \tag{4.11}$$

Such a limit is not unique: it depends on the particular choice of τ_i. This sensibility to the choice of location $x(\tau_i)$ at which the function is evaluated is a consequence of the unbounded variation of the Wiener process. Popular choices are $\tau_i = t_{i-1}$, $\tau_i = \frac{1}{2}(t_{i-1} + t_i)$ or $\tau_i = t_i$, corresponding to the *Itô*, *Stratonovich* and *backward Itô* stochastic integrals, respectively. Hence, for a stochastic integral such as (4.9) to be well-defined, its *interpretation* must be stated explicitly. Namely, one must specify if the integral is taken in the *Itô*, *Stratonovich* or *backward Itô* sense, or any other interpretation corresponding to a given choice of τ_i. Only the Itô and backward Itô interpretations will be considered in this work. To avoid any confusion, we will denote the Itô integral by $(\mathbf{I})\!\int$ and the backward Itô integral by $(\mathbf{bI})\!\int$. Hence

$$(\mathbf{I})\int_{t_0}^{t} b(x(s), s)\mathrm{d}W(s) = \underset{n\to\infty}{\text{ms-lim}} \sum_{i=1}^{n} b(x(t_{i-1}), t_{i-1})[W(t_i) - W(t_{i-1})], \tag{4.12a}$$

$$(\mathbf{bI})\int_{t_0}^{t} b(x(s), s)\mathrm{d}W(s) = \underset{n\to\infty}{\text{ms-lim}} \sum_{i=1}^{n} b(x(t_i), t_i)[W(t_i) - W(t_{i-1})]. \tag{4.12b}$$

Remark In the backward Itô definition of the stochastic integral, only x needs to be evaluated at t_i: if $b(x,t)$ is differentiable in t (which is assumed for all the functions considered in this work), the integral is independent of the particular choice of value for t in the range $[t_{i-1},\, t_i]$. Hence, we could replace $b(x(t_i), t_i)$ by $b(x(t_i), \tau_i)$ with $\tau_i \in [t_{i-1},\, t_i]$ in the definition (4.12b). Note that this remark is not restricted to backward Itô integration: if b is differentiable in t, only the location $x(\tau_i)$ at which b is evaluated in the sum affects the limit.

Conventionally, a stochastic differential equation such as (4.2) is written in the form

$$\begin{cases} \mathrm{d}x(t) = a(x(t), t)\mathrm{d}t + b(x(t), t)\mathrm{d}W(t), \\ x(t_0) = x_0. \end{cases} \tag{4.13}$$

Equation (4.13) has to be interpreted as the implicit integral equation

$$x(t) = x_0 + \int_{t_0}^{t} a(x(s), s)\mathrm{d}s + \int_{t_0}^{t} b(x(s), s)\mathrm{d}W(s), \tag{4.14}$$

and the interpretation of the stochastic integral must thus be stated. For example, one will talk about an Itô SDE or a backward Itô SDE, and the stochastic process $x(t)$ which is the solution to the SDE (4.13) is different whether the stochastic integral is computed in the Itô or backward Itô sense.

It can be shown that an Itô stochastic integral $(\mathbf{I})\!\int_{t_0}^{t} G(s)\mathrm{d}W(s)$ exists whenever the function G is *continuous* and *nonanticipating* on the closed interval $[t_0,\, t]$. G is a *nonanticipating* function of t is for all t and s such that $t < s$, $G(t)$ is statistically independent of $W(s) - W(t)$. This condition seems obviously satisfied for any deterministic function $b(x(t), t)$ of the stochastic process $x(t)$ obeying the SDE (4.13), since $x(t)$ only depends on anterior values of the Wiener process. For example, in the context of the position of a brownian particle, it is intuitively obvious that the unknown future collisions cannot affect the present position of the particle. From now on, we assume thus that we are dealing with *nonanticipating* functions.

4.1.2 Properties of the Itô integral

When working with nonanticipating functions, it is possible to take advantage of the fact that $G(t_{i-1})$ is independent of $W(t_i) - W(t_{i-1})$ to derive particularly useful properties of the Itô integral. Such properties are generally not true for Stratonovich or backward Itô integrals. However, there is a formula called Ito's formula that allows to build rules to transform any SDE into an Itô SDE. Hence, when working with a Stratonovich or backward Itô SDE, it is often useful to transform that SDE into an equivalent Itô SDE so that the previously mentioned properties can be applied. In this section, we first derive those properties, and then we show Itô's formula for the change of variables. Those results are fundamental for connecting a SDE to the Fokker-Plank equation and for building consistent numerical schemes, as we shall see later.

Rules for the differentials

One preliminary result of first-order importance in the context of stochastic integration is that, for any nonanticipating function G

$$(\mathbf{I})\int_{t_0}^{t} G(s)[dW(s)]^2 = \int_{t_0}^{t} G(s)\mathrm{d}s, \tag{4.15}$$

i.e. that

$$\lim_{n\to\infty}\left\langle\left(\sum_{i=1}^{n} G(t_{i-1})[W(t_i) - W(t_{i-1})]^2 - \int_{t_0}^{t} G(s)\mathrm{d}s\right)^2\right\rangle = 0. \tag{4.16}$$

Let $\Delta W_i := W(t_i) - W(t_{i-1})$ and $\Delta t_i = t_i - t_{i-1}$. By the properties of the Wiener process, $\Delta W_i \sim \mathcal{N}(0, \Delta t_i)$. Hence, $\Delta W_i^2/\Delta t_i$ follows a χ-squared distribution with one degree of freedom. It has thus mean 1 and variance 2 and thus

$$\left\langle \Delta W_i^2 \right\rangle = \Delta t_i, \quad \text{and} \tag{4.17a}$$

$$\left\langle (\Delta W_i^2 - \Delta t_i)^2 \right\rangle = 2\Delta t_i^2. \tag{4.17b}$$

Using Riemann sums, the left hand side of equation (4.16) can be rewritten as

$$\lim_{n\to\infty}\left\langle\left(\sum_{i=1}^{n} G(t_{i-1})(\Delta W_i^2 - \Delta t_i)\right)^2\right\rangle, \tag{4.18}$$

which, by developing the square and using the linearity of the expectation operator is equal to

$$\lim_{n\to\infty}\sum_{i=1}^{n}\left[\left\langle G^2(t_{i-1})(\Delta W_i^2 - \Delta t_i)^2\right\rangle + \sum_{j=1}^{i-1}\left\langle 2G(t_{i-1})G(t_{j-1})(\Delta W_j^2 - \Delta t_j)(\Delta W_i^2 - \Delta t_i)\right\rangle\right]. \tag{4.19}$$

Since G is nonanticipating, we can use independence between terms to rewrite (4.19) as

$$\lim_{n\to\infty}\sum_{i=1}^{n}\Bigg[\Big\langle G^2(t_{i-1})\Big\rangle \underbrace{\Big\langle(\Delta W_i^2-\Delta t_i)^2\Big\rangle}_{=2\Delta t_i^2} \\ +\sum_{j=1}^{i-1}\Big\langle 2G(t_{i-1})G(t_{j-1})(\Delta W_j^2-\Delta t_j)\Big\rangle \underbrace{\Big\langle(\Delta W_i^2-\Delta t_i)\Big\rangle}_{=0}\Bigg]. \tag{4.20}$$

Notice that this step is only possible for Itô integration since otherwise we would not have independence. Equation (4.20) simplifies to

$$2\lim_{n\to\infty}\sum_{i=1}^{n}\Big\langle G^2(t_{i-1})\Big\rangle\,\Delta t_i^2. \tag{4.21}$$

If G is bounded on $[t_0,\, t]$, the latter goes to zero, which concludes the proof. Since $[\mathrm{d}W(t)]^2$ only appears in the context of stochastic integration, property (4.15) is often written

$$[\mathrm{d}W(t)]^2=\mathrm{d}t, \tag{4.22}$$

but one must not forget that the underlying meaning of this expression is relation (4.15), which is only valid in the context of Itô integration.

By a similar method, one can show that for any $N\geq 3$

$$[\mathrm{d}W(t)]^N=0, \tag{4.23}$$

and that for any $N_1\geq 1$, $N_2\geq 1$

$$[\mathrm{d}t]^{N_1}\,[\mathrm{d}W(t)]^{N_2}=0. \tag{4.24}$$

Those results are often summarized by saying that $\mathrm{d}W(t)$ is an infinitesimal of order $\frac{1}{2}$ in $\mathrm{d}t$ and that infinitesimals of order higher than 1 are discarded when it comes to compute differentials. Intuitively, $\mathrm{d}W(t)$ is a gaussian of variance $\mathrm{d}t$; a characteristic magnitude for $\mathrm{d}W(t)$ is its standard deviation, $\sqrt{\mathrm{d}t}$ which is indeed of order $\frac{1}{2}$.

The Itô formula

Consider an arbitrary function $\phi(x(t),t)$ with $x(t)$ obeying the Itô SDE (4.13). We are interested in the SDE governing ϕ. By definition,

$$\mathrm{d}\phi(x(t),t)=\phi(x(t)+\mathrm{d}x(t),t+\mathrm{d}t)-\phi(x(t),t). \tag{4.25}$$

Expanding $\phi(x(t)+\mathrm{d}x(t),t+\mathrm{d}t)$ in Taylor series and keeping the terms up to order 1 in $\mathrm{d}t$ yields

$$\begin{aligned}\mathrm{d}\phi(x,t)&=\frac{\partial\phi}{\partial t}\mathrm{d}t+\frac{\partial\phi}{\partial x}\mathrm{d}x(t)+\frac{1}{2}\frac{\partial^2\phi}{\partial x^2}[\mathrm{d}x(t)]^2+\dots\\&=\frac{\partial\phi}{\partial t}\mathrm{d}t+\frac{\partial\phi}{\partial x}\left(a\mathrm{d}t+b\mathrm{d}W(t)\right)+\frac{1}{2}\frac{\partial^2\phi}{\partial x^2}\Big(b^2[\mathrm{d}W(t)]^2+\dots\Big)+\dots,\end{aligned}$$

where a, b and the derivatives are evaluated at $(x(t),t)$. Now, using relation (4.22), we get the Itô formula:

$$\mathrm{d}\phi(x(t),t)=\left[\frac{\partial\phi}{\partial t}+\frac{\partial\phi}{\partial x}a(x(t),t)+\frac{1}{2}\frac{\partial^2\phi}{\partial x^2}b^2(x(t),t)\right]\mathrm{d}t+\frac{\partial\phi}{\partial x}b(x(t),t)\mathrm{d}W(t). \tag{4.26}$$

4.1.3 Link between Itô and backward Itô SDE's

A same stochastic process $x(t)$ can be described both by a Itô and by a backward Itô SDE. Suppose that $x(t)$ obeys the Itô SDE

$$dx(t) = a_I(x(t),t)\mathrm{d}t + b_I(x(t),t)\mathrm{d}W(t), \tag{4.27}$$

and the equivalent backward Itô SDE

$$dx(t) = a_{bI}(x(t),t)\mathrm{d}t + b_{bI}(x(t),t)\mathrm{d}W(t). \tag{4.28}$$

The goal is to compute the relations between the functions a_I, b_I and a_{bI}, b_{bI}. By (4.28),

$$x(t) = x(t_0) + \int_{t_0}^{t} a_{bI}(x(s),s)\mathrm{d}s + (\mathbf{bI})\int_{t_0}^{t} b_{bI}(x(t),t)\mathrm{d}W(t). \tag{4.29}$$

Let us rewrite the backward Itô stochastic integral term

$$(\mathbf{bI})\int_{t_0}^{t} b_{bI}(x(t),t)\mathrm{d}W(t) \triangleq \underset{n\to\infty}{\text{ms-lim}} \sum_{i=1}^{n} b_{bI}(x(t_i),t_i)[W(t_i) - W(t_{i-1})] \tag{4.30}$$

as an Itô integral. Since $x(t)$ satisfies the Itô SDE (4.27), we can apply Itô formula. This yields

$$\begin{aligned} b_{bI}(x(t_i),t_i) = {} & b_{bI}(x(t_{i-1}),t_{i-1}) \\ & + \left[\frac{\partial b_{bI}}{\partial t} + \frac{\partial b_{bI}}{\partial x} a_I(x(t_{i-1}),t_{i-1}) + \frac{1}{2}\frac{\partial^2 b_{bI}}{\partial x^2} b_I^2(x(t_{i-1}),t_{i-1})\right](t_i - t_{i-1}) \\ & + \frac{\partial b_{bI}}{\partial x} b_I(x(t_{i-1}),t_{i-1})(W(t_i) - W(t_{i-1})), \end{aligned} \tag{4.31}$$

where the derivatives are evaluated at $(x(t_{i-1}),t_{i-1})$. Introducing (4.31) in (4.30) and setting $[\mathrm{d}W(t)]^2 = \mathrm{d}t$ yields, after dropping the terms in $\mathrm{d}t\mathrm{d}W(t)$ and $\mathrm{d}t^2$:

$$\begin{aligned} (\mathbf{bI})\int_{t_0}^{t} b_{bI}(x(t),t)\mathrm{d}W(t) = \underset{n\to\infty}{\text{ms-lim}} \sum_{i=1}^{n} \Big(& b_{bI}(x(t_{i-1}),t_{i-1})[W(t_i) - W(t_{i-1})] \\ & + b_I(x(t_{i-1}),t_{i-1})\frac{\partial b_{bI}}{\partial x}[t_i - t_{i-1}]\Big), \end{aligned} \tag{4.32}$$

and finally

$$(\mathbf{bI})\int_{t_0}^{t} b_{bI}(x(t),t)\mathrm{d}W(t) = (\mathbf{I})\int_{t_0}^{t} b_{bI}(x(t),t)\mathrm{d}W(t) + \int_{t_0}^{t} b_I(x(s),s)\frac{\partial b_{bI}}{\partial x}(x(s),s)\mathrm{d}s. \tag{4.33}$$

Therefore, we have the equivalences

$$\begin{array}{ccc} \underline{\text{Itô SDE}} & & \underline{\text{backward Itô SDE}} \\ \mathrm{d}x(t) = a_I\mathrm{d}t + b_I\mathrm{d}W(t) & \Leftrightarrow & \mathrm{d}x(t) = \left[a_I - b_I\dfrac{\partial b_I}{\partial x}\right]\mathrm{d}t + b_I\mathrm{d}W(t) \end{array} \tag{4.34}$$

and conversely

$$\begin{array}{ccc} \underline{\text{backward Itô SDE}} & & \underline{\text{Itô SDE}} \\ \mathrm{d}x(t) = a_{bI}\mathrm{d}t + b_{bI}\mathrm{d}W(t) & \Leftrightarrow & \mathrm{d}x(t) = \left[a_{bI} + b_{bI}\dfrac{\partial b_{bI}}{\partial x}\right]\mathrm{d}t + b_{bI}\mathrm{d}W(t). \end{array} \tag{4.35}$$

Here, the dependence of a_I, b_I, a_{bI} and b_{bI} on $x(t)$ and t have been made implicit to simplify the notations.

4.1.4 Connection between Itô and backward Itô SDE's and the Fokker Planck equation

Consider a particle in one dimension whose position $x(t)$ obeys the Itô SDE

$$(\mathbf{I})\ \begin{cases} \mathrm{d}x(t) = a(x(t),t)\mathrm{d}t + b(x(t),t)\mathrm{d}W(t), \\ x(t_0) = x_0. \end{cases} \tag{4.36}$$

Let $p(x,t;y,s)$ be the probability density function of the position x of the particle at time t given that the particle was in position y at time s, with $s < t$. For an infinitesimal $\mathrm{d}x$ and $\bar{x} \in \Omega$, the probability that the random variable $x(t)$ describing the position of the particle at time t has value between $\bar{x}$ and $\bar{x} + \mathrm{d}x$ is given by:

$$\Pr\left(\bar{x} < x(t) < \bar{x} + \mathrm{d}x \,\middle|\, x(s) = y\right) = p(\bar{x},t;y,s)\mathrm{d}x. \tag{4.37}$$

From (4.36) we are going the derive the partial differential equation governing the evolution of $p(x,t;x_0,t_0)$. Let $K(x)$ be an arbitrary function of $\mathcal{C}^2$ with compact support. In the next, all the derivative terms of functions a and b are evaluated at $(x(t),t)$ and the derivatives of K are evaluated at $x(t)$, unless otherwise stated. By Itô's formula:

$$\mathrm{d}K(x(t)) = \left[\frac{\partial K}{\partial x}a(x(t),t) + \frac{1}{2}b^2(x(t),t)\frac{\partial^2 K}{\partial x^2}\right]\mathrm{d}t + \frac{\partial K}{\partial x}b(x(t),t)\mathrm{d}W(t). \tag{4.38}$$

Taking the expectation yields

$$\langle \mathrm{d}K(x(t))\rangle = \left\langle \frac{\partial K}{\partial x}a(x(t),t) + \frac{1}{2}b^2(x(t),t)\frac{\partial^2 K}{\partial x^2}\right\rangle \mathrm{d}t, \tag{4.39}$$

and thus

$$\frac{d\,\langle K(x(t))\rangle}{dt} = \left\langle \frac{\partial K}{\partial x}a(x(t),t) + \frac{1}{2}b^2(x(t),t)\frac{\partial^2 K}{\partial x^2}\right\rangle. \tag{4.40}$$

Using the probability density $p(x,t;x_0,t_0)$, we can rewrite (4.40) as

$$\begin{aligned}\frac{d}{dt}\int_{-\infty}^{\infty} K(x)p(x,t;x_0,t_0)\mathrm{d}x = &\underbrace{\int_{-\infty}^{\infty}\frac{\partial K}{\partial x}a(x,t)p(x,t;x_0,t_0)\mathrm{d}x}_{:=I_1} \\ &+ \frac{1}{2}\underbrace{\int_{-\infty}^{\infty}\frac{\partial^2 K}{\partial x^2}b^2(x,t)p(x,t;x_0,t_0)\mathrm{d}x}_{:=I_2}.\end{aligned} \tag{4.41}$$

Integrating I_1 by parts yields:

$$\begin{aligned} I_1 &= \left[a(x,t)p(x,t;x_0,t_0)K(x)\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} K(x)\frac{\partial(ap)}{\partial x}\mathrm{d}x \\ &= -\int_{-\infty}^{\infty} K(x)\frac{\partial(ap)}{\partial x}\mathrm{d}x, \end{aligned} \tag{4.42}$$

where the second equality is obtained because K has a compact support. Integrating I_2 by parts yields successively:

$$\begin{aligned} I_2 &= \left[b^2(x,t)p(x,t;x_0,t_0)\frac{\partial K}{\partial x}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty}\frac{\partial K}{\partial x}\frac{\partial(b^2 p)}{\partial x} \\ &= \left[b^2(x,t)p(x,t;x_0,t_0)\frac{\partial K}{\partial x}\right]_{-\infty}^{\infty} - \left[\frac{\partial(b^2 p)}{\partial x}K(x)\right]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} K(x)\frac{\partial^2(b^2 p)}{\partial x^2}\mathrm{d}x \\ &= \int_{-\infty}^{\infty} K(x)\frac{\partial^2(b^2 p)}{\partial x^2}\mathrm{d}x. \end{aligned} \tag{4.43}$$

Again, the surface terms vanish because of the compact support of K. Inserting (4.42) and (4.43) in (4.41) and rearranging the terms yields

$$\int_{-\infty}^{\infty} K(x) \left(\frac{\partial p}{\partial t} + \frac{\partial (ap)}{\partial x} - \frac{1}{2} \frac{\partial^2 (b^2 p)}{\partial x^2} \right) \mathrm{d}x = 0. \tag{4.44}$$

Since K is arbitrary we must have that

$$\frac{\partial p}{\partial t} = -\frac{\partial (ap)}{\partial x} + \frac{1}{2} \frac{\partial^2 (b^2 p)}{\partial x^2}, \tag{4.45}$$

the Fokker-Planck equation (or Kolmogorov forward equation) corresponding to the Itô SDE (4.36).

Now suppose that the particle's position $x(t)$ obeys the backward Itô SDE

$$(\mathbf{bI}) \begin{cases} \mathrm{d}x(t) = a(x(t), t)\mathrm{d}t + b(x(t), t)\mathrm{d}W(t), \\ x(t_0) = x_0. \end{cases} \tag{4.46}$$

By (4.35), $x(t)$ is equivalently governed by the Itô SDE

$$(\mathbf{I}) \begin{cases} \mathrm{d}x(t) = \left(a(x(t), t) + b(x(t), t) \dfrac{\partial b}{\partial x} \right) \mathrm{d}t + b(x(t), t)\mathrm{d}W(t) \\ x(t_0) = x_0. \end{cases} \tag{4.47}$$

By the above results, the probability density of $x(t)$ is governed by the partial differential equation

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x} \left[\left(a + b\frac{\partial b}{\partial x} \right) p \right] + \frac{1}{2} \frac{\partial^2 (b^2 p)}{\partial x^2}. \tag{4.48}$$

The latter can be simplified to

$$\frac{\partial p}{\partial t} = -\frac{\partial (ap)}{\partial x} + \frac{1}{2} \frac{\partial}{\partial x} \left(b^2 \frac{\partial p}{\partial x} \right), \tag{4.49}$$

the Fokker-Planck equation corresponding to the backward Itô SDE (4.46).

