Large data reduction and structure comparison with topological data analysis

Maxime Soler

▶ **To cite this version:**

Maxime Soler. Large data reduction and structure comparison with topological data analysis. Image Processing [eess.IV]. Sorbonne Université, 2019. English. NNT : 2019SORUS364 . tel-02171190v3

HAL Id: tel-02171190

https://theses.hal.science/tel-02171190v3

Submitted on 16 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Spécialité

INFORMATIQUE

Présentée par

MAXIME SOLER

Pour obtenir le grade de

DOCTEUR DE SORBONNE UNIVERSITÉ

Réduction et comparaison de structures d'intérêt dans des jeux de données massifs par analyse topologique

Soutenance prévue le 20 juin 2019 devant le jury composé de :

M.	Georges-Pierre BONNEAU	Professeur, Grenoble Universités	**Rapporteur**
M.	Vijay NATARAJAN	Professeur, Indian Institute of Science	**Rapporteur**
M.	Holger THEISEL	Professeur, University of Magdeburg	**Examinateur**
M.	Bertrand MICHEL	Professeur, Ecole Centrale de Nantes	**Examinateur**
M.	Lionel LACASSAGNE	Professeur, Sorbonne Université	**Examinateur**
M.	Gilles DARCHE	Expert Simulation Réservoir, TOTAL S.A.	**Examinateur**
Mme	Mélanie PLAINCHAULT	TOTAL S.A.	**Co-encadrante**
M.	Julien TIERNY	HDR, CNRS, Sorbonne Université	**Directeur de thèse**

"The persistence question asks what it takes for something that is a person at one time to exist at another time as well. It asks what is necessary and sufficient for any past or future being, whether or not it is a person then, to be you or I. [...] Some general metaphysical views suggest that there is no unique right answer to the persistence question. The best-known example [...] says that for every period of time when you exist, short or long, there is a temporal part of you that exists only then. This gives us many likely candidates for being you—that is, many different beings now sitting there and thinking your thoughts."

Eric T. Olson [Ols10]

Acknowledgments

Merci Étienne, Pierre et Guilhem Du. ; vous m'avez motivé à m'engager dans cette thèse qui s'est révélée riche en rencontres et en péripéties. Merci Martin, Bruno, Gilles, Frédéric, pour m'avoir accueilli avec bienveillance au sein de vos équipes, dans RES mais aussi à la CIG, où j'avais déjà passé quelques années et auprès de laquelle j'avais beaucoup appris. Merci à toutes les personnes que j'ai rencontrées ou retrouvées au laboratoire, et plus particulièrement à Charles, Guillaume F. et Jules ; merci pour toute votre aide et pour la bonne ambiance.

Je tiens également à remercier H. Theisel, B. Michel, L. Lacassagne et G. Darche, qui ont aimablement accepté de faire partie de mon jury de thèse, et surtout les rapporteurs G.-P. Bonneau et V. Natarajan, pour leur retours et l'intérêt qu'ils ont manifesté pour mon travail. Je ne peux évidemment pas oublier mes encadrants, Mélanie et Julien T., qui m'ont accompagné et ont été disponibles à tous moments pendant ces trois ans ; merci de m'avoir coaché, de m'avoir fait progresser et maturer, dans les multiples sens du terme.

Merci les amis, 美樹 et ギエム, les mots me manquent pour exprimer à quel point votre soutien et votre enthousiasme m'ont été précieux. Éloi, Guillaume D., Julien V., Romain D., merci pour les aventures, les soirées, les instants de détente, les discussions passionnées, je compte bien qu'on ne s'en tienne pas là malgré la distance géographique qui nous sépare ; Pierrick, Sébastien, Gérald, François, merci pour les bons moments, les échanges, les découvertes ; Xavier D., Adrien, merci d'avoir comblé les silences quand ils devenaient trop pesants. Merci enfin à mes parents et ma famille, pour votre support et pour votre aide en toutes circonstances.