4.1.5 Generalization to multiple dimensions

Here we generalize the results of previous sections to the cases with n variables and m independent noise components. In that case, $\mathbf{x}(t)$ and $\mathbf{a}(\mathbf{x}, t)$ are n-dimensional vectors, $\mathbf{B}(\mathbf{x}, t)$ is a $n \times m$ matrix and $\mathbf{W}(t)$ is a m-dimensional multivariate Wiener process of mean $\mathbf{0}$. Hence, $\mathbf{W}(t) = [W_1(t), W_2(t), \ldots, W_m(t)]^\intercal$ is a vector of m *independent* Wiener processes such as defined in (4.5). The general form of a multidimensional SDE is then

$$\begin{cases} \mathrm{d}\mathbf{x}(t) = \mathbf{a}(\mathbf{x}(t), t)\mathrm{d}t + \mathbf{B}(\mathbf{x}(t), t)\mathrm{d}\mathbf{W}(t), \\ \mathbf{x}(t_0) = \mathbf{x}_0. \end{cases} \tag{4.50}$$

The Itô differential rules are similar to the one-dimensional case. For $N \geq 3$, N_1, $N_2 \geq 1$ and $i, j \in \{1, 2, \ldots, m\}$, we have (in the context of Itô integration):

$$\begin{cases} \mathrm{d}W_i(t)\mathrm{d}W_j(t) = \delta_{ij}\mathrm{d}t, & \text{(4.51a)} \\ [\mathrm{d}W_i(t)]^N = 0, & \text{(4.51b)} \\ [\mathrm{d}t]^{N_1} [\mathrm{d}W_i(t)]^{N_2} = 0. & \text{(4.51c)} \end{cases}$$

Itô's formula for a function ϕ of the n-dimensional vector $\mathbf{x}(t)$ satisfying the Itô SDE (4.50) is

$$\begin{aligned} \mathrm{d}f(\mathbf{x}) = & \left[\sum_{i=1}^{n} a_i(\mathbf{x}(t),t)\frac{\partial f}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\mathbf{B}(\mathbf{x}(t),t)\mathbf{B}^{\mathsf{T}}(\mathbf{x}(t),t)]_{ij}\frac{\partial^2 f}{\partial x_i \partial x_j}\right]\mathrm{d}t \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}[\mathbf{B}(\mathbf{x}(t),t)]_{ij}\frac{\partial f}{\partial x_i}\mathrm{d}W_j(t), \end{aligned} \tag{4.52}$$

where the derivatives of f are evaluated at $\mathbf{x}(t)$.

Let $\mathbf{D} = \mathbf{B}\mathbf{B}^{\mathsf{T}}$. If (4.50) is interpreted as an Itô SDE, the corresponding Fokker-Planck equation is

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{n}\frac{\partial(a_i p)}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial^2\left([\mathbf{D}]_{ij}p\right)}{\partial x_i \partial x_j}. \tag{4.53}$$

If (4.50) is interpreted as a backward Itô SDE, the corresponding Fokker-Planck equation is

$$\frac{\partial p}{\partial t} = -\sum_{i=1}^{n}\frac{\partial(a_i p)}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial}{\partial x_i}\left([\mathbf{D}]_{ij}\frac{\partial p}{\partial x_j}\right). \tag{4.54}$$

In equations (4.53) and (4.54), all the derivatives are evaluated at $(\mathbf{x}(t), t)$.

Generally, if we have a Fokker-Planck equation and want to compute a corresponding SDE, we have to compute $\mathbf{B}$ from $\mathbf{D} = \mathbf{B}\mathbf{B}^{\mathsf{T}}$. The solution to that equation is not unique, so that SDE's with different $\mathbf{B}$ could be consistent with the same Fokker-Planck equation. If $\mathbf{D}$ is positive-semidefinite, the Cholesky decomposition *exists*, namely there is a *lower triangular* matrix $\mathbf{B}$ which is the solution to $\mathbf{D} = \mathbf{B}\mathbf{B}^{\mathsf{T}}$. Besides, if $\mathbf{D}$ is positive-definite, the Cholesky decomposition is *unique* if we require that the diagonal elements of $\mathbf{B}$ are strictly positive. This provides a way to compute $\mathbf{B}$ in the case of a positive-semidefinite matrix $\mathbf{D}$.

Remark The Itô and backward Itô SDE's considered in this section have exactly the same form, given by equation (4.50). However, if $\mathbf{B}$ is not constant, they correspond to different Fokker-Planck equations and thus to different *transport processes.* An important implication that does not arise in the context of deterministic integration is that the numerical method has to be chosen consistently with the type (Itô, backward Itô, Stratonovich, etc.) of the SDE. The derivation of simple numerical schemes consistent with either an Itô or a backward Itô SDE is the topic of the next section.

4.1.6 Numerical methods

We present here the *Euler* and *backward Euler* methods, which are the simplest numerical schemes for the simulation of an Itô and a backward Itô stochastic process, respectively. The goal of this section is really to provide some intuition about why the *Euler method* is relevant to simulate an Itô process whereas the *backward Euler method* is relevant in the context of a backward Itô process. We do not aim at providing fully rigorous proofs of convergence and consistency. Such a formalism can be found in the well-known book by Kloeden and Platen about stochastic numerical methods [41]. A more recent handbook about stochastic numerical methods is [42], and a very nice introductory article is found in [43]. The backward Euler method has been introduced more recently by LaBolle [44].

Consider the one-dimensional SDE

$$\begin{cases} \mathrm{d}x(t) = a(x(t), t)\mathrm{d}t + b(x(t), t)\mathrm{d}W(t), \\ x(t_0) = x_0. \end{cases} \tag{4.55}$$

Through a numerical approximation of (4.55), we can only compute x at discrete times $t_0 < t_1 < \cdots < t_{n-1} < t_n = T$, where T is the final integration time. We consider a constant time step Δt such that $t_{i+1} = t_i + \Delta t$ for any $i \in \{0, 1, \ldots, n-1\}$. Let X_i denote the numerical approximation of $x(t_i)$, and let $\Delta W_{i+1} := W(t_{i+1}) - W(t_i) \sim \mathcal{N}(0, \Delta t)$. ΔW_i is thus a gaussian noise of mean 0 and variance Δt, and ΔW_i is independent of ΔW_j for any $i \neq j$. In the next, we shall only verify that the schemes are *consistent*, namely that they tend to the proper SDE when $t_i \to t$, $\Delta t \to \mathrm{d}t$ and $\Delta W_i \to \mathrm{d}W(t)$.

Let us first consider the case where (4.55) is a Itô SDE. Suppose that $x(t_i)$ is known, and we want to compute $x(t_{i+1})$. Now, consider a partition $t_i = t_{i_0} < t_{i_1} < \cdots < t_{i_{m-1}} < t_{i_m} = t_{i+1}$ of $[t_i,\, t_{i+1}]$. By (4.55) and the definition of the Itô stochastic integral:

$$\begin{aligned} x(t_{i+1}) = x(t_i) + \lim_{m\to\infty} \sum_{k=0}^{m-1} a\left(x(t_{i_k}), t_{i_k}\right)\left(t_{i_{k+1}} - t_{i_k}\right) \\ + \operatorname*{ms-lim}_{m\to\infty} \sum_{k=0}^{m-1} b\left(x(t_{i_k}), t_{i_k}\right)[W(t_{i_{k+1}}) - W(t_{i_k})]. \end{aligned} \tag{4.56}$$

The simplest approximation to that expression is to take $m = 1$. This gives precisely the *Euler method*, also called the *Euler-Maruyama method*:

$$X_{i+1} = X_i + a(X_i, t_i)\Delta t + b(X_i, t_i)\Delta W_{i+1} \tag{4.57}$$

Note that if $R_0, R_1, \ldots, R_{n-1}$ are independent standard gaussian random variables, then we can replace ΔW_{i+1} by $\sqrt{\Delta t} R_i$ which could be more practical to implement.

Now consider the case where (4.55) is a backward Itô SDE. A similar reasoning as set out above for the Itô case yields :

$$\begin{aligned} x(t_{i+1}) = x(t_i) + \lim_{m\to\infty} \sum_{k=0}^{m-1} a\left(x(t_{i_k}), t_{i_k}\right)\left(t_{i_{k+1}} - t_{i_k}\right) \\ + \operatorname*{ms-lim}_{m\to\infty} \sum_{k=0}^{m-1} b\left(x(t_{i_{k+1}}), t_{i_k}\right)[W(t_{i_{k+1}}) - W(t_{i_k})]. \end{aligned} \tag{4.58}$$

Notice that since the first limit corresponds to a deterministic integral, we can choose to evaluate a at any time in $[t_{i_k},\, t_{i_{k+1}}]$.[1] The fact that b is evaluated at time t_{i_k} follows from the implicit assumption that b is differentiable in t, cfr. the remark at page 29. Taking $m = 1$ yields the approximation

$$X_{i+1} = X_i + a(X_i, t_i)\Delta t + b(X_{i+1}, t_i)\Delta W_{i+1}, \tag{4.59}$$

which is an *implicit* scheme since b has to be evaluated at X_{i+1}. In the general case b is nonlinear in x and solving (4.59) is nontrivial. In particular, it is not always possible to

[1] The choice $a(x(t_{i_k}), t_{i_k})$ corresponds to the Darboux integration, which can be seen as a particular case of the Riemann integration. Remember that a function is Darboux-integrable if and only if it is Riemann-integrable, and the values of the two integrals, if they exist, are equal.

invert b and hence to find an explicit formula for X_{i+1}. The idea is thus to rely on a predictor-corrector method: we first compute an estimate X^*_{i+1} of X_{i+1} using an explicit formula, and then we compute X_{i+1} as

$$X_{i+1} = X_i + a(X_i, t_i)\Delta t + b(X^*_{i+1}, t_i)\Delta W_{i+1}. \tag{4.60}$$

Now the question is: how to compute X^*_{i+1} ? One might be tempted to use the Euler method, which is explicit. However, we will see shortly that we do not need to include the advective transport term in the estimation X^*_{i+1} of X_{i+1}. The predictor-corrector scheme is thus

$$\begin{cases} \Delta Y_{i+1} = b(X_i, t_i)\Delta W_{i+1}, & \text{(4.61a)} \\ X_{i+1} = X_i + a(X_i, t_i)\Delta t + b(X_i + \Delta Y_{i+1}, t_i)\Delta W_{i+1}. & \text{(4.61b)} \end{cases}$$

In order to see that the scheme (4.61) consistently approximate the backward Itô SDE (4.55), we can consider a stochastic process $y(t)$ and interpret ΔY_{i+1} as the difference between two successive iterates: $\Delta Y_{i+1} = Y_{i+1} - Y_i$. But then, the scheme (4.61) is precisely the *Euler method* applied on the Itô SDE with two variables

$$(\mathbf{I}) \begin{cases} \mathrm{d}y(t) = b(x(t), t)\mathrm{d}W(t), & \text{(4.62a)} \\ \mathrm{d}x(t) = a(x(t), t)\mathrm{d}t + b(x(t) + \mathrm{d}y(t), t)\mathrm{d}W(t). & \text{(4.62b)} \end{cases}$$

Therefore, by the equivalence formula between Itô and backward Itô SDE's (4.35), showing that the scheme (4.61) is consistent with the backward Itô SDE (4.55) amounts to show that (4.62) is equivalent to the Itô SDE

$$\mathrm{d}x(t) = \left(a(x(t), t) + b(x(t), t)\frac{\partial b}{\partial x}\right)\mathrm{d}t + b(x(t), t)\mathrm{d}W(t), \tag{4.63}$$

where $\partial_x b$ is evaluated at $(x(t), t)$. Using the same convention for all the derivatives of b, a stochastic Itô-Taylor expansion on $b(x(t) + \mathrm{d}y(t), t)$ yields

$$\begin{aligned} b(x + \mathrm{d}y(t), t) &= b(x, t) + \frac{\partial b}{\partial x}\mathrm{d}y(t) + \frac{1}{2}\frac{\partial^2 b}{\partial x^2}[\mathrm{d}y(t)]^2 + \dots & \text{(4.64a)} \\ &= b(x, t) + \frac{\partial b}{\partial x}\mathrm{b}(x, t)\mathrm{d}W(t) + \frac{1}{2}\frac{\partial^2 b}{\partial x^2}b^2(x, t)[\mathrm{d}W(t)]^2 + \dots & \text{(4.64b)} \\ &= b(x, t) + \frac{\partial b}{\partial x}\mathrm{b}(x, t)\mathrm{d}W(t) + \frac{1}{2}\frac{\partial^2 b}{\partial x^2}b^2(x, t)\mathrm{d}t + \dots & \text{(4.64c)} \end{aligned}$$

where we have noted x instead of $x(t)$ in order to save space. Finally, introducing (4.64c) in (4.62b) yields

$$\begin{aligned} \mathrm{d}x(t) &= a(x, t)\mathrm{d}t + b(x, t)\mathrm{d}W(t) + \frac{\partial b}{\partial x}\mathrm{b}(x, t)[\mathrm{d}W(t)]^2 + \frac{1}{2}\frac{\partial^2 b}{\partial x^2}b^2(x, t)\mathrm{d}t\mathrm{d}W(t), & \text{(4.65a)} \\ &= \left(a(x, t) + b(x, t)\frac{\partial b}{\partial x}\right)\mathrm{d}t + b(x, t)\mathrm{d}W(t), & \text{(4.65b)} \end{aligned}$$

where we have used the properties (4.22) and (4.24) of Itô integration. Since (4.65b) is exactly (4.63), we have shown the consistency of the backward Euler method (4.61) for the numerical integration of the backward Itô SDE (4.55). Notice that including the advective transport in the computation of ΔY_{i+1} in equation (4.61a) leads to a term of order $\mathrm{d}t\mathrm{d}W$ in equation (4.65a). By (4.24), $\mathrm{d}t\mathrm{d}W = 0$ and this term does not appear in (4.65b).

Remark We have implicitly assumed through all the above developments that b is *smooth enough*, and the equivalence between the Itô and backward Itô formulation has only be proven in that case. In the case of discontinuous b, the consistency of the backward Itô formulation with the Fokker-Planck equation is shown in [44] for the one-dimensional case and demonstrated in the two-dimensional case. There is however no proof for the multi-dimensional case. The efficacy of the backward Euler method in the case of discontinuous diffusivities is assessed in [29] on two one-dimensional test cases. The second test case is an advection-diffusion equation with constant velocity and a piecewise constant diffusivity. It shows a significantly better performance of the backward Euler method with respect to Itô and Stratonovich methods in estimating the residence time of a tracer.

Strong and Weak convergence of the schemes

Let us now present briefly the notion of strong and weak convergence for the numerical approximation of a SDE. The difference between those two definitions of the convergence resides in the criterion used to measure the difference between the exact solution $X(t_i)$ and its numerical approximation X_i. Using the expected value $\langle |X(t_i) - X_i| \rangle$ leads to the concept of *strong convergence*. Formally, a method has a strong order of convergence equal to γ if there exists a constant β such that

$$\langle |X(t) - X_i| \rangle \leq \beta \Delta t^{\gamma} \tag{4.66}$$

at any fixed $t = i\Delta t$ and Δt sufficiently small. Hence, the strong order of convergence measure the rate at which the average error decreases when Δt goes to zero.

An alternative, less demanding definition of convergence is obtained if we measure the "error of the means" instead of the "mean of the error". This leads to the concept of *weak convergence*. Formally, a method has a weak order of convergence equal to γ if there exists a constant β such that

$$|\, \langle f(X(t)) \rangle - \langle f(X_i) \rangle \,| \leq \beta \Delta t^{\gamma} \tag{4.67}$$

at any fixed $t = i\Delta t$ and Δt sufficiently small, and where the function f is an arbitrary function with polynomial growth.

Under appropriate conditions on the functions a and b, it can be shown that the Euler-Maruyama and the backward Euler scheme both have strong order of convergence $\frac{1}{2}$ and weak order of convergence 1.

4.1.7 Dealing with no-through boundary conditions

Only no-through boundary conditions are encountered in the problems that we will consider. Such conditions are enforced by bouncing particles off boundaries, as suggested in [45]. Under such conditions, no particle can leave the domain. Bouncing is assumed to be perfect, and we only consider straight lines boundaries. Hence, bouncing amounts to remap particles leaving the domain into the domain by an orthogonal symmetry with the boundary.

4.1.8 Computation of the concentration

The above discussions shows how to simulate individual particles trajectories. However, what we ultimately want is to properly estimate the solution to (4.1), namely to compute an approximation of the concentration from the particles locations. Several methods exists, amongst which the *box counting method* and the *kernel density estimation*. Both methods are discussed in [28] and [46].

The *box counting* method has been extensively used to estimate the concentration in studies using random walk modeling, see e.g. [47]. The method goes as follows: suppose a grid discretization of the domain, the *box counting methods* consists in counting the number of particles in a grid cell[2]; the estimation of the concentration is then obtained by multiplying the number of particles with their mass, and dividing this total mass by the volume of the grid cell. This method amounts to compute histograms: the concentration profile computed this way is constant in each grid cell and discontinuous at the boundaries between the grid cells. Such an estimation of the concentration depends thus on the size and center of the grid cells, but there is unfortunately no physical argument that would allow to choose those parameters. The choice of the averaging volumes is thus often made empirically as the result of a trade-off between regularity and a satisfying resolution of the solution. On one hand, choosing large grid cells could lead to the loss of important details since the estimated concentration cannot be described in a grid cell more precisely than a constant. Hence the solution tends to be oversmoothed when the number of grid cells is too small. On the other hand, opting for many small grid cells lead to a concentration profile that tends to become very irregular or noisy. Another disadvantage of the method is that the number of particles needed to make the uncertainties becomes negligible might be excessively large. However, this method is the most efficient for problem where the goal is precisely to compute volume averages over grid cells; but if one aims at computing point concentration, then the kernel density estimation turns out to be more efficient.

The *kernel density estimation* method allows to reduce drastically the number of particles, and does not suffer from the resolution limit inherent to the box counting method. However, the kernel estimation method introduces some difficulties. A first difficulty is that this method depends on a smoothing parameter called the *bandwidth* for which finding a relevant value is not trivial. Second, the method as such does not perform well at the boundaries, and a specific treatment of the boundaries must be introduced. Classical references on the density kernel estimation are [48] and [49].

For the purposes of this work, we will exclusively be interested in volume averages of the concentration over grid cells so that only the box-counting method will be used. Indeed, grid cells are precisely the nodes of the underlying network, and the transition probabilities between the nodes are estimated from the concentration in the grid cells.

4.2 Lagrangian equations corresponding to the advection-diffusion transport model

The previous section introduces all the theoretical tools needed to compute the Lagrangian equations corresponding to the transport model, namely the general advection-

[2] Such grid cells are also called *boxes* in the literature. To avoid confusion, we prefer to reserve the term "box" to design compartments in this work.

diffusion equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (-\mathbf{u}C + \mathbf{K}\nabla C), \tag{4.68}$$

where $\mathbf{K}$ is the symmetric and positive-definite diffusivity tensor. From now on, we restrict ourselves to two dimensions in the cartesian coordinate system (y, z). The transport equation (4.68) can be interpreted as a Fokker-Planck equation where C is the probability density function of the position $\mathbf{x}(t) = (y(t),\, z(t))$ of the particle. Let

$$\mathbf{K} = \begin{pmatrix} K_{yy} & K_{yz} \\ K_{zy} & K_{zz} \end{pmatrix}, \tag{4.69}$$

with $K_{yz} = K_{zy}$ since $\mathbf{K}$ is symmetric. In order to get the Itô and backward-Itô systems of SDEs corresponding to the transport model, we must rewrite (4.68) in the forms (4.53) and (4.54) respectively. The systems of SDEs can then be deduced straightforwardly by analogy.

Let us first compute the Itô SDEs corresponding to (4.68). Equation (4.68) can be rewritten as

$$\begin{aligned} \frac{\partial C}{\partial t} = &-\frac{\partial}{\partial y}\left[\left(v + \frac{\partial K_{yy}}{\partial y} + \frac{\partial K_{yz}}{\partial z}\right)C\right] - \frac{\partial}{\partial z}\left[\left(w + \frac{\partial K_{zy}}{\partial y} + \frac{\partial K_{zz}}{\partial z}\right)C\right] \\ &+ \frac{1}{2}\left[\frac{\partial^2}{\partial y^2}(2K_{yy}C) + \frac{\partial^2}{\partial y \partial z}(2K_{yz}C) + \frac{\partial^2}{\partial z \partial y}(2K_{zy}C) + \frac{\partial^2}{\partial z^2}(2K_{zz}C)\right]. \end{aligned} \tag{4.70}$$

This is precisely equation (4.53) with $\mathbf{x} = (y, z)$, $p = C$, $\mathbf{a} = (v + \partial_y K_{yy} + \partial_z K_{yz},\, w + \partial_y K_{zy} + \partial_z K_{zz})$ and $\mathbf{D} = 2\mathbf{K}$. Using those notations, $\mathbf{x}(t) = (y(t),\, z(t))$ obeys thus the Itô SDE

$$\mathrm{d}\mathbf{x}(t) = \mathbf{a}(x(t), t)\mathrm{d}t + \mathbf{B}(x(t), t)\mathrm{d}\mathbf{W}(t), \tag{4.71}$$

where $\mathbf{B}$ has to be solved from $2\mathbf{K} = \mathbf{B}\mathbf{B}^\intercal$. Since $2\mathbf{K}$ is positive semidefinite, a possible Cholesky decomposition is given by

$$\mathbf{B} = \begin{pmatrix} B_{yy} & 0 \\ B_{zy} & B_{zz} \end{pmatrix} = \begin{pmatrix} \sqrt{2K_{yy}} & 0 \\ B_* & \sqrt{2K_{zz} - B_*^2} \end{pmatrix}, \tag{4.72}$$

where

$$B_* = \begin{cases} 0 & \text{if } K_{yy} = 0, \\ \frac{2K_{yz}}{B_{yy}} & \text{otherwise.} \end{cases} \tag{4.73}$$

The Itô SDE (4.71) can be rewritten as

$$(\mathbf{I}) \begin{cases} \mathrm{d}y(t) = \left(v + \dfrac{\partial K_{yy}}{\partial y} + \dfrac{\partial K_{yz}}{\partial z}\right)\mathrm{d}t + B_{yy}\mathrm{d}W_1(t) & (4.74\text{a}) \\ \mathrm{d}z(t) = \left(w + \dfrac{\partial K_{zy}}{\partial y} + \dfrac{\partial K_{zz}}{\partial z}\right)\mathrm{d}t + B_{zy}\mathrm{d}W_1(t) + B_{zz}\mathrm{d}W_2(t) & (4.74\text{b}) \\ (y(0), z(0)) = (y_0, z_0), & (4.74\text{c}) \end{cases}$$

where $W_1(t)$ and $W_2(t)$ are independent Wiener processes. The derivatives of the elements of $\mathbf{K}$ appearing in equation (4.74) are called the *gradient drift terms.* If $\mathbf{K}$ is known analytically, those terms can be computed for the numerical implementation. If not, finite differences can be used. In the next of this work, we will encounter problems where $\mathbf{K}$ is discontinuous along segments in the domain, so that a part of the gradient drift terms is infinite at those points. Such an issue is addressed in [50] by neglecting the gradient drift terms all together, and in [51] by evaluating gradient drift terms via finite differences. Such methods are probably good enough for the simple problems considered in this work. However, we prefer the *backward-Itô* approach as this method applies to a wider range of problems with discontinuous diffusivities.

To derive the backward Itô SDE corresponding to the transport model, notice that equation (4.68) can also be rewritten as

$$\begin{aligned}\frac{\partial C}{\partial t} &= -\frac{\partial}{\partial y}(vC) - \frac{\partial}{\partial z}(wC) \\ &\qquad + \frac{1}{2}\left[\frac{\partial}{\partial y}\left(2K_{yy}\frac{\partial C}{\partial y} + 2K_{yz}\frac{\partial C}{\partial z}\right) + \frac{\partial}{\partial z}\left(2K_{zy}\frac{\partial C}{\partial y} + 2K_{zz}\frac{\partial C}{\partial z}\right)\right]. \qquad (4.75)\end{aligned}$$

This is precisely equation (4.54) with $\mathbf{x} = (y, z)$, $p = C$, $\mathbf{a} = (v, w)$ and $\mathbf{D} = 2\mathbf{K}$. Hence, we can use the matrix $\mathbf{B}$ computed in (4.72). Then, $\mathbf{x}(t) = (y(t), z(t))$ also obeys the backward Itô SDE

$$(\mathbf{bI})\begin{cases} \mathrm{d}y(t) = v\mathrm{d}t + B_{yy}\mathrm{d}W_1(t) & (4.76a) \\ \mathrm{d}z(t) = w\mathrm{d}t + B_{zy}\mathrm{d}W_1(t) + B_{zz}\mathrm{d}W_2(t) & (4.76b) \\ (y(0), z(0)) = (y_0, z_0). & (4.76c) \end{cases}$$

Interestingly, there is no gradient drift term in (4.76), i.e. no derivative of the diffusivities. In the context of a problem with discontinuous diffusivities, the backward Itô interpretation is thus particularly interesting: see [44] and [29] for more complete discussions about the use of the backward Euler method on problems with discontinuous diffusivities.

4.3 The code

The preceding sections cover all the material needed to implement a Lagrangian code that solves a two-dimensional advection-diffusion problem. For the need of this work, a C++ code has been implemented. The choice of C++ is motivated by the fact that it is *fast*, and that it comes together with a wide range of *open source* supporting tools. Another reason is that C++ is an *object-oriented language*, and it is widely held that writing in an object-oriented style leads to programs which are easier to understand, to extend, to maintain and to refactor [52].

The code deals with the two-dimensional transport equation

$$\frac{\partial C}{\partial t} = \nabla \cdot (-\mathbf{u}C + \mathbf{K}\nabla C) \qquad (4.77)$$

on rectangular domains with no-through boundary conditions. It allows to simulate trajectories, to compute the concentration and to build the transition probability matrix for a given partitioning of the domain. The trajectories are simulated using the backward Euler method, applied on the system of backward Itô SDE's (4.76). In this work, only the box-counting method is used for the estimation of the concentration (and thus also for the computation of the transition probability matrix) but the density kernel estimation method has also been implemented for the sake of completeness.

In order to use the code on a particular problem meeting the above specifications, a class that defines the problem must be implemented. That class must inherit from the abstract base class `AbstractAdvDiffProblem` (see listing 1), and must at least implement the two pure virtual functions of the abstract base class:

```
class AbstractAdvDiffProblem
{
    protected:
        double mH0, mH1; // boundaries of the domain in the z-direction : H0 <= z <= H1
        double mL0, mL1; // boundaries of the domain in the y-direction : L0 <= y <= L1

    public:
        AbstractAdvDiffProblem(double H0, double H1, double L0, double L1);
        virtual ~AbstractAdvDiffProblem(){};
        double getH0() const;
        double getH1() const;
        double getL0() const;
        double getL1() const;
        virtual SymMatrix getK(double y, double z) const=0; // diffusivity tensor
        virtual LowerTriMatrix getB(double y, double z) const; // 2K = BB'
        virtual Vec2 getU(double y, double z) const=0; // velocity vector
        virtual void printInfo(std::ofstream& f) const;
};
```

Listing 1 – The abstract base class `AbstractAdvDiffProblem`.

- `SymMatrix getK(double y, double z)`: returns the diffusivity tensor $\mathbf{K}$ evaluated at (y, z). The return value is of type `SymMatrix`, a structure intended to store a 2×2 symmetric matrix with only three elements stored in memory. Instantiating an object `A` of type `SymMatrix` is pretty simple: `SymMatrix A(a,b,c)` creates the matrix
$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix},$$
where `a`, `b` and `c` are of type `double`. The elements of `A` are then accessed via the syntax `A(i,j)` which uses one-based indexing. Hence, `A(1,1) = a`, `A(1,2) = A(2,1) = b` and `A(2,2) = c`. The same syntax can be used to modify the elements of `A`.
- `Vec2 getU(double y, double z)`: returns the velocity vector $\mathbf{u}$ evaluated at (y, z). The return value is of type `Vec2`, a structure that stores two elements of type `double`. The syntax `Vec2 v(a,b)` is used to create the two-dimensional vector $v = (a, b)$. The elements of `v` are accessed via the syntax `v(i)` which also uses one-based indexing: `v(1) = a` and `v(2) = b`.

By default, the code computes the matrix $\mathbf{B}$ using (4.72) and (4.73). This is done by the function `LowerTriMatrix GetB(double y, double z)`. In some cases, it can be interesting to overload that definition of `GetB`, which is possible since this function is virtual. The return value must be of type `LowerTriMatrix`, which is a structure similar to `SymMatrix` but is intended to store lower triangular 2×2 matrices instead of symmetric 2×2 matrices.

Notice that the code as such implements the dimensional form of the transport model. However, it can be used to run simulations on the adimensional form of the transport model. To this end, it suffice to define the functions `getK` and `getU` accordingly: `getK` shall return the inverse of the Peclet matrix, and `getU` shall return the adimensional velocity vector.

Once a class describing the problem is properly defined, three methods can be used to compute either the trajectories, the normalized concentration or the transition probability matrix. We call those methods the *compute methods*. Their signatures are given in listing 2. Since those functions are well documented in the code, we invite the interested reader to refer to the code for further explanations about those functions.

```
void ComputeTrajectories(const AbstractAdvDiffProblem& prob, std::string model,
                         double dt, double T, int Nloc, double yStart, double zStart);
void ComputeConcentration(const AbstractAdvDiffProblem &prob, std::string model,
                          double dt, double T, std::string estimator, int Nloc,
                          double yStart, double zStart, int nboxy, int nboxz);
void ComputeTransitionProbabilities(const AbstractAdvDiffProblem& prob,
                                    std::string model, int nboxy, int nboxz, int nyloc,
                                    int nzloc, double dt, double Times[], int nTimes,
                                    bool binary, std::string estimator = "box");
```

Listing 2 – Signatures of the *compute methods*.

Test cases with analytical solutions have been built to assess the validity of the implementation. They are presented in appendix B and the numerical solution is compared to the analytical solution, producing satisfying results.

Chapter 5

From clusters to compartments: the method

In this chapter, we formalize the method used to go from the numerical implementation of the particle trajectories to the clustering allowing to delineate the compartments in a box model. In other words, it is explained how to use a clustering algorithm on an advection-diffusion problem, and then how to choose which communities found by the stability method to use as compartments in a box model.

The underpinning idea of the method is that communities found on the dynamic of the flow could be relevant compartments for the box model. The idea makes sense although it is far from being obvious that the communities are indeed the compartments that we seek. In a first instance, we should only check that, at least in some cases, community detection leads to relevant partitions. In the next chapter, a test problem is build for which we know in advances what compartments to expect, and the method is tested on that problem.

Remark In the context of interpreting the communities as compartments for a box model, we should require that the communities have vertical and horizontal boundaries. For example, when dealing with marine problems, the goal of a box model is to provide a simple and intuitive description of the problem. Complex shaped compartments are neither simple nor intuitive for marine models, hence the requirement.

5.1 Description of the method

In order to apply a clustering algorithm on a physical advection-diffusion problem, we have to define how the problem can be considered as a graph. For the next, we consider a two-dimensional problem in the coordinate system (y, z). Let us partition the domain into $n_{cell,\,y} \times n_{cell,\,z}$ grid cells, and denote $N_{cell} = n_{cell,\,y} n_{cell,\,z}$ the total number of grid cells. Figure 5.1 represents an example of such a domain decomposition of the rectangular domain $[0,\, L] \times [0,\, H]$ with $n_{cell,\,y} = 15$ and $n_{cell,\,z} = 10$. For any time T, the corresponding directed graph is build as follows: each node represents a grid cell, and the weight of the edge between nodes i and j is the probability $m_{i,j}(T)$ that a particle ends up in grid cell j after a time T if it was initially in grid cell i. If $m_{i,j}(T) = 0$, one can equivalently consider that there is no edge between nodes i and j. We restrict ourselves to stationary velocity field and diffusivity tensor, hence

$m_{i,j}(T)$ 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 $m_{i,j}(T)$: $[\mathbf{M}(T)]_{i,j} = m_{i,j}(T)$. It is thus precisely the transition probability matrix introduced in chapter 1 which is needed to apply the stability clustering method. Since we consider passive tracer's particles in a domain with no-through boundary condition, we have that for any time T, $\mathbf{M}(T)\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is the N_{cell}-dimensional unit column vector. The latter has a straightforward physical interpretation: every particle remains in the domain and particles are neither created nor destructed.

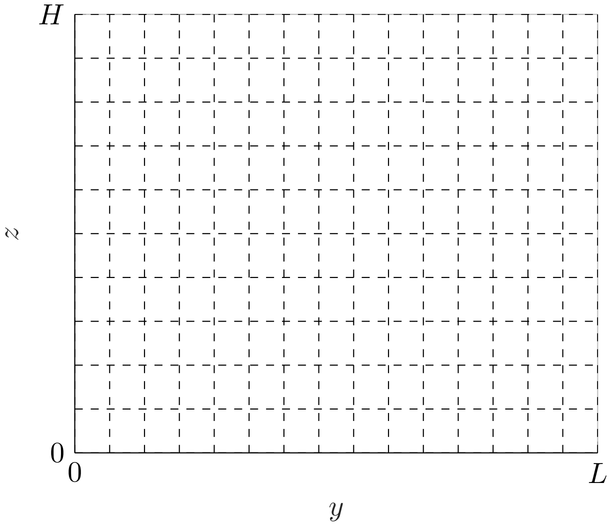


Figure 5.1 – Illustration of the decomposition of a rectangular domain into grid cells with $n_{cell,\,y} = 15$ and $n_{cell,\,z} = 10$.

To estimate the probabilities $m_{i,j}(T)$, a Lagrangian simulation is run for a time T with each grid cell containing initially P_0 uniformly distributed particles. Let $P_{i\to j}(T)$ denote the number of particles in grid cell j at time T, which were initially in grid cell i. Using that notation, $m_{i,j}(T)$ is then numerically estimated as

$$m_{i,j}(T) = \frac{P_{i\to j}(T)}{P_0}, \tag{5.1}$$

i.e. the number of particles having started in grid cell i and ending up in grid cell j after time T, divided by P_0. This is exactly the *box counting* method introduced in section 4.1.8 for the computation of the concentration, but instead of dividing the tracer's mass in a grid cell by the volume of that grid cell, we divide it by P_0. This yields an adimensional quantity that can be interpreted as a transition probability. Note that the box counting method is more adapted than a density kernel estimation for this problem because the volume average over grid cells is precisely what we want.

5.2 Dealing with the time scales

An important feature of the stability method for detecting community structures is that it is *dynamic*: community structures are revealed as a function of the Markov time t_M. For the problems that we consider, this Markov time is intrinsically linked to the physical time: for a given time T, suppose that the stability method is applied on the adjacency matrix $\mathbf{M}(T)$. In the discrete framework, a particle jumps from one node to another at every integer Markov time, and in the continuous framework, the expected time between jumps is $\langle\langle \mathbf{q} \rangle / \mathbf{q}\rangle = 1$ (cfr. page 8). Hence, a Markov time step of 1 corresponds to a physical time step of T.

From the above discussion, two possibilities arise for dealing with the time scales: either we compute the adjacency matrix at one unique time T and then compute the stability on the desired range of Markov times, or we compute the adjacency matrix at different times and then compute the stability on each adjacency matrix but for the Markov time $t_M = 1$ only. The advantage of the first method is that we do not have to fix *a priori* the time scales at which we compute clusterings: such times scales arise naturally as plateaux in the community curve, with a low corresponding variation of information. Hence the relevant time scales are deduced from the stability curve as being the ones at which robust clusterings arise. At the contrary, the second method imposes that we choose the time scales beforehand; doing so, we lose one of the most appealing features of the stability approach. Furthermore, the first method is computationally lest costly. Even for a relatively coarse partitioning of the domain, say of about 300 grid cells, if we release 10 000 particles in each grid cell there is a total of $3{\times}10^6$ trajectories to simulate. For long T, the simulation time might become restrictive, especially in the case where one does not have access to supercomputers to run the code in parallel. The same situation leads to a 300 nodes network, which is a relatively small network size that can easily be handled by the stability software. The first method has however one important drawback: the errors in the adjacency matrix are spread and even amplified across the Markov times. If those errors become too important, the community structures found at large Markov times might become irrelevant. In other words, simulating the transition probability matrix for a time T and taking the nth power of that matrix is not necessarily equivalent to simulating the transition probability matrix for a time nT. The ideal methodology is thus probably to use the first method to detect the interesting time scales and compute the corresponding community structures, and then to check that we get similar community structures at the same time scales using the second method.

5.3 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 chapter 1. The `stability` function is simply called as follows :

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

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',tau` allows to specify the value of the teleportation probability τ to `tau`, the default value being 0.15. In most cases, we will choose `tau` $= 0$. This choice is motivated by the fact that if our approximation of the transition probability matrix is close enough the the exact one, then if the diffusivities are everywhere strictly positive the graph is ergodic (notice that there can be no dangling node whatever the precision of our approximation). Further in this work, one example where the graph is not ergodic will be encountered. In that case, the value of `tau` must be chosen strictly positive in order to ensure ergodicity.

A small value is then preferred, in order to minimize the impact of random teleportations on the dynamics of the graph. We will typically choose `tau` $= 10^{-3}$ in such a case.

Unfortunately, the software does not handle discrete-time stability. Instead, it allows to choose which type of laplacian should be used to calculate the (continuous-time) 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 P_0 released in each grid cell. Hence, $k_i = P_0$ for every node i and $\langle \mathbf{k} \rangle = P_0$, 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 $N_{cell} \times n$ 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)`.

Chapter 6

The "overturner" circulation model

A velocity field is proposed here that will be used further to build problems on which the method will be applied. That circulation model was initially proposed by *E. Deleersnijder* [53] as an idealized, two-dimensional representation of the meridian circulation in the Atlantic ocean. It has been studied extensively by *C. Timmermans* in her master's thesis [54]. The justifications of the model can be found in the latter. In this chapter, we only present the circulation model without justification.

6.1 An idealized velocity field

Let us consider a rectangular domain in the (y, z)-coordinate system. The domain Ω is delimited by

$$0 \leq y \leq L, \quad 0 \leq z \leq H, \tag{6.1}$$

where L and H are positive constants. The stationary velocity field is denoted $\mathbf{u}(y, z) = (v(y, z),\ w(y, z))$. Under the Boussinesq approximation, the continuity equation reads

$$\nabla \cdot \mathbf{u} = 0, \tag{6.2}$$

For our particular problem, this amounts to

$$\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{6.3}$$

No-through boundary conditions are imposed at the boundaries of the domain, which implies that $\mathbf{u}(y, z) \cdot \hat{\mathbf{n}} = 0$ everywhere on $\partial\Omega$ (where $\hat{\mathbf{n}}$ is the outwards unit normal at the boundary), or equivalently :

$$v(0, z) = 0, \quad v(L, z) = 0, \quad w(y, 0) = 0 \quad \text{and} \quad w(y, H) = 0. \tag{6.4}$$

In order to build a velocity field that satisfies the continuity equation (6.2), a *streamfunction* approach is used. The whole approach is explained in appendix C along with some important properties of the streamfunction in two dimensions that will be useful in the next. Let ψ denote the streamfunction. The velocity field satisfies the relation

$$\mathbf{u} = -\nabla \times \psi. \tag{6.5}$$

In particular, ψ must be such that the boundary conditions (6.4) are satisfied. Those conditions state that $\mathbf{u}$ must be tangent to the boundary everywhere on $\partial\Omega$, which precisely amounts to require that ψ is constant on $\partial\Omega$ (see property C.2). Without loss of generality, we can choose this constant to be zero. Hence, we require that

$$\psi(0,z)=0,\ \psi(L,z)=0,\ \psi(y,0)=0 \text{ and } \psi(y,H)=0, \text{ for all } (y,z)\in\Omega. \tag{6.6}$$

In order to build an acceptable idealization of the meridian circulation in the Atlantic ocean, *Deleersnijder* proposes in his working paper [53] to suppose that the meridian streamfunction has a unique extremum Ψ, which is a maximum, and that it reaches that maximum at the point of coordinates (y_0, z_0), located near the surface and near the North boundary of the domain. It is important to recall that the second partial derivatives of ψ must exist and be continuous for the above relations to hold.

Let $\xi_0 \in \mathbb{R}_0^+$, and let $\phi(\xi, \xi_0)$ be defined as

$$\phi(\xi,\xi_0)=\frac{\xi(2\xi_0-\xi)}{\xi_0^2}, \tag{6.7}$$

The derivative $\phi'(\xi,\xi_0)$ of ϕ with respect to ξ is

$$\phi'(\xi,\xi_0)=\frac{2(\xi_0-\xi)}{\xi_0^2}. \tag{6.8}$$

An expression of the meridian streamfunction that satisfies the above constraints is then

$$\psi(y,z)=\Psi\begin{cases}\phi(y,y_0)\phi(z,z_0) & \text{if } 0\le y<y_0, \quad 0\le z<z_0,\\ \phi(y,y_0)\phi(H-z,H-z_0) & \text{if } 0\le y<y_0, \quad z_0<z\le H,\\ \phi(L-y,L-y_0)\phi(H-z,H-z_0) & \text{if } y_0<y\le L, \quad z_0<z\le H,\\ \phi(L-y,L-y_0)\phi(z,z_0) & \text{if } y_0<y\le L, \quad 0\le z<z_0,\\ \phi(z,z_0) & \text{if } y=y_0, \quad 0\le z<z_0,\\ \phi(H-z,H-z_0) & \text{if } y=y_0, \quad z_0<z\le H,\\ \phi(y,y_0) & \text{if } 0\le y<y_0, \quad z=z_0,\\ \phi(L-y,L-y_0) & \text{if } y_0<y\le L, \quad z=z_0,\\ 1 & \text{if } y=y_0, \quad z=z_0.\end{cases} \tag{6.9}$$

Figure 6.1 shows the isolines of the adimensional streamfunction ψ/Ψ for a given choice of the parameters. The point is that this circulation describes revolutions around the point (y_0, z_0). With the choice $\mathbf{u}=-\nabla\times\psi$, those revolutions are clockwise.