Je n'oublie pas tous ceux que je n'ai pas la place de citer ici mais qui ont eu un impact sur mon travail, mes idées ou mon regard ; nous prenons chaque saison davantage la couleur de ce qui nous traverse.

Publications

International Publications

Conferences

- **Maxime Soler**, Mélanie Plainchault, Bruno Conche, Julien Tierny,
  "Topologically Controlled Lossy Compression",
  IEEE Pacific Conference on Visualization, Kobe, Japan, pp. 46-55, 2018

- **Maxime Soler**, Mélanie Plainchault, Bruno Conche, Julien Tierny,
  "Lifted Wasserstein Matcher for Fast and Robust Topology Tracking",
  IEEE Symposium on Large Data Analysis and Visualization, Berlin, Germany, 2018,
  Best paper honorable mention award

Submitted to Journals

- **Maxime Soler**, Mélanie Plainchault, Martin Petitfrère, Gilles Darche, Bruno Conche, Julien Tierny,
  "Ranking Viscous Finger Simulations to an Acquired Ground Truth with Topology-aware Matchings",
  2019

Other

Tutorials

- Guillaume Favelier, Charles Gueunet, Attila Gyulassy, Julien Jomier, Joshua Levine, Jonas Lukasczyk, Daisuke Sakurai, **Maxime Soler**, Julien Tierny, Will Usher, Qi Wu,
  "Topological Data Analysis Made Easy with the Topology ToolKit",
  IEEE VIS Tutorial, 2018

Abstract-only national workshops

- **Maxime Soler**, Mélanie Plainchault, Bruno Conche, Julien Tierny,
  "Topologically lossy L_∞ compression",
  MATHIAS 2017, Marne-la-Vallée, France

- **Maxime Soler**, Mélanie Plainchault, Bruno Conche, Julien Tierny,
  "Compression avec perte contrôlée par la topologie",
  Journée Visu 2018, Palaiseau, France

- **Maxime Soler**, Mélanie Plainchault, Bruno Conche, Julien Tierny,
  "Suivi topologique rapide et robuste par appariement de Wasserstein augmenté",
  Journée Visu 2019, Paris, France

Contents

Notations

Topological Data Analysis

$\emptyset$	empty set
$\mathbb{X}$	set
$\mathbb{N}$	set of non-negative integers
$\mathbb{Z}$	set of integers
$\{0,1\}$	set of integers modulo 2
$(\mathbb{X}, T)$	topological space
$\mathbb{M}$	topological manifold
$\mathbb{R}$	set of real numbers
$\mathbb{R}^d$	d-dimensional Euclidean space
$]a,b[$, $[a,b]$	open (resp. closed) interval of $\mathbb{R}$
PL	piecewise-linear
$\mathcal{K}$	simplicial complex
$\mathcal{T}$	triangulation
$\mathcal{M}$	PL d-manifold
$f : \mathcal{M} \to \mathbb{R}$	PL scalar field
$St(v)$	star of a vertex v
$Lk(v)$	link of a vertex v
$Lk^-(v)$, $Lk^+(v)$	lower link, upper link of a vertex v
$f^{-1}(i)$	level-set of an isovalue i
$f^{-1}_{-\infty}(i)$, $f^{-1}_{+\infty}(i)$	sub-level set, sur-level set of an isovalue i
$\|\|f-g\|\|_p$	p-norm
β_p	p^{th} Betti number
$\mathcal{I}$	critical point index
$\mathcal{D}(f)$	persistence diagram of f
W_∞	bottleneck distance
W_2	2-Wasserstein distance
$\widehat{W}_2$	geometrically lifted 2-Wasserstein distance
$\mathcal{F}$	thresholded domain