The meridian and vertical components of the velocity are then expressed as

$$v(y,z)=\Psi\begin{cases}-\phi(y,y_0)\phi'(z,z_0) & \text{if } 0\le y<y_0, \quad 0\le z<z_0,\\ \phi(y,y_0)\phi'(H-z,H-z_0) & \text{if } 0\le y<y_0, \quad z_0<z\le H,\\ \phi(L-y,L-y_0)\phi'(H-z,H-z_0) & \text{if } y_0<y\le L, \quad z_0<z\le H,\\ -\phi(L-y,L-y_0)\phi'(z,z_0) & \text{if } y_0<y\le L, \quad 0\le z<z_0,\\ -\phi'(z,z_0) & \text{if } y=y_0, \quad 0\le z<z_0,\\ \phi'(H-z,H-z_0) & \text{if } y=y_0, \quad z_0<z\le H,\\ 0 & \text{if } 0\le y\le L, \quad z=z_0.\end{cases} \tag{6.10}$$

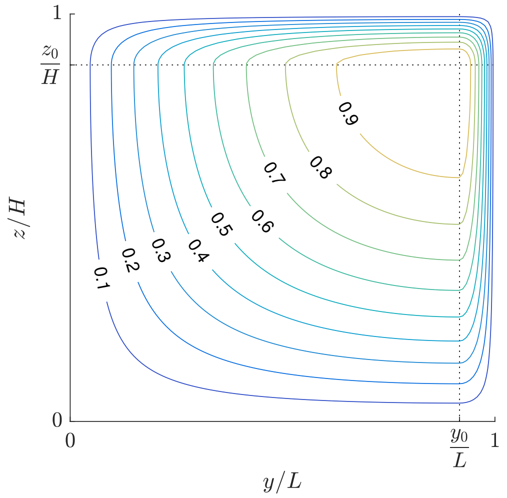


Figure 6.1 – Some isolines of the adimensional meridian streamfunction $\psi(y,z)/\Psi$, which are also streamlines of the flow.

and

$$w(y,z) = \Psi \begin{cases} \phi'(y,y_0)\phi(z,z_0) & \text{if} \quad 0 \le y < y_0, \quad 0 \le z < z_0, \\ \phi'(y,y_0)\phi(H-z,H-z_0) & \text{if} \quad 0 \le y < y_0, \quad z_0 < z \le H, \\ -\phi'(L-y,L-y_0)\phi(H-z,H-z_0) & \text{if} \quad y_0 < y \le L, \quad z_0 < z \le H, \\ -\phi'(L-y,L-y_0)\phi(z,z_0) & \text{if} \quad y_0 < y \le L, \quad 0 \le z < z_0, \\ \phi'(y,y_0) & \text{if} \quad 0 \le y < y_0, \quad 0 \le z = z_0, \\ -\phi'(L-y,L-y_0) & \text{if} \quad y_0 < y \le L, \quad z = z_0, \\ 0 & \text{if} \quad y = y_0. \end{cases} \tag{6.11}$$

Chapter 7

Application: the bi-overturner problem

This section can be considered as a kind of *sanity check* of the method: we define a class of problems, the *bi-overturner* problems, for which the box decomposition is intuitively obvious and we check that a community detection algorithm applied on the problem allows to find back that box-decomposition. Then, compartment models are build and their validity is discussed.

7.1 The bi-overturner class of problems

The bi-overturner problems basically consist of two *overturner* circulations models side-by-side, hence the name. However, although the overturner circulation was initially developed as an idealization of the meridional circulation in the Atlantic ocean, bi-overturner problems do not model any "real-life" problem. Therefore, the values of the different physical parameters are given without justification, although most of the quantities are inspired from the values proposed in [54]. The bi-overturner problems are really used as a mathematical tool to test the method, and using the overturner circulation ensures that the velocity field that we consider satisfies the continuity equation and no-through boundary condition everywhere on $\partial\Omega$. The domain that we consider is $\Omega = [-L,\, L] \times [0,\, H]$. Let $\Omega^- = [-L,\, 0[\times[0,\, H]$ and $\Omega^+ =]0,\, L] \times [0,\, H]$. If $\psi(y, z; y_0, z_0)$ denote the streamfunction defined in (6.9) with parameters $y_0 \in]0,\, L[$ and $z_0 \in]0,\, H[$, then the streamfunction φ of the bi-overturner problems is defined as

$$\varphi(y,z) = \begin{cases} \psi(L+y, z; y_0, z_0) & \text{if} \quad (y,z) \in \Omega^-, \\ 0 & \text{if} \quad (y,z) \in \{(0,z) \mid z \in [0,\, H]\}, \\ -\psi(y, z; L-y_0, z_0) & \text{if} \quad (y,z) \in \Omega^+, \end{cases} \tag{7.1}$$

The streamfunction φ has two extrema of equal strengths: a maximum at (y_0^-, z_0) with $y_0^- := -L+y_0 = -y_0^-$ and a minimum at (y_0^+, z_0) with $y_0^+ := L-y_0$. The overturner-like circulation is clockwise in Ω^- and counterclockwise in Ω^+. The horizontal and vertical velocities v and w are given by

$$v = -\frac{\partial\varphi}{\partial z}, \quad w = \frac{\partial\varphi}{\partial y}. \tag{7.2}$$

Isolines are shown in figures 7.1 and 7.2 respectively. The key feature is that $v(0,z) = 0$, namely the horizontal velocity is zero on the whole segment $y = 0$. Hence, if there is no horizontal diffusion, a passive tracer's particle starting in Ω^- can never reach Ω^+ and conversely. In that case, we can imagine that there is a vertical wall implying no-through boundary conditions at $y = 0$ and the graph is not ergodic. But if the horizontal diffusivity is nonzero in some area near $y = 0$, then exchange of particles between Ω^- and Ω^+ can happen in that area. Now we suppose that the diffusivity tensor $\mathbf{K}$ is diagonal:

$$\mathbf{K}(y,z) = \begin{pmatrix} K_{yy}(y,z) & 0 \\ 0 & K_{zz} \end{pmatrix}. \tag{7.3}$$

We assume that the vertical diffusivity K_{zz} is constant and equal to 10^{-3} [m^2/s]. Now we introduce the parameter $\alpha \in [0,1]$ and define $z^* = \alpha H$. We choose an horizontal diffusivity K_{yy} of the form

$$K_{yy}(y,z) = \begin{cases} 10^4 \text{ [m}^2\text{/s]} & \text{if} \quad y_0^- \leq y \leq y_0^+ \quad \text{and} \quad z^* \leq z \leq H, \\ 10^3 \text{ [m}^2\text{/s]} & \text{if} \quad -L \leq y < y_0^- \quad \text{or} \quad y_0^+ < y \leq L, \\ 0 \text{ [m}^2\text{/s]} & \text{otherwise.} \end{cases} \tag{7.4}$$

Now, exchange between Ω^- and Ω^+ is possible but only above z^*.[1] Hence, we can imagine that there is a vertical, no-through wall of height z^* at $y = 0$. The situation is depicted on figure 7.3. For the next, we call *exchange zone* the area where $K_{zz} = 10^4$ [m^2/s] (dark gray zone in figure 7.3). Making α vary between 0 and 1 defines a class of bi-overturner problem where the vertical wall's height z^* at $y = 0$ vary between 0 and H. Two examples of trajectories with the same initial condition are shown for $\alpha = 0.75$ in figures 7.4 and 7.5.

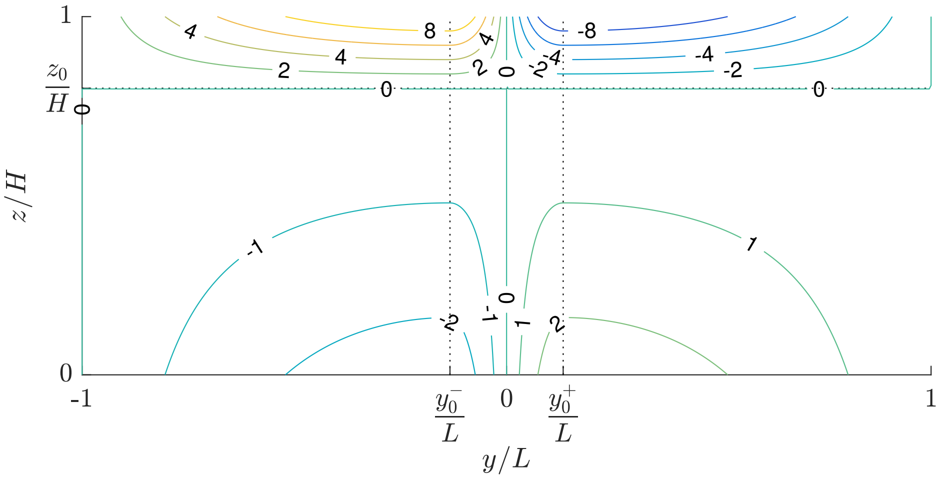


Figure 7.1 – Isolines of the horizontal velocity v for bi-overturner problems.

When $\alpha = 1$, the obvious compartmental model is made of the two compartments Ω^- and Ω^+ which do not communicate with each other. Suppose that different amounts of passive tracer are released into Ω^- and Ω^+ at a given time; the concentration in each compartment tends to become uniform in time due to diffusion, but the concentration in Ω^- depends only on the initial quantity of tracer released in Ω^- and similarly for the concentration in Ω^+. At the contrary, when $\alpha = 0$ the concentration tends to become uniform over the whole domain when time goes to infinity. Hence, we can expect three types of partitioning to be dominant at different time scales: a three-communities

[1] From a numerical, random-walk, point of view, we should also ensure that $y_0^+ = |y_0^-|$ is sufficiently large. If not, it would be numerically possible for particles lying below z^* and before y_0^- (resp. after y_0^+) to jump from Ω^- (resp. Ω^+) to Ω^+ (resp. Ω^-).

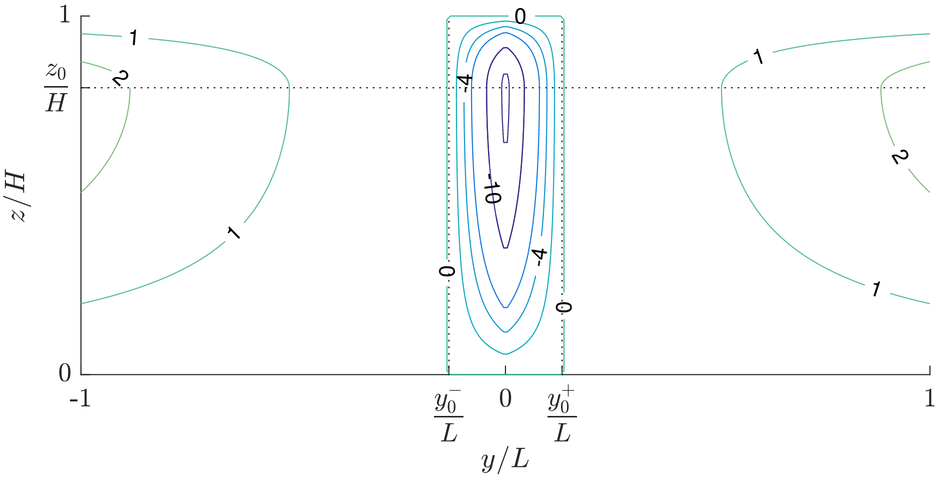


Figure 7.2 – Isolines of the vertical velocity w for bi-overturner problems.

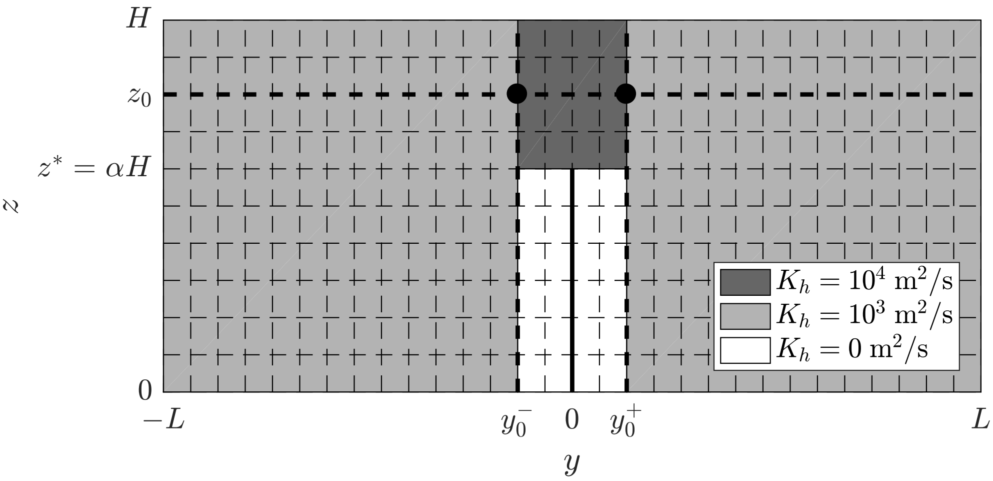


Figure 7.3 – Illustration of the decomposition of the domain into grid cells corresponding to the nodes of the directed graph. The values of K_{yy} are also shown for $\alpha = 0.6$, and the fictitious wall is represented by the black continuous line.

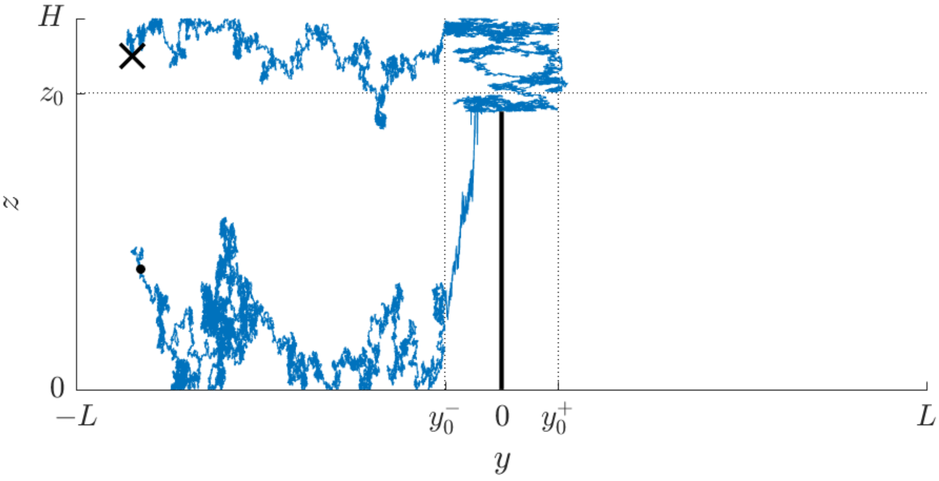


Figure 7.4 – Example of a particle trajectory in the bi-overturner model with $\alpha = 0.75$. The black cross represents the initial position whereas the black dot shows the final position. The simulation time is 200 years. Here the particle enters the exchange zone but finally stays in Ω^-.

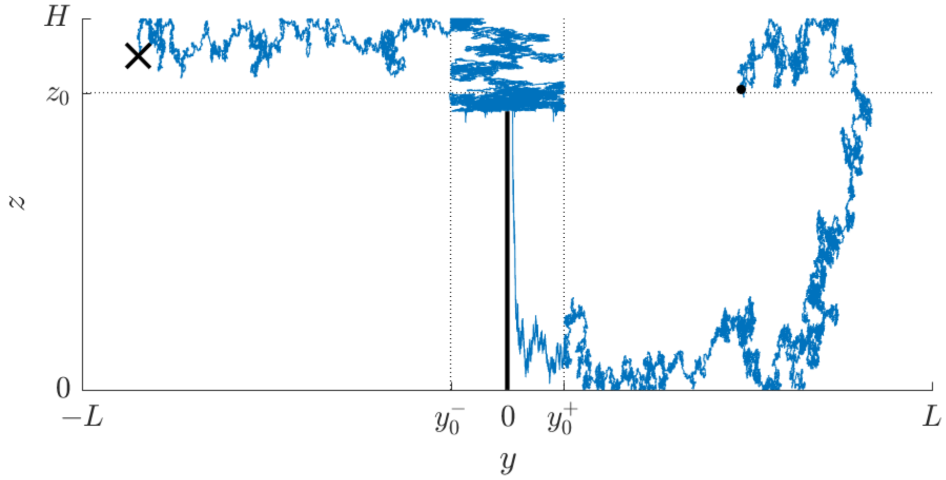


Figure 7.5 – Example of a particle trajectory in the bi-overturner model with $\alpha = 0.75$. The black cross represents the initial position whereas the black dot shows the final position. The simulation time is 200 years. Here the particle enters the exchange zone and finally goes in Ω^+.

partitioning with left and right compartments and an intermediate zone in between; a two-communities partitioning corresponding to Ω^- and Ω^+; and finally a trivial partitioning with one single community for very long time scales. For intermediate values of α, we expect a behavior similar to the case $\alpha = 0$ when α is close to 0. When α is close to 1 but still different from 1, exchange can still happen between Ω^- and Ω^+. However, in terms of communities, we expect the two-community partitioning to be dominant for long time scales.

7.2 Application of the stability method

The partitioning results are presented here for $\alpha = 1$, $\alpha = 0.75$, $\alpha = 0.5$, $\alpha = 0.25$ and $\alpha = 0$. For every value of α, the transition probability matrix is computed at $T = 1$ year on a discretization like the one presented in figure 7.3, namely with $n_{cell,\,y} = 30$ and $n_{cell,\,z} = 10$. $P_0 = 10\,000$ particles are initially released in every grid cell. The stability software is run using a vector of Markov times taking values between 10 and 10^3. Since we have computed the transition probability matrix for $T = 1$ year, one unit of Markov time correspond here to one year. Notice that in the case where $\alpha = 1$, the graph is not ergodic (it is composed of two ergodic classes) and a random teleportation probability `tau` $= 10^{-3}$ is used when running the stability software. When $\alpha < 1$, `tau` $= 0$ is used. The stability, number of communities and variation of information curves are shown in figures 7.6, 7.7, 7.8, 7.9 and 7.10 for the different values of α. Most robust communities correspond to plateaux in the community curve with a corresponding low variation of information. Whatever the value of α, the number of communities goes to 2 after a given time and the corresponding variation of information is almost zero. This happens around 50 years when $\alpha > 0.25$, and a bit later for $\alpha = 0.25$ and $\alpha = 0$. In every case, the two-communities partitioning corresponds as expected to Ω^- and Ω^+. It is shown in figure 7.12 for the case $\alpha = 0.75$. Exactly the same two-communities clusterings are found in the cases $\alpha = 0.25$ and $\alpha = 1$. In those cases, the boundary is perfectly straight: this corresponds to the intuition and it is conform to the remark made on page 45. When $\alpha = 0$ or $\alpha = 0.5$, the boundary is not exactly a straight line but almost. The situation is depicted in figure 7.11 for $\alpha = 0.5$. However, we have to remember that neither the transition probability matrix nor the stability partitioning is solved exactly. Hence, we can consider that the irregularity in the boundary of the communities is due to those numerical artifacts: if a box model has to be build from the partitioning shown in figure 7.11, the compartments should of course be chosen with a vertical boundary. This illustrates the fact that when using a community-detection algorithm to build compartments for a box model, the communities should not be blindly interpreted as being the relevant compartments. In particular, if the boundaries of the communities are almost but not exactly vertical and horizontal, one should consider straight boundaries for the compartments. Community detection should thus be considered as a guide towards choosing relevant compartments, rather than as a method providing the exact perfect compartments.

When $\alpha = 0.75$, a small three-communities plateau starts appearing around 40 years, just before the two-communities plateau. This plateau grows as α decreases. The corresponding clusterings are shown in figure 7.13 for $\alpha = 0.25$ and in figure 7.14 for $\alpha = 0$. A central intermediate community appears, as expected.

In figure 7.10 (stability, number of communities and VI for the case $\alpha = 0$), peaks corresponding to oscillations between two and three communities are observed around 300 and 400 years. As stated in chapter 1, the number of communities should decrease with time, and those oscillations are thus due to the fact that the stability partitioning is only solved approximately. However, this shows that the two- and three-communities clusterings have similar stabilities at those times. Finally, remember that we expected to find a single-community partitioning for very long Markov times in the case $\alpha = 0$, which does not appear here. Such a clustering would probably appear if we run the stability software for longer Markov times.

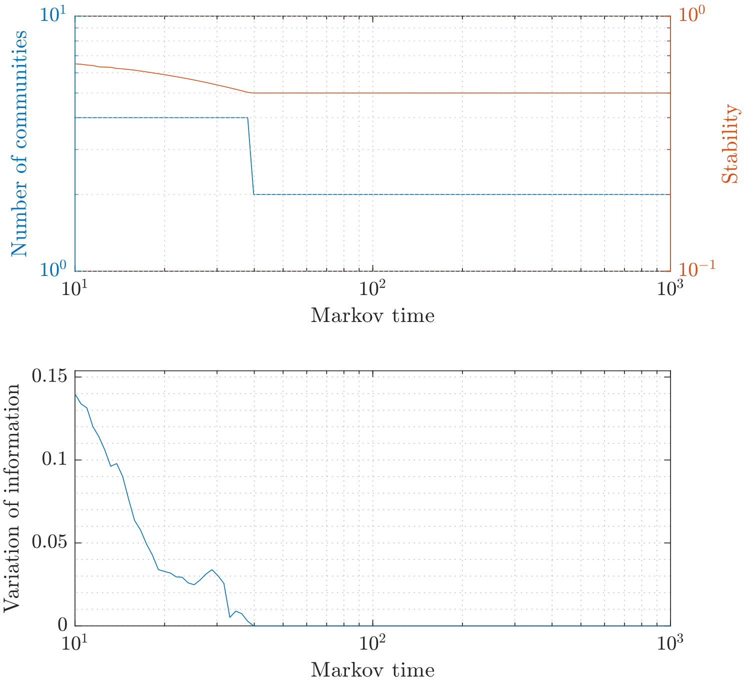


Figure 7.6 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 1$.

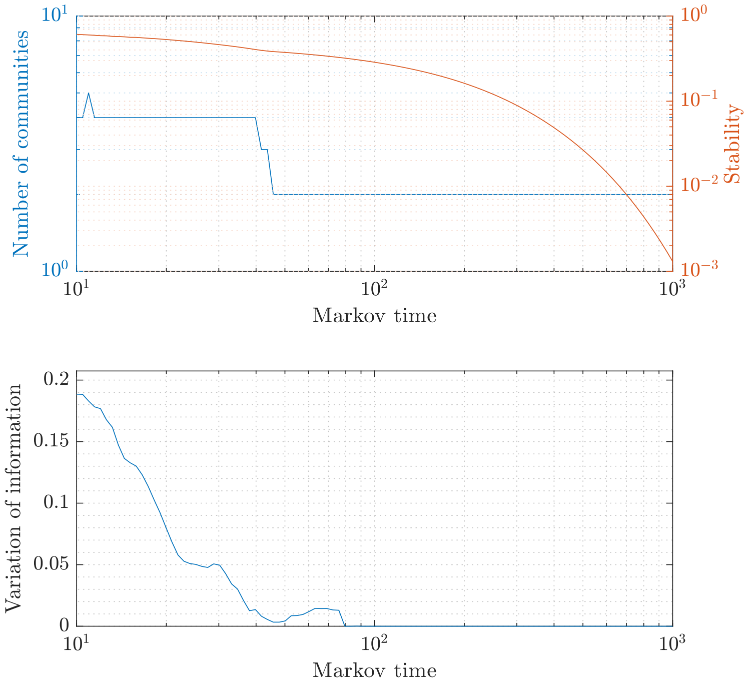


Figure 7.7 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 0.75$.

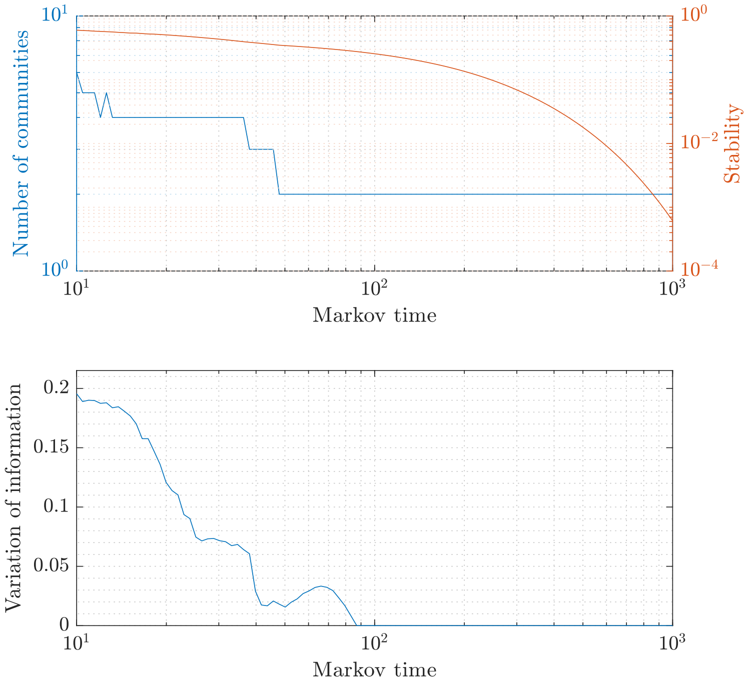


Figure 7.8 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 0.5$.

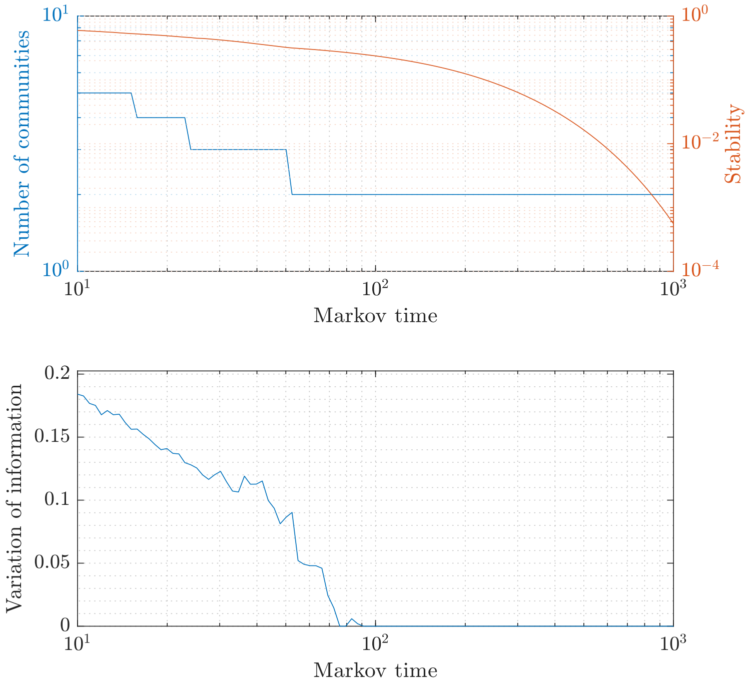


Figure 7.9 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 0.25$.

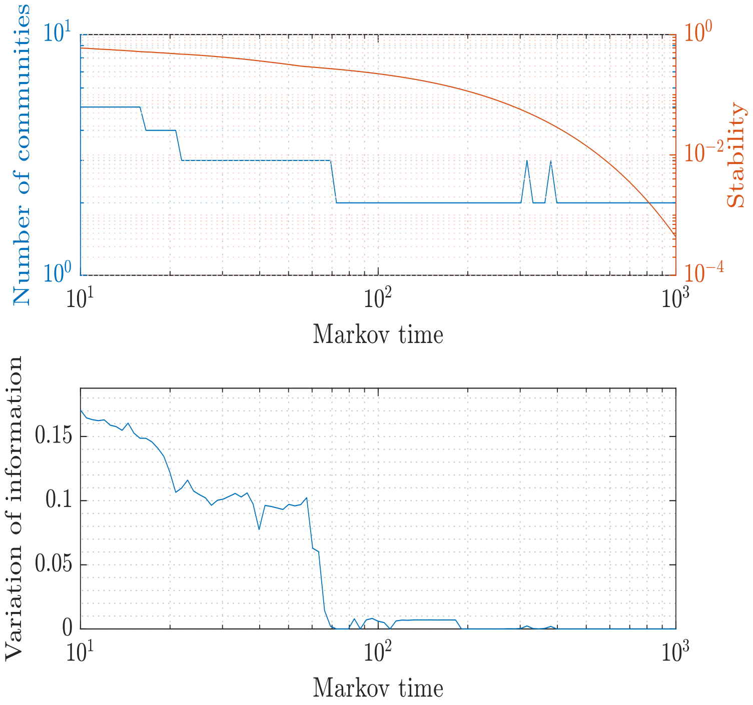


Figure 7.10 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 0$.

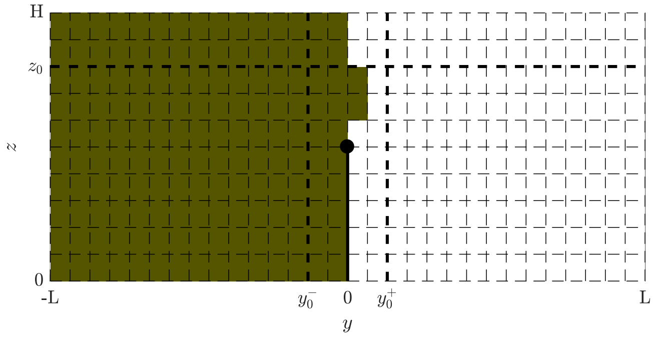


Figure 7.11 – Illustration of the two-communities partitioning for $\alpha = 0.5$.

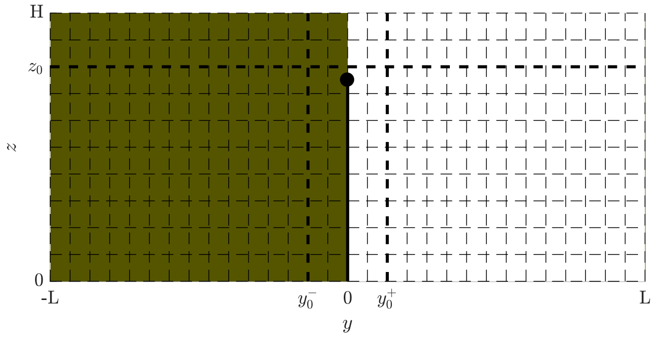


Figure 7.12 – Illustration of the two-communities partitioning for $\alpha = 0.75$. Exactly the same two-communities clustering is found for $\alpha = 0.25$, $\alpha = 0.9$ and $\alpha = 1$, but in this figure the wall is represented to have height $z^* = 0.75H$.

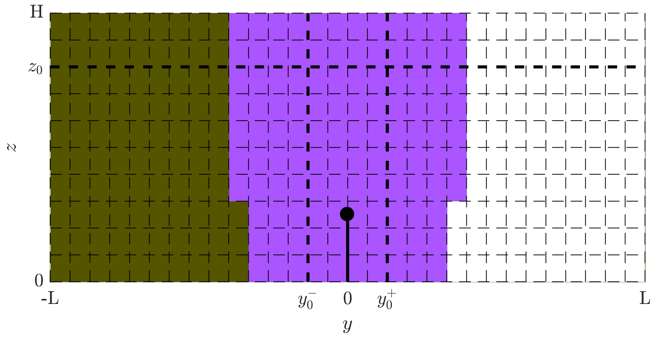


Figure 7.13 – Illustration of the three-communities partitioning for $\alpha = 0.25$.

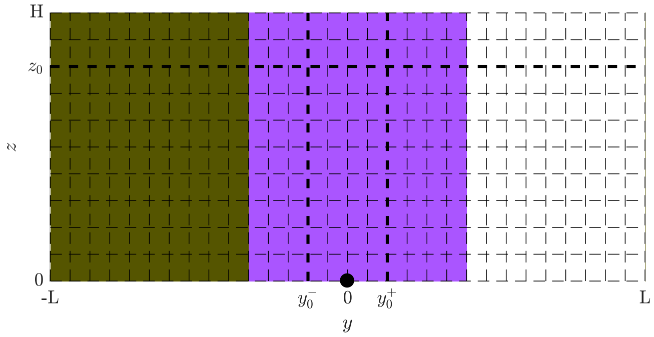


Figure 7.14 – Illustration of the three-communities partitioning for $\alpha = 0$.

7.3 Building a compartment model

Let us now focus on the case $\alpha = 0.75$. The community detection method applied in the previous section suggests a two compartments model with $\Omega_1 = \Omega^-$ and $\Omega_2 = \Omega^+$, see figure 7.12. This clustering appears around 50 years. In order to ensure that this clustering is indeed relevant at that time scale, we apply the second approach proposed in section 5.2 for dealing with the time scales: the stability is computed on the transition probability matrix $\mathbf{M}(50)$, numerically approximated under the same conditions as $\mathbf{M}(1)$ in section 7.2. The stability method is applied for Markov times in the interval [1, 10]: this corresponds to physical times between 50 and 500 years. The stability, community and variation of information curves are shown in figure 7.15. A two-communities partitioning is indeed found in the range of Markov times [1.028, 4.074]. This corresponds to the range of physical times [51.4, 203.7] years. The two communities are exactly the same as the ones obtained with $\mathbf{M}(1)$ and shown in figure 7.12, indicating a robust community structure.

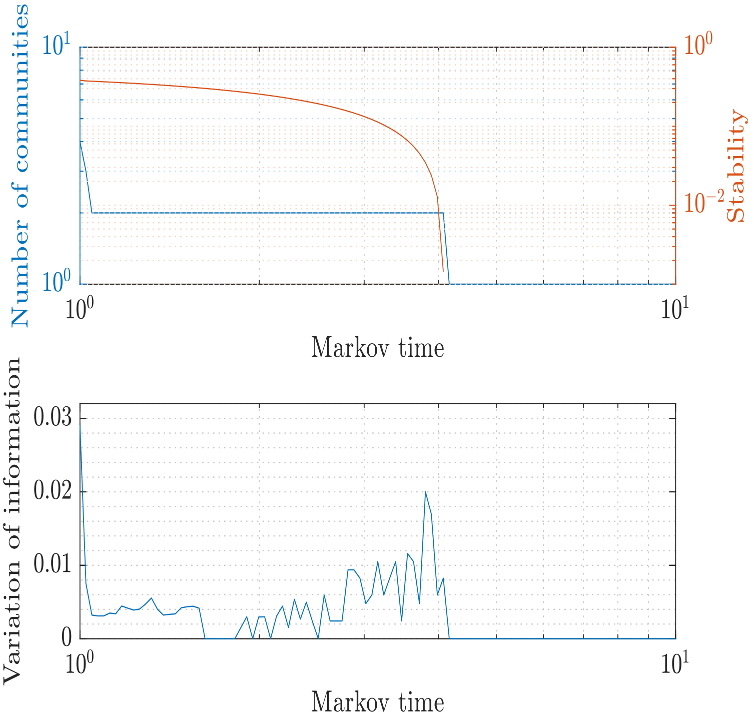


Figure 7.15 – Stability, number of communities and variation of information as a function of the Markov time for $\alpha = 0.75$. Those results are obtained from the transition probability matrix evaluated a $T = 50$ years.

Now we are ready to build a two-compartment model based on the clustering from figure 7.12. First, in section 7.3.1, the limitations of such a two-compartment model are discussed *a priori*. Then, two approaches are proposed in order to build a compartment model: in section 7.3.2, a continuous-time compartment model is build which depends on one parameter, and the analytical solution to that compartment model is used together with a numerical simulation to compute a relevant value for that parameter; in section 7.3.3, a discrete-time compartment model is build which also depends on one parameter, and a numerical simulation is used to estimate that parameter.

In this section, $\mathbf{c}(t) = (C_1(t), C_2(t))$ denotes the vector of the average concentrations over the compartments:

$$C_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} C(\mathbf{x}, t) \mathrm{d}\Omega_i \quad \text{for } i = 1, 2, \tag{7.5}$$

where $C(\mathbf{x}, t)$ denotes the concentration function. Notice that since we consider a passive tracer in an isolated domain, the mean concentration $\bar{C}(t)$ is constant and

$$\bar{C}(t) = \bar{C} = \frac{|\Omega_1|C_1 + |\Omega_2|C_2}{|\Omega|} = \frac{C_1 + C_2}{2}, \tag{7.6}$$

where we have used the fact that $|\Omega_1| = |\Omega_2| = |\Omega|/2$. Hence, we can express C_2 as a linear function of C_1:

$$C_2 = 2\bar{C} - C_1. \tag{7.7}$$

In the next, it will thus be sufficient to analyze only the quantity C_1, since the error on C_2 is exactly the opposite of the error on C_1.

7.3.1 Limitations of the compartment model

The main limitation of a compartment model is that it only "sees" the average concentration over compartments. In particular, for a same initial condition $C_1(0)$ to the compartment model, an infinity of tracer repartitions within Ω_1 are possible. In this section, we illustrate three different initial repartitions of the particles leading to the same initial condition $C_1(0)$ and thus to the same function $C_1(t)$ in the compartment model.

A first possible initial repartition of the particles is when the tracer mass is uniformly distributed over Ω_1. We denote that initial condition $C^1(0)$. The second case that we consider is when the particles are uniformly distributed over $[-L, -L/2] \times [0, H]$ and there is no particle in $]-L/2, 0[\times[0, H]$; it is denoted $C^2(0)$. The last case is when all the tracer mass is concentrated on a single point, chosen here to be in the lower left corner of the domain (the precise location is $(-\frac{14}{15}L, \frac{1}{10}H)$); it is denoted $C^3(0)$. Those three possible initial repartitions of the particles, all leading to the same compartment initial condition $C_1(0)$ (and thus to the same $C_1(t)$ for all $t \geq 0$), are represented in figure 7.16.

Let $C_1^i(t)$ denote the aggregated concentration over compartment 1 at time t corresponding to the initial repartition of the particles $C^i(0)$ for $i = 1, 2, 3$. The evolutions of $C_1^1(t)$, $C_1^2(t)$ and $C_1^3(t)$ over 1000 years are shown in figure 7.17. The compartment model assumption is that the concentration is approximately uniform over the compartments. Hence, we shall expect the compartment model solution $C_1(t)$ corresponding to the initial condition $C_1(0) = C_1^1(0) = C_1^2(0) = C_1^3(0) = 2\bar{C}$ to be close to $C_1^1(t)$. As expected, $C_1^1(t)$ starts decreasing immediately, since there are already many particles in the exchange zone at $t = 0$. In the two other cases, plateaux are observed. Their lengths correspond to the time for the particles to reach the exchange zone and then possibly enter Ω_2. Obviously, this time is larger in case 3 than in case 2. In both cases, once particles have entered the exchange zone, their concentration in the exchange zone is larger than in case 1, allowing for a larger flux of particles towards Ω_2. Hence, at the end of the plateau, the concentration decreases faster in case 2 than it does initially in case 1 and it decreases still faster in case 3 than in case 2. After 100 years, $C_1^1(t)$ and $C_1^2(t)$ are almost confounded but $C_1^3(t)$ stays clearly different until it reaches equilibrium after approximately 700 years.

The point is that those three cases could never be rendered exactly by a compartment model with two compartments since they correspond exactly to the same compartment solution. Hence, we shall not expect too much from our compartment model: a good compartment model should produce a curve that corresponds approximately to $C_1^1(t)$.

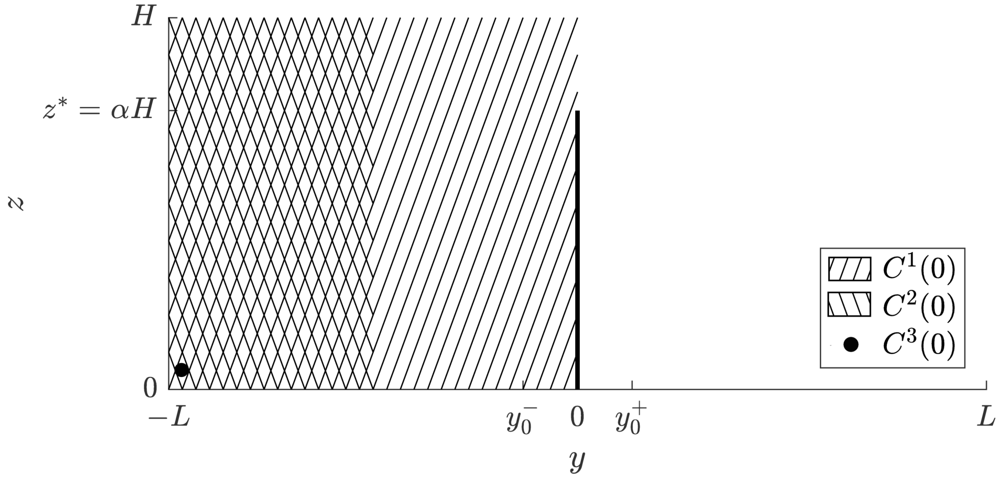


Figure 7.16 – Illustration of the three different initial repartitions of the tracer's particles.

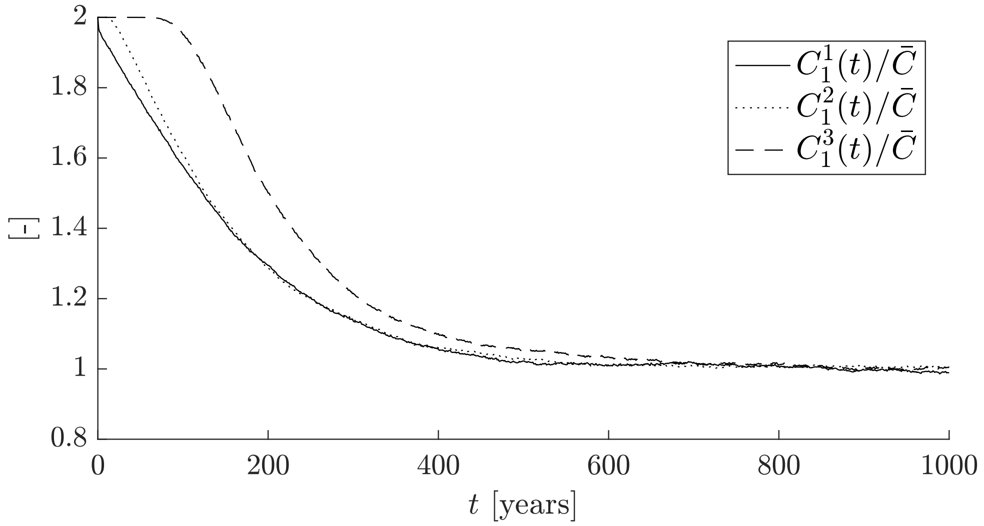


Figure 7.17 – Comparison of the compartment model solution with the numerical solution for the initial condition $C_1(0) = 2\bar{C}$.

7.3.2 Continuous-time compartment model

Let $\mathbf{A}$ be the 2×2 interaction matrix, and let $\mathbf{c}$ and $\mathbf{\Omega}$ be defined as in (3.31). The evolution of the concentration in the two compartments is given by

$$\mathbf{\Omega}\dot{\mathbf{c}} = \mathbf{A}\mathbf{c}. \tag{7.8}$$

It remains to propose an expression for $\mathbf{A}$. In the general case, a 2×2 matrix such as $\mathbf{A}$ has four independent entries. However, properties 3.3.1, 3.3.2 and 3.3.3 shown in chapter 3 imply that $\mathbf{A}$ has only one degree of freedom. Besides, for $a > 0$, $\mathbf{A}$ must have the following form:

$$\mathbf{A} = \begin{pmatrix} -a & a \\ a & -a \end{pmatrix}. \tag{7.9}$$

This is nothing but a straightforward implication of the combination of the three properties. Another way of seeing this is by looking at the general form of **A** shown in (3.33). In this case, that expression reduces to (remember that $U_{i,i} = 0 = V_{i,i}$):

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{2}U_{1,2} - V_{1,2} & -\frac{1}{2}U_{1,2} + V_{1,2} \\ -\frac{1}{2}U_{2,1} + V_{2,1} & -\frac{1}{2}U_{2,1} - V_{2,1} \end{pmatrix}. \tag{7.10}$$

By the continuity equation for the compartment model (3.17), $U_{1,2} = U_{2,1} = 0$, and by the property (3.14) of $V_{i,j}$, $V_{1,2} = V_{2,1} > 0$. With those considerations, (7.10) simplifies to

$$\mathbf{A} = \begin{pmatrix} -V_{1,2} & V_{1,2} \\ V_{1,2} & -V_{1,2} \end{pmatrix}, \tag{7.11}$$

which is exactly (7.9) with $V_{1,2} = a$.

The compartment model considered here has in fact an analytic solution that is relatively easy to compute. Indeed, by (7.7), we can express C_2 as a linear function of C_1 and (7.8) reduces to a simple ODE in C_1 with one parameter a:

$$\frac{dC_1}{dt} = -2aC_1 + 2a\bar{C}, \tag{7.12}$$

where we have redefined a as $V_{1,2}/|\Omega_1|$ in order to simplify the notations. Let $C_{1,0}$ be the initial condition on C_1, namely $C_1(0) = C_{1,0}$. The solution to (7.12) is easily computed as:

$$C_1(t) = (C_{1,0} - \bar{C})\,\mathrm{e}^{-2at} + \bar{C}. \tag{7.13}$$

Let us consider the scaled form of the concentration $\tilde{C}_1 = C_1/\bar{C}$. The solution is expressed in terms of $\tilde{C}_1$ as:

$$\tilde{C}_1(t) = (\tilde{C}_{1,0} - 1)\,\mathrm{e}^{-2at} + 1. \tag{7.14}$$

The goal now is to estimate the parameter a. To this end, we propose the following approach:

1. Run a simulation on a particular instance of the bi-overturner problem (i.e. a particular initial condition).

2. Using that simulation, compute the average concentration in Ω_1 at m different times $t_0 < t_1 < \cdots < t_m$. For the next, let $\bar{C}_1(t)$ denote the average concentration over Ω_1 at time t (whereas $C_1(t)$ stands for the compartment model concentration in the compartment corresponding to the subdomain Ω_1).

3. Apply the method of least squares to compute a:

$$a = \arg\min_a \sum_{i=1}^{m} \left(C_1(t_i) - \bar{C}_1(t_i) \right)^2. \tag{7.15}$$

This method leads to a value of a based on one single particular instance of the problem (i.e. a particular initial condition). Of course, we will need to check that the results obtained for other initial conditions are sufficiently close to the simulations.

The method is applied for the initial condition $\tilde{C}_{1,0} = 2$ (and hence $\tilde{C}_2(0) = 0$), and with the tracer particles initially uniformly distributed over Ω_1. The simulation is run for 1000 years, and $\bar{C}_1(t)$ is evaluated every year. The minimization of (7.15) is performed in MATLAB© using the Gauss-Newton algorithm, which is well-suited to solve non-linear least squares problems.[2] The value

$$a = 0.003125 \tag{7.16}$$

is found. Figure 7.18 compares the evolution of $\tilde{C}_1(t) = C_1(t)/\bar{C}$, the analytical solution of the compartment model, with the evolution of $\tilde{C}_1(t)/\bar{C}$, the scaled mean value of the concentration over Ω_1 computed numerically, which we may consider as an approximation of the exact solution.

todo : autres conditions initiales

7.3.3 Discrete-time compartment model

When there are more than two compartments, finding an analytic solution is not always possible and the methodology presented in the previous section cannot be applied. For this reason, another approach is proposed in this section.

The method

Here a *discrete* interaction matrix $\mathbf{A}_{\Delta t}$ is build based on a numerical simulation over a period Δt. This methodology is thus far more general than the one presented in the previous section. It goes as follows: to compute the entry $[\mathbf{A}_{\Delta t}]_{i,j}$, run a simulation over a period Δt, with all the particles initially uniformly distributed over compartment j (i.e. subdomain Ω_j). Let P_0 be the total number of particles, and $P_{j\to i}(\Delta t)$ the number of particles amongst those initially in compartment j that end up in compartment i after a time Δt. The factor $\frac{P_{j\to i}(\Delta t)}{P_0}$ is an approximation of the probability to go from compartment j to compartment i within a time period Δt. But this probability is precisely given by $\frac{|\Omega_i|}{|\Omega_j|}[\mathbf{A}_{\Delta t}]_{i,j}$, as explained page 21. We have thus the approximation

$$[\mathbf{A}_{\Delta t}]_{i,j} \approx \frac{P_{j\to i}(\Delta t)}{P_0}\frac{|\Omega_j|}{|\Omega_i|}. \tag{7.17}$$

Using that formula to compute an approximation of $\mathbf{A}_{\Delta t}$, it is seen that properties 3.4.1, 3.4.2 and 3.4.5 are always respected *a priori*. Properties 3.4.3 and 3.4.4 should by verified a posteriori.

Results

In a first instance, we can take advantage of the properties developed in section 3.4 to deduce the general form of $\mathbf{A}_{\Delta t}$ for this problem. In the present case, Ω_1 and Ω_2 have the same size; this implies that property 3.4.2 reduces to corollary 3.4.2, and that property 3.4.5 becomes equivalent to property 3.4.4. Putting everything together, the matrix $\mathbf{A}_{\Delta t}$ that we search is a 2×2 matrix whose entries are comprised between 0 and

[2] See for example the wikipedia page of the algorithm : `https://en.wikipedia.org/wiki/Gauss-Newton_algorithm`.

1 and which satisfies $\mathbf{1}^\intercal \mathbf{A}_{\Delta t} = \mathbf{1}^\intercal$ and $\mathbf{A}_{\Delta t}\mathbf{1} = \mathbf{1}$. This implies that $\mathbf{A}_{\Delta t}$ must have the following form:

$$\mathbf{A}_{\Delta t} = \begin{pmatrix} a & 1-a \\ 1-a & a \end{pmatrix}, \tag{7.18}$$

for some $a \in [0,\ 1]$.

The discrete interaction matrix is approximated for $\Delta t = 1$ year with $P_0 = 10\ 000$. Mentionner l'étude de convergence, mais la rendre plus rigoureuse. Using formula (7.17), we get

$$\mathbf{A}_1 = \begin{pmatrix} 0.9839 & 0.0159 \\ 0.0161 & 0.9841 \end{pmatrix}. \tag{7.19}$$

This approximation of $\mathbf{A}_1$ does not exactly match the general form (7.18). Hence, we improve that approximation by choosing a as an average:

$$a = \frac{[\mathbf{A}_1]_{1,1} + (1 - [\mathbf{A}_1]_{1,2}) + (1 - [\mathbf{A}_1]_{2,1}) + [\mathbf{A}_1]_{2,2}}{4} = 0.9840, \tag{7.20}$$

leading to

$$\mathbf{A}_1^{discr,1} = \begin{pmatrix} 0.9840 & 0.0160 \\ 0.0160 & 0.9840 \end{pmatrix}, \tag{7.21}$$

which match the general form (7.18). For any initial condition, we can now approximate the concentration in both compartments after T years as

$$\mathbf{c}^{discr,1}(T) = \left(\mathbf{A}_1^{discr,1}\right)^T \mathbf{c}(0). \tag{7.22}$$

The resulting evolution of the concentration in compartment 1 is shown in figure 7.18 for the initial condition $\mathbf{c}(0) = (2\bar{C}, 0)^\intercal$. Clearly, this result is not satisfying as the concentration decreases much too fast in compartment 1. Therefore, we propose another approach: recall from figure 7.7 that the two-communities clustering is returned by the stability method for times larger than (approximately) 50 years. This suggests that computing the discrete interaction matrix $\mathbf{A}_{50}$ could provide more interesting results. We get numerically

$$\mathbf{A}_{50} = \begin{pmatrix} 0.8753 & 0.1206 \\ 0.1248 & 0.8795 \end{pmatrix}, \tag{7.23}$$

which is corrected as

$$\mathbf{A}_{50} = \begin{pmatrix} 0.87735 & 0.12265 \\ 0.12265 & 0.87735 \end{pmatrix}. \tag{7.24}$$

This discrete interaction matrix only allows to compute the concentration at times that are multiples of 50 years. We have the approximation

$$\mathbf{c}^{discr,2}(T) = (\mathbf{A}_{50})^{\frac{T}{50}}\, \mathbf{c}(0) \qquad \text{for } T = 0, 50, 100, \ldots \tag{7.25}$$

The resulting evolution of the concentration in compartment 1 is also shown in figure 7.18. The result is far better than with the first method. Notice that in this case, approximating $\mathbf{A}_1$ as

$$\mathbf{A}_1^{discr,2} = (\mathbf{A}_{50})^{\frac{1}{50}}, \tag{7.26}$$

yields

$$\mathbf{A}_1^{discr,2} = \begin{pmatrix} 0.9972 & 0.0028 \\ 0.0028 & 0.9972 \end{pmatrix}. \tag{7.27}$$

The resulting approximation of the concentration is the same as in equation (7.25) but now we are able to compute the concentration in the compartments every year. Notice that this does not provide a general method to compute $\mathbf{A}_1$ since we have no a priori guarantee that the entries of $(\mathbf{A}_{\Delta t})^{\frac{1}{\Delta t}}$ are real.

This section presents a procedure to numerically estimate the discrete transition matrix $\mathbf{A}_{\Delta t}$ for any time step Δt. However, the bad results obtained with our first estimation of $\mathbf{A}_1$ show that the choice of Δt is crucial. Let $[t_a,\, t_b]$ denote the time range at which the clustering chosen to delineate the subdomains is found by the stability method. The results of this section suggest that $\Delta t = t_a$ could be a satisfying choice. Unfortunately, no conclusion can be drawn based only on this simple example, and the method would need to be applied on more complicated problems.

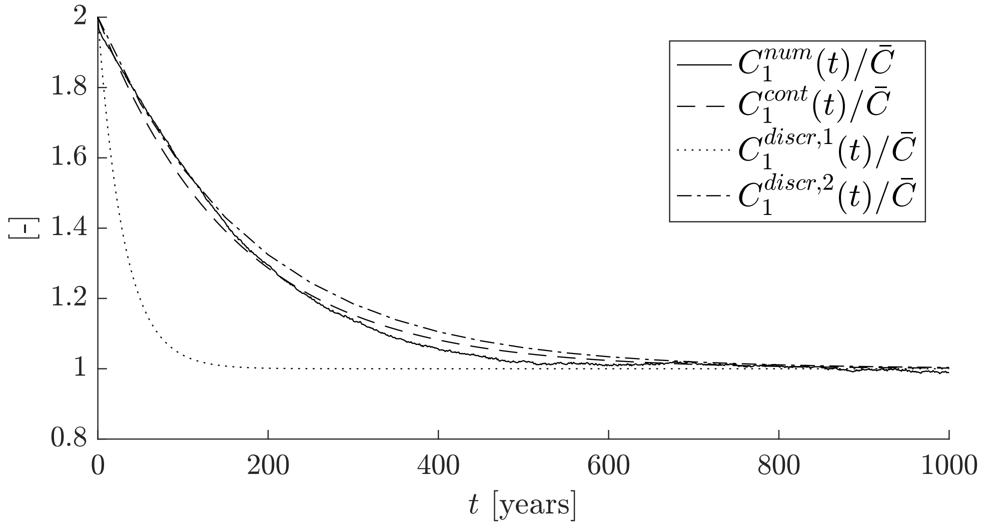


Figure 7.18 – Comparison of the different compartment model solutions with the numerical solution for the initial condition $C_1(0) = 2\bar{C}$.

Conclusion

This work starts with an introduction of the concept of communities in a network. In particular, chapter 1 is entirely dedicated to the *stability measure*, which allows to quantify the quality of a particular clustering:

$$r(t) = \max_{\mathcal{P}} \left\{ \operatorname{trace} \left[H_{\mathcal{P}}^{\intercal} (\mathbf{\Pi} \mathbf{P}(t) - \boldsymbol{\pi}^{\intercal} \boldsymbol{\pi}) \mathbf{H}_{\mathcal{P}} \right] \right\},$$

where t is the Markov time, which acts as an intrinsic resolution parameter. This chapter is intended to be a standalone theoretical chapter providing a complete overview of the stability measure.

In chapter 2, the *reactive transport* equation

$$\frac{\partial C}{\partial t} = q - \nabla \cdot (\mathbf{u} C - \mathbf{K} \nabla C) \,.$$

is presented and the main properties of the solution to that equation are shown. This allows to propose a compartment model structure in chapter 3 that exhibits similar properties:

$$\Omega_i \frac{dC_i}{dt} = \Omega_i q_i - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left[U_{i,j} \frac{C_i + C_j}{2} - V_{i,j} (C_j - C_i) \right] - \phi_{i,e} \quad \text{for } i = 1, \ldots, N.$$

The equivalent matrix form

$$\mathbf{\Omega} \dot{\mathbf{c}} = \mathbf{\Omega} \mathbf{q} + \mathbf{A} \mathbf{c}$$

is also developed for convenience, and the previously mentioned properties are translated into the matrix language.

Anticipating the fact that we do not consider any problem involving reactive phenomena in this work, the attention is then focused on the *advection-diffusion* equation in chapter 4. It is the particular form of the reactive transport equation obtained when the reactive term q is equal to zero. In order to numerically solve such problems, the Eulerian methods are compared to the Lagrangian methods to arrive at the conclusion that a Lagrangian approach is better suited to the needs of this work. An effort is made to derive the Lagrangian equations describing the position of an individual particle which are consistent with the advection-diffusion transport model. This leads to the beautiful and complex theory of stochastic differential equations (SDEs). The backward-Itô interpretation of stochastic differential equations is shown to be particularly useful for dealing with problems involving a discontinuous diffusivity tensor. Indeed, at the contrary to the Itô or Stratonovich interpretations, the backward-Itô interpretation leads to Lagrangian equations that does not include *gradient drift* *terms*, namely terms involving spatial derivatives of the diffusivity tensor:

$$(\mathbf{bI}) \quad \begin{cases} \mathrm{d}\mathbf{x}(t) = \mathbf{u}(\mathbf{x}(t), t)\mathrm{d}t + \mathbf{B}(\mathbf{x}(t), t)\mathrm{d}\mathbf{W}(t), \\ \mathbf{B}\mathbf{B}^\intercal = 2\mathbf{K}, \\ \mathbf{x}(t_0) = \mathbf{x}_0. \end{cases}$$

The most simple numerical scheme consistent with the backward-Itô interpretation of a SDE, the backward Euler scheme, is then presented. The *box-counting* method which allows to estimate the concentration from a set of particles trajectories completes the tools needed to implement an efficient code to numerically solve a two-dimensional advection-diffusion problem. The code is briefly presented in section 4.3.

Finally, the whole method is summarized in chapter 5, allowing to tackle a simple test problem in chapters 6 and 7. The test problem is chosen such that a two compartments decomposition of the domain is intuitively obvious. Satisfactorily, the method leads to the expected box decomposition. For that simple problem, a continuous-time compartment model satisfying the properties from chapter 3 is build relatively easily using an ad-hoc method. A more general approach leading to a discrete-time compartment model is also proposed. Both models are assessed by comparing their results to the ones obtained by numerical simulations.

For the simple test problem considered, the method leads to a satisfying two-compartments model. It would be presumptuous to draw conclusions on the method based only on the results obtained for such a simple test problem. Instead, this encouraging result strengthens our idea that further works are worthwhile. Ideas of problems on which the method may be tested include the idealized two-dimensional model of the meridian circulation in the Atlantic [53], the Great Barrier Reef, Australia [14] or the Pacific Ocean [40].

To conclude, let us mention the several prospects of amelioration that comes to our mind at the time of finishing this work. As already raised, the method should be applied on more complex problems, and at the very least on problems having a physical meaning. Furthermore, it could be interesting to compare the results obtained when using different clustering algorithms. For example, the *infomap* algorithm proposed by Martin Rosvall also seems well adapted to this study.[3] Finally, we could extend the C++ code towards a complete toolbox for the simulation of two-dimensional SDEs. Such a toolbox should allow to choose the boundary conditions and to deal with the reactive term, and it could integrate higher order numerical schemes like Milstein's scheme, but this is of course outside the scope of the present work. A parallelization of the code with MPI also figures amongst the perspective of further work that have been considered.[4]

[3] See the excellent website `http://www.mapequation.org` for a presentation of the algorithm and links to the related publications.

[4] In fact, an MPI version of the code has actually been implemented. But we have not managed to test that implementation with many nodes because of incompatibility problems with the versions of MPI on the clusters, which are still not solved at time time of finishing this work.

Bibliography

[1] Eric Deleersnijder, Jia Wang, and Christopher NK Mooers. A two-compartment model for understanding the simulated three-dimensional circulation in prince william sound, alaska. *Continental Shelf Research*, 18(2):279–287, 1998.

[2] V. Maderich, R. Bezhenar, R. Heling, G. de With, K. T. Jung, J. G. Myoung, Y.-K. Cho, F. Qiao, and L. Robertson. Regional long-term model of radioactivity dispersion and fate in the northwestern pacific and adjacent seas: application to the fukushima dai-ichi accident. *Journal of environmental radioactivity*, 131:4–18, 2014.

[3] Karline Soetaert and Peter M. J. Herman. Estimating estuarine residence times in the westerschelde (the netherlands) using a box model with fixed dispersion coefficients. *Hydrobiologia*, 311(1):215–224, 1995.

[4] Peter Köhler, Hubertus Fischer, Guy Munhoven, and Richard E. Zeebe. Quantitative interpretation of atmospheric carbon records over the last glacial termination. *Global Biogeochemical Cycles*, 19(4), 2005.

[5] Guy Munhoven and Louis M. François. Glacial-interglacial variability of atmospheric co2 due to changing continental silicate rock weathering: A model study. *Journal of Geophysical Research: Atmospheres*, 101(D16):21423–21437, 1996.

[6] Mark E. J. Newman. *Networks: an introduction*. Oxford university press, 2010.

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

[8] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[9] Petter Holme, Mikael Huss, and Hawoong Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538, 2003.

[10] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *nature*, 433(7028):895, 2005.

[11] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *arXiv preprint physics/0506133*, 2005.

[12] Gary William Flake, Steve Lawrence, C Lee Giles, and Frans M Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002.

[13] Mikael Huss and Petter Holme. Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks. *IET systems biology*, 1(5):280–285, 2007.

[14] Christopher J. Thomas, Jonathan Lambrechts, Eric Wolanski, Vincent A. Traag, Vincent D. Blondel, Eric Deleersnijder, and Emmanuel Hanert. Numerical modelling and graph theory tools to study ecological connectivity in the great barrier reef. *Ecological Modelling*, 272:160–174, 2014.

[15] Martin Nilsson Jacobi, Carl André, Kristofer Döös, and Per R. Jonsson. Identification of subpopulations from connectivity matrices. *Ecography*, 35(11):1004–1016, 2012.

[16] Vincent Rossi, Enrico Ser-Giacomi, Cristóbal López, and Emilio Hernández-García. Hydrodynamic provinces and oceanic connectivity from a transport network help designing marine reserves. *Geophysical Research Letters*, 41(8):2883–2891, 2014.

[17] 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.

[18] 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.

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

[20] Eric Deleersnijder. The unreasonable effectiveness of dimension reduction in complex geophysical flow modelling. Presentation slides, 2009. Belgian National Committee for Geodesy and Geophysics.

[21] 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.

[22] 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.

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

[24] Eric Deleersnijder, Jean-Michel Campin, and Eric J.M. Delhez. The concept of age in marine modelling: I. theory and preliminary model results. *Journal of Marine Systems*, 28(3):229–267, 2001.

[25] Eric Deleersnijder. On the structure of a compartment model for tracer transport. 2014. Available at `http://hdl.handle.net/2078.1/155581`.

[26] Eric J.M. Delhez. A matrix approach of compartment models. 2010.

[27] Jean Van Schaftingen. Lmat1223: Équations différentielles ordinaires. Lecture notes, 2013.

[28] 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.

[29] D. Spivakovskaya, A. W. Heemink, and E. Deleersnijder. The backward îto method for the lagrangian simulation of transport processes with large space variations of the diffusivity. *Ocean Science Discussions*, 4(4):623–652, 2007.

[30] T. L. Van Stijn. Positive advection schemes for environmental studies. *Numerical methods in laminar and turbulent flow*, 1987.

[31] Y. Yang, L. T. Wilson, M. E. Makela, and M. A. Marchetti. Accuracy of numerical methods for solving the advection–diffusion equation as applied to spore and insect dispersal. *Ecological modelling*, 109(1):1–24, 1998.

[32] Chunmiao Zheng, Gordon D. Bennett, et al. *Applied contaminant transport modeling*, volume 2. Wiley-Interscience New York, 2002.

[33] J. R. Hunter. The application of lagrangian particle-tracking techniques to modelling of dispersion in the sea. *North-Holland Mathematics Studies*, 145:257–269, 1987.

[34] J. R. Hunter, P. D. Craig, and H. E. Phillips. On the use of random walk models with spatially variable diffusivity. *Journal of Computational Physics*, 106(2):366–376, 1993.

[35] G. T. Yeh. A lagrangian-eulerian method with zoomable hidden fine-mesh approach to solving advection-dispersion equations. *Water Resources Research*, 26(6):1133–1144, 1990.

[36] Eric M. LaBolle, Graham E. Fogg, and Andrew F. B. Tompson. Random-walk simulation of transport in heterogeneous porous media: Local mass-conservation problem and implementation methods. *Water Resources Research*, 32(3):583–593, 1996.

[37] C. W. Gardiner. Stochastic methods. *Springer-Verlag*, 1985.

[38] Arnold Heemink and et al. Modeling transport processes using stochastic differential equations. Slides, 2011.

[39] Roland Keunings. Mathematical modeling of physical problems. Lecture notes, 2016.

[40] Syed Hyder Ali Muttaqi Shah, François W. Primeau, Eric Deleersnijder, and Arnold W. Heemink. Tracing the ventilation pathways of the deep north pacific ocean using lagrangian particles and eulerian tracers. *Journal of Physical Oceanography*, 47(6):1261–1280, 2017.

[41] Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlin Heidelberg New York, 1995.

[42] Pere Colet and Raúl Toral. *Stochastic numerical methods: an introduction for students and scientists*. John Wiley & Sons, 2014.

[43] Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

[44] Eric M. LaBolle, Jeremy Quastel, Graham E. Fogg, and Janko Gravner. Diffusion processes in composite porous media and their numerical integration by random walks: Generalized stochastic differential equations with discontinuous coefficients. *Water Resources Research*, 36(3):651–662, 2000.

[45] Andrew F. B. Tompson and D. E. Dougherty. Particle-grid methods for reacting flows in porous media with application to fisher's equation. *Applied mathematical modelling*, 16(7):374–383, 1992.

[46] Peter de Haan. On the use of density kernels for concentration estimations within particle and puff dispersion models. *Atmospheric Environment*, 33(13):2007–2021, 1999.

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

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

[49] M. P. Wand and M. C. Jones. Kernel smoothing. 1995. *Chapman&Hall, London*, 1995.

[50] Thomas A. Prickett, Carl G. Lonnquist, Thomas G. Naymik, et al. A" random-walk" solute transport model for selected groundwater quality evaluations. *Bulletin/Illinois State Water Survey; no. 65*, 1981.

[51] Andrew F. B. Tompson, Efstratios G. Vomvoris, and Lynn W. Gelhar. Numerical simulation of solute transport in randomly heterogeneous porous media: motivation, model development, and application. Technical report, Lawrence Livermore National Lab., CA (USA), 1987.

[52] Joe Pitt-Francis and Jonathan Whiteley. *Guide to scientific computing in C++*. Springer Science & Business Media, 2012.

[53] Eric Deleersnijder. Éléments d'un modèle latitude-profondeur très simple — application à l'injection de "co2 idéalisée" dans l'océan. 2006. Available at `http://hdl.handle.net/2078.1/155304`.

[54] Catherine Timmermans. Ventilation et rétention dans un bassin océanique idéalisé. Master's thesis, UCL, 2006.

[55] Mark E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review*, 74(3):036104, 2006.

[56] Eric Deleersnijder. Modelling transport processes in geophysical and environmental flows. Lecture notes, 2014. UCL.

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

[58] 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.

List of Figures

Appendix A

Stochastic Differential Equations

A.1 Stieltjes integral

The Riemann-Stieltjes integral is a generalization of the Riemann integral. Let f and g be real-valued functions defined on a closed interval $[a,\, b]$. The Riemann-Stieltjes integral of f with respect to g is denoted

$$\int_a^b f(t)\mathrm{d}g(t). \tag{A.1}$$

Consider a partition of the interval

$$a = t_0 < t_1 < \cdots < t_{n-1} < t_n = b, \tag{A.2}$$

and define

$$h_n \triangleq \max_{i\in\{1,2,\ldots,n\}}(t_i - t_{i-1}). \tag{A.3}$$

Now take the Riemann sum

$$\sum_{i=1}^{n} f(\tau_i)[g(t_i) - g(t_{i-1})], \tag{A.4}$$

with $\tau_i \in [t_{i-1},\, t_i]$. If the sum tends to a fixed number I as $n \to \infty$ and $h_n \to 0$, then

$$\int_a^b f(t)\mathrm{d}g(t) = I. \tag{A.5}$$

A.2 Mean-square limit

Suppose that we have a probability space Ω, and a sequence of random variables X_n defined on Ω. We say that X_n converges to X in the mean-square sense if

$$\lim_{n\to\infty} \left\langle (X_n - X)^2 \right\rangle = 0, \tag{A.6}$$

and we note

$$\operatorname*{ms-lim}_{n\to\infty} X_n = X. \tag{A.7}$$

Appendix B

Test cases to assess the implementation of the Lagrangian solver

Two test cases adapted from *Eric Deleersnijder*'s working paper [57] are presented here: after introducing the problems, the analytic solutions are computed as well as their main properties. The code is then run on those problems, and the numerical solution and its properties are compared to the analytical ones. Rather than a validation of the numerical solver, this can be considered as a *sanity check*.

B.1 Governing equations

Let us consider a water domain, whose width is denoted $B(\mathbf{x}, t)$, where $\mathbf{x} = (y, z)$ is the position vector and t is the time. The continuity equation is

$$\frac{\partial B}{\partial t} + \nabla \cdot (B\mathbf{u}) = 0, \tag{B.1}$$

where $\mathbf{u}(t, \mathbf{x})$ is the latitudinally-averaged meridional velocity. Assuming that mixing along the parallels is sufficiently efficient, we may study the concentration of a passive tracer by means of a two-dimensional model. The latitudinally-averaged concentration of the tracer $C(\mathbf{x}, t)$ obeys the following partial differential equation :

$$\frac{\partial (BC)}{\partial t} + \nabla \cdot (B\mathbf{u}C) = Q\delta(\mathbf{x} - \mathbf{x}_1) + \nabla \cdot (B\mathbf{K} \cdot \nabla C), \tag{B.2}$$

where $\mathbf{K}$ is the diffusivity tensor (symmetric and positive definite); δ is the Dirac delta function with $\delta(\mathbf{x} - \mathbf{x}_n) = \delta(x - x_n)\delta(y - y_n)$; $Q(t)$ is the rate of release of a lineic source of length B along the latitude direction located at $\mathbf{x} = \mathbf{x}_1$. If $C(\mathbf{x}, t)$ represents the density of the tracer in water, then $Q(t)$ is the mass of tracer released per second by the source.

Equation (B.2) is the so-called conservative form of the model. The convective form is obtained by combining equations (B.1) and (B.2):

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \frac{Q}{B}\delta(\mathbf{x} - \mathbf{x}_1) + \frac{1}{B}\nabla \cdot (B\mathbf{K} \cdot \nabla C). \tag{B.3}$$

B.2 An idealized model

For our test case to be interesting, we must be able to compute its analytical solution. Accordingly, we make some simplifying assumptions which will allow us to compute the solution analytically. First, we assume a constant width B and a constant velocity field

$$\mathbf{u}(t, \mathbf{x}) = v\mathbf{e}_y + w\mathbf{e}_z, \tag{B.4}$$

where $\mathbf{e}_y$ and $\mathbf{e}_z$ are the unit vectors associated respectively with the y- and z-coordinate axis. Furthermore, the diffusivity tensor is supposed constant and diagonal :

$$\mathbf{K} = \begin{pmatrix} K_{yy} & 0 \\ 0 & K_{zz} \end{pmatrix}, \tag{B.5}$$

where $K_{yy},\ K_{zz} > 0$. Finally, we consider a sudden pointwise release of tracer at $t = 0$. Hence, $Q(t)$ is of the form :

$$Q(t) = M\delta(t), \tag{B.6}$$

where M is the mass of tracer released at $t = 0$.

Under these assumptions, equation (B.3) simplifies to :

$$\frac{\partial C}{\partial t} + v\frac{\partial C}{\partial y} + w\frac{\partial C}{\partial z} = J\delta(t)\delta(y - y_1)\delta(z - z_1) + K_{yy}\frac{\partial^2 C}{\partial y^2} + K_{zz}\frac{\partial^2 C}{\partial z^2}, \tag{B.7}$$

where $J := M/B$. For the sake of simplicity, we can forget about the fact that our model is width-integrated and consider that it is a purely two-dimensional model with a point-source

$$Q := J\delta(t). \tag{B.8}$$

A part of the physical meaning of the model is lost but this makes representations of the problem easier. C now represents the two-dimensional density (i.e., in $[kg/m^2]$) of the tracer in water. J can then be regarded as the mass of tracer released by the sudden point source at $\mathbf{x} = \mathbf{x}_1$. The three- and two-dimensional interpretations of the problem are represented on figure B.1.

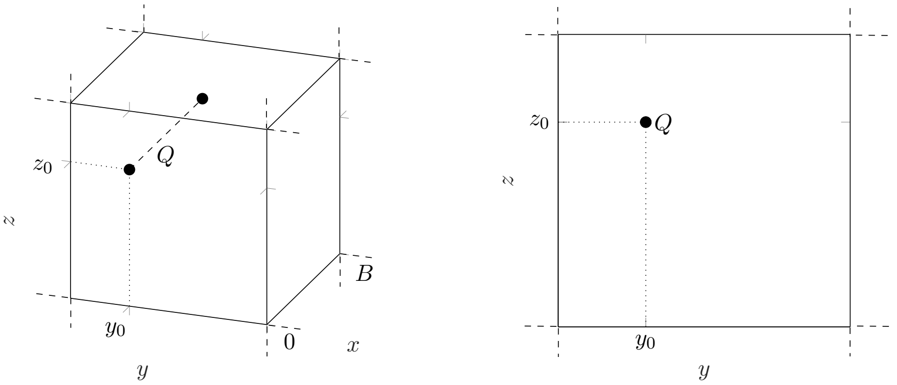


Figure B.1 – Illustration of the 3D and 2D interpretations of the model.

B.2.1 Test case 1 : infinite domain

The first (an most simple) test case consists in considering an infinite domain, i.e.

$$-\infty < y,\, z < \infty, \tag{B.9}$$

with nonzero velocities v and w. This test case provides a check that our numerical implementation handles the diffusion and advection processes properly both in the y- and z-directions. The parameters are chosen from the values proposed by *C. Timmermans* for the overturner model in her master's thesis [54]:

$$v = \frac{\Psi}{H} = 4\times10^{-4}\ [m/s], \quad w = \frac{\Psi}{L} = 1.33\times10^{-7}\ [m/s], \tag{B.10}$$

and

$$K_{yy} = 10^{3}\ [m^2/s], \quad K_{zz} = 10^{-4}\ [m^2/s]. \tag{B.11}$$

For the length scales of the overturner model, those diffusivities corresponds to Péclet numbers

$$Pe_y = \frac{v}{K_{yy}/L} = 6, \quad Pe_z = \frac{w}{K_{zz}/H} = 6.67. \tag{B.12}$$

Hence, in both the y- and z-directions, the transport is neither dominated by advection nor by diffusion. This is interesting as a test case since it allows to assess how the numerical solver handles both physical processes in both directions. Finally, $J = 10\,000$ particles are released at $t = 0$ at the location $(y_1, z_1) = (0, 0)$.

B.2.2 Test case 2 : semi-infinite domain

Another interesting case is to consider a semi-infinite domain with a wall at $z = 0$:

$$-\infty < y < \infty, \quad 0 < z < \infty. \tag{B.13}$$

This is useful to assess how our numerical model handles no-through boundary conditions. Again, the parameters values are related to the ones proposed in *C. Timmermans*'s master's thesis for the overturner model :

$$v = \frac{\Psi}{H} = 4\times10^{-4}\ [m/s], \quad w = 0\ [m/s], \tag{B.14}$$

and

$$K_{yy} = 10^{3}\ [m^2/s], \quad K_{zz} = 10^{-1}\ [m^2/s]. \tag{B.15}$$

Notice the choice of K_{zz} : it is chosen 10^3 times larger than the value chosen for test case 1. The goal here is to assess that the boundary condition is well handled by the solver. Since $w = 0$, only the vertical diffusivity could possibly drive the particles towards the wall. By increasing K_{zz}, we ensure that more particles will bounce against the wall, which is relevant in this context. $J = 10\,000$ particles are released at $t = 0$ at the location $(y_1, z_1) = (0, H)$.

B.3 Analytical solution and properties

B.3.1 Green's function

In order to build the analytical solution of the problem, we need to compute the Green's function associated to this particular problem. We derive the Green's function G associated to test case 1. We will show later how this function can be used to compute the concentration for both test case 1 and test case 2. $G(\mathbf{x}, t, t')$ is zero for $t < t'$ and is the solution of

$$\begin{cases} \frac{\partial G}{\partial t} + v\frac{\partial G}{\partial y} + w\frac{\partial G}{\partial z} = K_{yy}\frac{\partial^2 G}{\partial y^2} + K_{zz}\frac{\partial^2 G}{\partial z^2} \\ G(y, z, t, t')|_{t=t'} = \delta(y)\delta(z) \end{cases} \tag{B.16}$$

for $t \geq 0$, and on an infinite domain $\infty < y,\, z < \infty$. It can be shown that

$$G(y, z, t, t') = \frac{\exp\left[-\frac{(y-s_v)^2}{4K_{yy}\tau} - \frac{(z-s_w)^2}{4K_{zz}\tau}\right]}{4\pi\sqrt{K_{yy}K_{zz}}\tau}, \tag{B.17}$$

where $\tau = t - t'$ and

$$\mathbf{s}(t, t') = (s_v(t, t'), s_w(t, t')) = \left(\int_{t'}^{t} v\mathrm{d}\xi, \int_{t'}^{t} w\mathrm{d}\xi\right) = (v\tau, w\tau)\,. \tag{B.18}$$

G has some interesting properties. The "mass" of the solution is

$$m(t, t') \equiv \int_{\mathbb{R}^2} G(\mathbf{x}, t, t')\mathrm{d}\mathbf{x} = 1. \tag{B.19}$$

The "center of mass" is located at

$$\mathbf{r}(t, t') \equiv \frac{1}{m(t, t')} \int_{\mathbb{R}^2} \mathbf{x}G(\mathbf{x}, t, t')\mathrm{d}\mathbf{x} = \mathbf{s}(t, t'). \tag{B.20}$$

The variance of the solution is

$$\sigma^2(t, t') \equiv \frac{1}{m(t, t')} \int_{\mathbb{R}^2} |\mathbf{x} - \mathbf{r}(t, t')|^2 G(\mathbf{x}, t, t')\mathrm{d}\mathbf{x} = 2(K_{yy} + K_{zz})\tau. \tag{B.21}$$

B.3.2 Test case 1

The analytical solution of test case 1 is now obtained with the help of the Green's function derived above by computing the convolution between G and the source terms :

$$\begin{aligned} C(\mathbf{x}, t) &= \int_0^t \int_{\mathbb{R}^2} G(\mathbf{x} - \mathbf{x}', t, t') J\delta(t)\delta(\mathbf{x} - \mathbf{x}_1)\mathrm{d}\mathbf{x}'\mathrm{d}t' \\ &= JG(\mathbf{x} - \mathbf{x}_1, t, 0). \end{aligned} \tag{B.22}$$

The concentration profile for test case 1 is thus

$$C(y, z, t) = \frac{J}{4\pi\sqrt{K_{yy}K_{zz}}t} \exp\left[-\frac{(y - s_v)^2}{4K_{yy}t} - \frac{(z - s_w)^2}{4K_{zz}t}\right]. \tag{B.23}$$

The total mass of tracer present in the domain is

$$m(t) \equiv \int_{\mathbb{R}^2} C(\mathbf{x}, t)\mathrm{d}\mathbf{x} = J. \tag{B.24}$$

Note that this number is independent of the transport processes.

The mass center is located at

$$\begin{aligned} \mathbf{r}(t) &\equiv \frac{1}{m(t)} \int_{\mathbb{R}^2} \mathbf{x}C(\mathbf{x}, t)\mathrm{d}\mathbf{x} \\ &= \int_{\mathbb{R}^2} \mathbf{x}G(\mathbf{x} - \mathbf{x}_1, t, 0)\mathrm{d}\mathbf{x} \\ &= \int_{\mathbb{R}^2} (\mathbf{x} - \mathbf{x}_1)G(\mathbf{x} - \mathbf{x}_1, t, 0) + \mathbf{x}_1 G(\mathbf{x} - \mathbf{x}_1, t, 0)\mathrm{d}\mathbf{x} \\ &= \mathbf{x}_1 + \mathbf{s}(t, 0), \end{aligned} \tag{B.25}$$

where properties (B.19) and (B.20) are used to perform the last step.

Finally, the variance of the solution is

$$\begin{aligned}\sigma^2(t) &= \frac{1}{m(t)} \int_{\mathbb{R}^2} |\mathbf{x} - \mathbf{r}(t)|^2 C(\mathbf{x}, t) \mathrm{d}\mathbf{x} \\ &= \int_{\mathbb{R}^2} |(\mathbf{x} - \mathbf{x}_1) - \mathbf{s}(t, 0)|^2 G(\mathbf{x} - \mathbf{x}_1, t, 0) \mathrm{d}\mathbf{x} \\ &= 2(K_{yy} + K_{zz})t, \end{aligned} \tag{B.26}$$

where property (B.21) is used.

B.3.3 Test case 2

To compute the solution to test case 2, a little trick must be applied. Consider the problem on an infinite domain with two sudden point sources of equal intensity located at $z = H$ and $z = -H$. By symmetry, one can see that the concentration of that problem in the region $[-\infty, \infty] \times [0, \infty]$ is precisely the concentration of test case 2. Hence, we can use the Green's function G derived for test case 1 to compute the concentration. In this case, the convolution has to be performed with two point sources :

$$\begin{aligned} C(\mathbf{x}, t) &= \int_0^t \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J\delta(t')\delta(y' - y_1) \left[\delta(z' - z_1) + \delta(z' + z_1)\right] G(y - y', z - z', t, t') \mathrm{d}y' \mathrm{d}z' \mathrm{d}t' \\ &= J \int_0^t \delta(t')[G(y - y_1, z - z_1, t, t') + G(y - y_1, z + z_1, t, t')] \\ &= J[G(y - y_1, z - z_1, t, 0) + G(y - y_1, z + z_1, t, 0)]. \end{aligned} \tag{B.27}$$

The concentration of the tracer for test case 2 is thus

$$C(y, z, t) = \frac{J}{4\pi\sqrt{K_{yy}K_{zz}}t} \exp\left[-\frac{(y - s_v)^2}{4K_{yy}t}\right] \left\{\exp\left[-\frac{(z - z_1)^2}{4K_{zz}t}\right] + \exp\left[-\frac{(z + z_1)^2}{4K_{zz}t}\right]\right\} \tag{B.28}$$

The mass is obtained as

$$m(t) \equiv \int_0^{\infty} \int_{-\infty}^{\infty} C(y, z, t) \mathrm{d}y \mathrm{d}z = J, \tag{B.29}$$

i.e. the number of particles released at $t = 0$. This result is obvious since there is no other source or sink, and we impose a no-through condition at the boundary.

No analytic solution has been found for the center of mass and the variance. However, an accurate estimation of those quantities can be computed numerically from the expression of the concentration.

B.4 Validation of the numerical solver

This section aims to show that the numerical results obtained with the solver are in good agreement with the analytical ones. As the combination of the two test cases cover the main features of the problems considered in this work, this constitutes a basic validation of the solver.

Both test cases are simulated for 1 year with a time step of 1 hour, and $P = 10\,000$ particles are released at $t = 0$. The concentration is computed at the final time $T = 1$

year on the domain $\Omega = [y_{min}, y_{max}] \times [z_{min}, z_{max}]$, where the subscripts *min* and *max* stands for the minimal and maximal position at time T amongst all the particles. The domain Ω is divided into 20×20 boxes denoted Ω_i for $i = 1, \dots, 200$. Let $C_i(t)$ denote the concentration in box i at time t and $C_{max}(t) = \max_i C_i(t)$. The scaled concentration in box i is computed as

$$\tilde{C}_i(T) := \frac{C_i(T)}{C_{i,max}(T)} = \frac{P_i(T)}{P_{max}(T)}, \tag{B.30}$$

where $P_i(t)$ is the number of particles in box i at time t and $P_{max}(t) = \max_i P_i(t)$. The last equality is true because all the boxes have the same volume. The analytic solutions of the concentration are scaled identically:

$$\tilde{C}(t) = \frac{C(t)}{C_{max}(t)}. \tag{B.31}$$

B.4.1 Test case 1

Figure B.2 shows a comparison between the numerical result and the analytical solution for the scaled concentration $C(\mathbf{x}, T)/\tilde{C}$. Dark (resp. light) shaded areas correspond to zones where the numerically computed concentration is "above" (resp. "below") the exact concentration. It would probably have been more rigorous to represent the numerically computed concentration as a three-dimensional histogram since the box-counting method computes the mean value of the concentration in each box. However, here we have considered that the numerically computed mean concentration in a box corresponds to the concentration at the center of that box. The continuous representation of $\tilde{C}_{num}$ is then obtained by interpolation. Figures B.3a and B.3b represent respectively a cut of the concentrations at fixed $y = r_{y,exact}$ and at fixed $z = r_{z,exact}$. The numerically computed concentration $\tilde{C}_{num}$ seems to be an appreciable approximation of the exact concentration $\tilde{C}_{exact}$. To be more specific, the maximal local error is

$$\|\tilde{C}_{exact} - \tilde{C}_{num}\|_\infty = 7.017 \times 10^{-2}. \tag{B.32}$$

The centers of mass are located at

$$\mathbf{r}_{exact} = (12\,614.4,\ 4.2)\ [m], \quad \mathbf{r}_{num} = (11\,715.3,\ 4.5)\ [m]. \tag{B.33}$$

The relative error is

$$\mathbf{e}_r = \left| \frac{\mathbf{r}_{exact} - \mathbf{r}_{num}}{\mathbf{r}_{exact}} \right| = \left(7.13 \times 10^{-2}, 7.89 \times 10^{-2}\right), \tag{B.34}$$

where the division is taken element-wise on the vectors. The euclidean norm of the relative error is

$$\|\mathbf{e}_r\|_2 = 1.06 \times 10^{-1}. \tag{B.35}$$

This can be seen as a quantification of the error on advection. To quantify the error on diffusion, we compute the variance of the concentration :

$$\sigma^2_{exact} = 6.31 \times 10^{10}\ [m^2], \quad \sigma^2_{num} = 6.43 \times 10^{10}\ [m^2]. \tag{B.36}$$

The relative error is

$$e_{\sigma^2} = \left| \frac{\sigma^2_{exact} - \sigma^2_{num}}{\sigma^2_{exact}} \right| = 1.86 \times 10^{-2}. \tag{B.37}$$

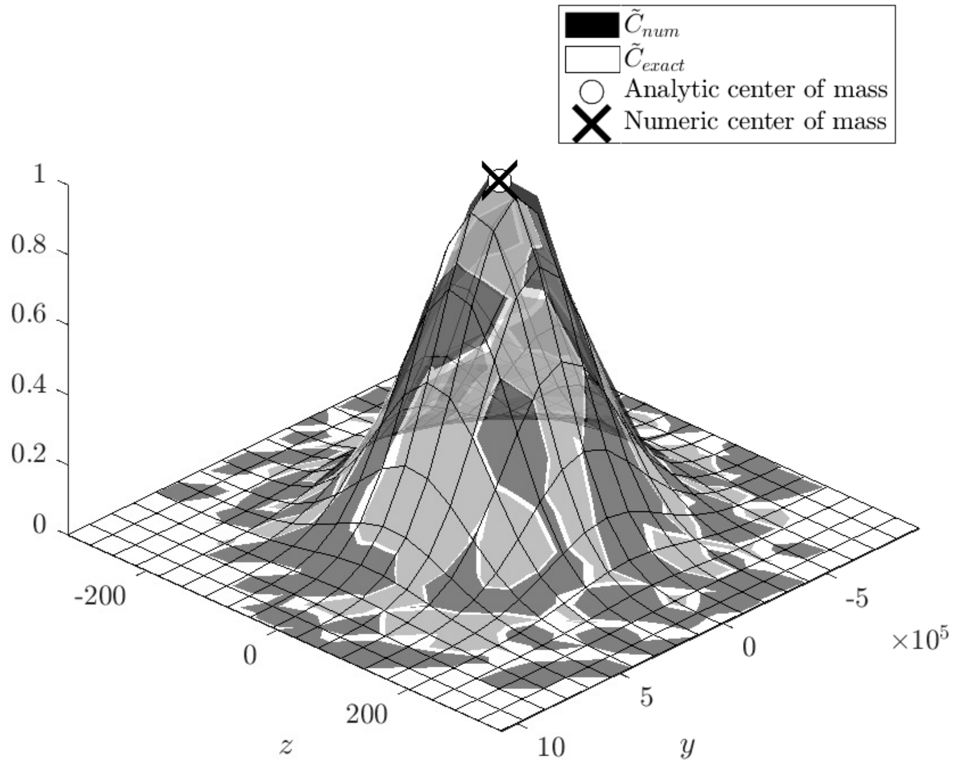
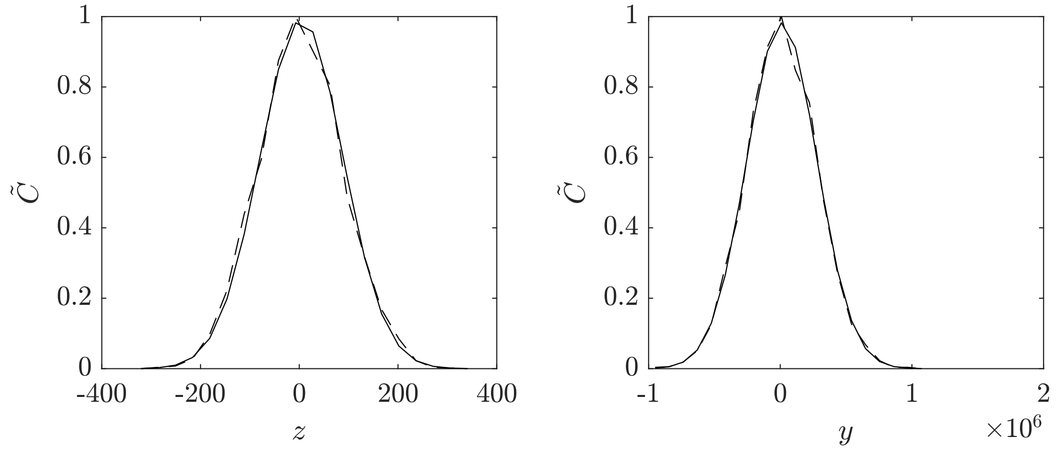


Figure B.2 – Comparison of the concentrations obtained analytically and numerically. The "centers of mass" of the concentration obtained numerically (black cross) and numerically (white bullet) are also shown on the figure.



(a) $\tilde{C}_{num}(y_1 + vT, z)$ and $\tilde{C}_{exact}(y_1 + vT, z)$. **(b)** $\tilde{C}_{num}(y, z_1 + wT)$ and $\tilde{C}_{exact}(y, z_1 + wT)$.

Figure B.3 – Cut of the concentrations at fixed $y = r_{y,exact}$ and at fixed $z = r_{z,exact}$. The dashed line represent $\tilde{C}_{num}$ and the continuous line is for $\tilde{C}_{exact}$.

B.4.2 Test case 2

Figure B.4 shows a comparison between the numerical result and the analytical solution for the scaled concentrations for test case 2. The same remark as the one made for test case 1 applies concerning the representation of the numerically computed concentration. Since we do not have analytical expressions of the center of mass and of the variance for this test case, the "exact" values are approximated numerically thanks to the analytical expression of the concentration. Figures B.5a and B.5b represent respectively a cut of the concentrations at fixed $y = r_{y,exact}$ and along the boundary $z = 0$. The big picture about test case 2 is the presence of a boundary with no-through condition. A first verification is to check if all the particles are still in the domain, which is indeed the case. Although this might seem trivial here, it is sometimes a real challenge to ensure that no particle crosses the boundary, especially for geometrically complex domains. Besides, one can see on figure B.5b that the concentration profile is well approximated along the boundary. Indeed, the maximal local error at the boundary is

$$\|\tilde{C}_{exact}(y,0) - \tilde{C}_{num}(y,0)\|_\infty = 2.964\times10^{-2}, \tag{B.38}$$

The maximal local error on the whole domain is

$$\|\tilde{C}_{exact} - \tilde{C}_{num}\|_\infty = 8.668\times10^{-2}. \tag{B.39}$$

The centers of mass are located at

$$\mathbf{r}_{exact} = (1.26\times10^4, 5.06\times10^3)\ [m], \quad \mathbf{r}_{num} = (1.37\times10^4, 5.05\times10^3)\ [m]. \tag{B.40}$$

The relative error is

$$\mathbf{e}_r = \left|\frac{\mathbf{r}_{exact} - \mathbf{r}_{num}}{\mathbf{r}_{exact}}\right| = \left(8.566\times10^{-2}, 2.718\times10^{-3}\right). \tag{B.41}$$

The euclidean norm of the relative error is

$$\|\mathbf{e}_r\|_2 = 8.571\times10^{-2}. \tag{B.42}$$

This can be seen as a quantification of the error on advection. To quantify the error on diffusion, we compute the variance of the concentration :

$$\sigma^2_{exact} = 6.39\times10^{10}\ [m^2], \quad \sigma^2_{num} = 6.25\times10^{10}\ [m^2]. \tag{B.43}$$

The relative error is

$$e_{\sigma^2} = \left|\frac{\sigma^2_{exact} - \sigma^2_{num}}{\sigma^2_{exact}}\right| = 2.269\times10^{-2}. \tag{B.44}$$

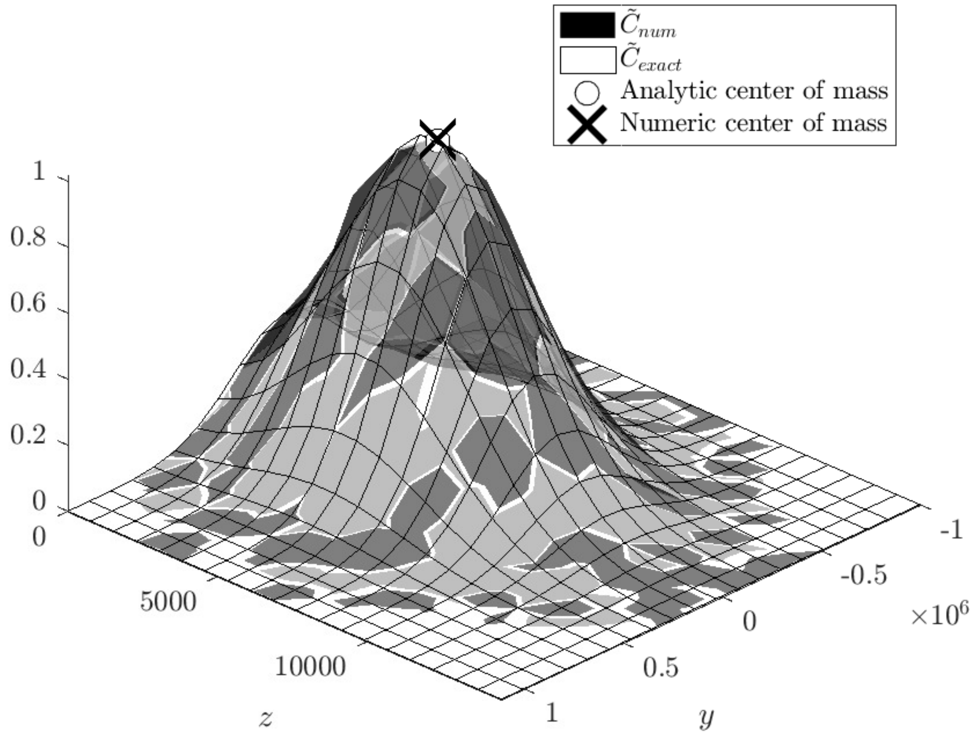
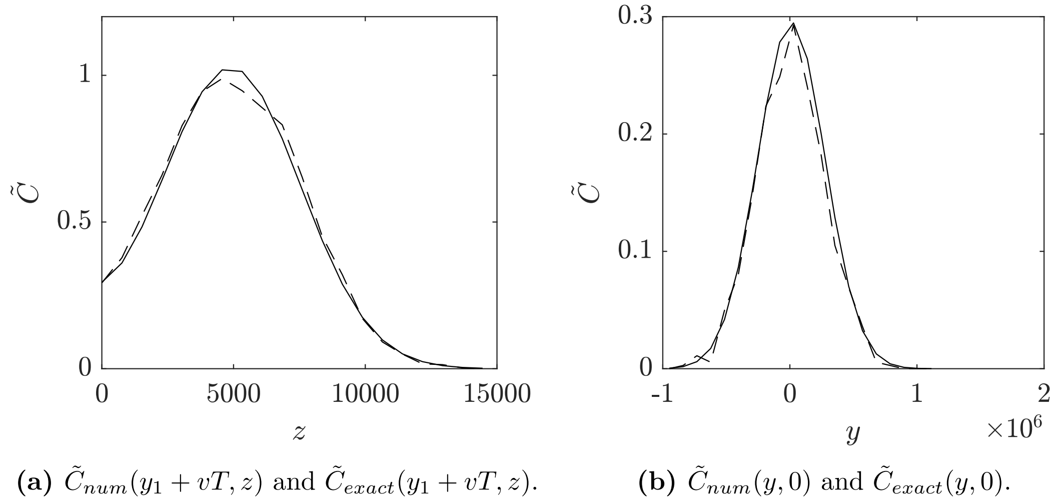


Figure B.4 – Comparison of the concentrations obtained analytically and numerically. The "centers of mass" of the concentration obtained numerically (black cross) and numerically (white bullet) are also shown on the figure.



(a) $\tilde{C}_{num}(y_1 + vT, z)$ and $\tilde{C}_{exact}(y_1 + vT, z)$.

(b) $\tilde{C}_{num}(y, 0)$ and $\tilde{C}_{exact}(y, 0)$.

Figure B.5 – Cut of the concentrations at fixed $y = r_{y,exact}$ and along the wall (fixed $z = 0$). The dashed line represent $\tilde{C}_{num}$ and the continuous line is for $\tilde{C}_{exact}$.

Appendix C

The streamfunction

The interest of the streamfunction comes from the following property.

Property C.1. *For any 3-dimensional potential vector* $\mathbf{a}(x, y, z)$ *whose second partial derivatives are continuous,*

$$\nabla \cdot (\nabla \times \mathbf{a}) = 0. \tag{C.1}$$

Proof. The proof is quite straightforward :

$$\begin{aligned}\nabla \cdot (\nabla \times \mathbf{a}(x, y, z)) &= \nabla \cdot \left[\left(\frac{\partial a_z}{\partial y} - \frac{\partial a_y}{\partial z}\right)\hat{\mathbf{e}}_x + \left(\frac{\partial a_x}{\partial z} - \frac{\partial a_z}{\partial x}\right)\hat{\mathbf{e}}_y + \left(\frac{\partial a_y}{\partial x} - \frac{\partial a_x}{\partial y}\right)\hat{\mathbf{e}}_z\right] \\ &= \frac{\partial^2 a_z}{\partial x \partial y} - \frac{\partial^2 a_y}{\partial x \partial z} + \frac{\partial^2 a_x}{\partial y \partial z} - \frac{\partial^2 a_z}{\partial y \partial x} + \frac{\partial^2 a_y}{\partial z \partial x} - \frac{\partial^2 a_x}{\partial z \partial y} \\ &= 0.\end{aligned}$$

The assumption on the continuity of the second partial derivatives allows to use *Schwarz's theorem* which states that the second partial derivatives are symmetric. □

This property suggests a simple method to build a divergence-free velocity field, namely one that satisfies the continuity equation (6.2). Indeed, it is sufficient to choose a relevant three-dimensional vector $\mathbf{a}(x, y, z) = (a_x(x, y, z),\ a_y(x, y, z),\ a_z(x, y, z))$ with a_x, a_y and a_z of class $\mathcal{C}^2$ and impose that

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = -\nabla \times \mathbf{a} = -\begin{pmatrix} \frac{\partial a_z}{\partial y} - \frac{\partial a_y}{\partial z} \\ \frac{\partial a_x}{\partial z} - \frac{\partial a_z}{\partial x} \\ \frac{\partial a_y}{\partial x} - \frac{\partial a_x}{\partial y} \end{pmatrix}. \tag{C.2}$$

This ensures that $\mathbf{u}$ is divergence-free since $\nabla \cdot \mathbf{u} = -\nabla \cdot (\nabla \times \mathbf{a}) = 0$.

In this work, we restrict ourselves to two-dimensional flows in the plane. Let us consider the plane $(\hat{\mathbf{e}}_y, \hat{\mathbf{e}}_z)$ without loss of generality. In that case, $u = 0$ and $\partial \cdot / \partial x = 0$. The relation (C.2) becomes

$$\mathbf{u} = \begin{pmatrix} 0 \\ v \\ w \end{pmatrix} = -\nabla \times \mathbf{a} = \begin{pmatrix} 0 \\ -\frac{\partial a_x}{\partial z} \\ \frac{\partial a_x}{\partial y} \end{pmatrix}, \tag{C.3}$$

where only the component a_x is needed to describe $\mathbf{u}$. Hence, the velocity field of a flow in the plane is described by a scalar quantity, the so-called *streamfunction*, generally noted ψ. The potential vector $\mathbf{a}$ is thus of the form $\mathbf{a}(y,z) = (\psi(y,z),\, 0,\, 0)$, and the components of the velocity vector are given by :

$$v = -\frac{\partial \psi}{\partial z}, \quad w = \frac{\partial \psi}{\partial y}. \tag{C.4}$$

Note that adding any constant to ψ leaves the velocity vector unchanged. This adds some freedom to the choice of ψ. In order to derive a streamfunction that is relevant to some particular problem, it is of top-notch interest to get some physical intuition about the streamfunction. To this end, two fundamental properties of the streamfunction in two dimensions are shown in the next paragraphs.

First, notice that assuming that $\psi \in \mathcal{C}^2$ implies straightforwardly that $\mathrm{d}\psi = (\partial\psi/\partial y)\mathrm{d}y + (\partial\psi/\partial z)\mathrm{d}z$ is an *exact differential* since by Schwarz's theorem

$$\frac{\partial^2 \psi}{\partial y \partial z} = \frac{\partial^2 \psi}{\partial z \partial y}. \tag{C.5}$$

Thus,

$$\int_{\mathbf{x}_1}^{\mathbf{x}_2} \mathrm{d}\psi = \psi(y_2, z_2) - \psi(y_1, z_1) \tag{C.6}$$

is path-independent.

Property C.2. *In two dimensions, the curves along which ψ is constant (i.e. the isolines of ψ) are exactly the streamlines of the flow, namely the family of curves that are instantaneously tangent to the velocity vector.*

Proof. Let an isoline of ψ be parametrized by $s \mapsto \mathbf{x}_S(s) = (y_S(s), z_S(s))$. The fact that ψ is constant along that curve implies that $\mathrm{d}\psi_S = (\partial\psi/\partial y)\mathrm{d}y_S + (\partial\psi/\partial z)\mathrm{d}z_S = \nabla\psi \cdot \mathrm{d}\mathbf{x}_S = 0$. This shows that the vector $\nabla\psi$ is normal to the curve $\mathbf{x}_S(s)$. Hence, showing that $\mathbf{x}_S(s)$ is everywhere tangent to $\mathbf{u}$ is equivalent to showing that $\mathbf{u} \cdot \nabla\psi = 0$ everywhere. The latter is straightforward using relation (C.4) :

$$\mathbf{u} \cdot \nabla\psi = -\frac{\partial \psi}{\partial z}\frac{\partial \psi}{\partial y} + \frac{\partial \psi}{\partial y}\frac{\partial \psi}{\partial z} = 0, \tag{C.7}$$

which concludes the proof. □

Property C.3. *In two dimensions, the volume flow rate between two streamlines of values ψ_1 and ψ_2 is equal to the difference of those streamlines, $\psi_1 - \psi_2$.*

Proof. Consider two infinitely close points $\mathbf{x}_1 = (y_1,\, z_1)$ and $\mathbf{x}_1 + \mathrm{d}\mathbf{x} = (y_1 + \mathrm{d}y,\, z_1 + \mathrm{d}z)$. At those points, the streamfunction has values $\psi(y_1, z_1) = \psi_1$ and $\psi(y_1 + \mathrm{d}y, z_1 + \mathrm{d}z) = \psi_1 + \mathrm{d}\psi$. Let us now consider the volume flow rate $\mathrm{d}q$ across the infinitesimal segment $[\mathbf{x}_1,\, \mathbf{x}_1 + \mathrm{d}\mathbf{x}]$, positive in the right-hand side direction of the segment if the latter is directed from $\mathbf{x}_1$ to $\mathbf{x}_1 + \mathrm{d}\mathbf{x}$. It is equal to $\mathbf{u} \cdot \hat{\mathbf{n}}$, where $\hat{\mathbf{n}} = (\mathrm{d}z,\, -\mathrm{d}y)$ is the unit vector normal to the segment, oriented in the right-hand side direction. Hence, $\mathrm{d}q = v\mathrm{d}z - w\mathrm{d}y$, which, using relation (C.4), amounts to

$$\mathrm{d}q = -\mathrm{d}\psi. \tag{C.8}$$

Now, consider any two points $\mathbf{x}_1 = (y_1, z_1)$ and $\mathbf{x}_2 = (y_2, z_2)$ in the (connected) domain. The volume flow rate $q_{1\to 2}$ across any curve $\gamma_{1\to 2}$ connecting $\mathbf{x}_1$ to $\mathbf{x}_2$, positive in the right-hand side direction of the directed segment $[\mathbf{x}_1,\, \mathbf{x}_2]$ is

$$q_{1\to 2} = \int_{\gamma_{1\to 2}} \mathrm{d}q = \int_{\mathbf{x}_1}^{\mathbf{x}_2} (-\mathrm{d}\psi) = \psi(y_1, z_1) - \psi(y_2, z_2), \tag{C.9}$$

where $\int_{\gamma_{1\to 2}}$ is the line integral along a curve connecting $\mathbf{x}_1$ to $\mathbf{x}_2$, afterwards noted $\int_{\mathbf{x}_1}^{\mathbf{x}_2}$ to emphasize the fact that it does not depend on the integration path, since $\mathrm{d}\psi$ is an exact differential. $\square$

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve www.uclouvain.be/epl