Reservoir Simulation

$i \in \{o, w\}$	oil or water phase
v_i	velocity of phase i
$\nabla \cdot$	divergence
$\overrightarrow{\nabla}$	gradient
ϕ	porosity
ρ_i	mass density of phase i
V_{tot}	total volume
m_i	mass of phase i
S_i	saturation of phase i
P_i	pressure of phase i
$\mathbf{K}$	absolute permeability tensor
kr_i	relative permeability of phase i
$\mathbf{g}$	acceleration of gravity
q_i	injection/production of phase i
P_c	capillary pressure
S_{or}	residual oil saturation
S_{wc}	connate water saturation
kr_w^0	water relative permeability endpoint
n_c, n_w	power law exponents

1 Introduction

Recent advances in computer science and data analysis powered by modern hardware have brought at reach many intricate and inaccessible problems in various scientific domains. Complex systems and phenomena in nature are often associated with *theoretical models*, based on mathematical descriptions. By combining these models with actual measurements, computer simulations can produce forecasts, in the form of large chunks of scientific data. This enables to check the reliability of such models, to study the properties and behavior of the associated systems, and to predict their probable evolutions.

Computer simulations have become an essential tool for studying natural phenomena in physics, chemistry, biology, as well as human systems in social sciences and economy. In particular, simulation is central to the field of geosciences, notably in the oil and gas industry, which is the applicative background of this academic-industrial Ph.D. thesis (CIFRE). In this context, it is used to understand the behavior of subsurface fluid flows. Since computing power kept on increasing in the past few decades, and more precise tools were available to produce accurate measurements, more data became available from scientific domains making use of numerical models. One of the last ground-breaking examples is the reconstruction of a black hole image from petabytes of data [Cas19; Aki+19].

This fast technological growth has brought many challenges in recent years as scientists are faced with constantly increasing volumes of data that can be very difficult to analyze. Topological data analysis (TDA) techniques [EH09] have proven their interest in this perspective, because they allow to capture meaningful structures in scalar data, as *topological features*.

In a large data context, however, a fundamental problem remains the limitation of computing infrastructures, with regard to input/output (I/O) capabilities. Typically, the black hole image reconstruction mentioned above [Cas19; Aki+19] required experts to ship hard drives by

plane by lack of faster data transfer solutions. This is particularly problematic for simulation data-sets which model time-dependent phenomena, since a high temporal resolution means a high I/O burden; even more so with parametric studies which perform many simulations simultaneously. There is a subsequent need to minimize data movement, for instance by performing compression, or by extracting minimal structures of interest in the data (*e.g.* with topological features).

In this thesis, we propose to model structures of interest in scientific scalar data with such topological features. First, we address data growth problematics thanks to a new feature-oriented compression algorithm. Then, we propose to address data analysis problematics thanks to a new feature tracking algorithm. Finally, we propose a way to assess the quality of the features produced by simulations compared to a ground truth, with a ranking framework which we apply in an industrial case study, using the *in-situ* paradigm to answer the data movement problematic.

1.1 Challenges

As scientific data produced by simulations or acquisition instruments grow in size and intricacy, new ways for exploration and visualization are needed, in order to extract meaningful knowledge. In particular, the problem of extracting, representing and measuring structures of interest in large simulations subject to infrastructure limitations is central to this thesis. As this work is inscribed in an academic-industrial partnership with a major actor of the oil and gas industry, our primary applicative context will be the simulation of fluid flows in porous media, for which specialists have to deal with such problematics.

The identification and interpretation of structures in simulations data, which quite often take the form of scalar (*i.e.* real-valued) data defined on some geometrical domain, gives rise to multiple challenges. We propose to study scalar data in a unified framework, based on topology, that enables us to tackle four of these scientific challenges, described in the following.

Structure extraction and characterization

When confronted with scalar data, the first step in a scientific analysis pipeline is to identify which subset of the data is of actual interest. In this context, topological data analysis (TDA) techniques have been extensively used to perform structure extraction, in a generic, robust and efficient way. The translation of meaningful regions of interest into topolog-

ical terms, however, greatly varies depending on the applicative domain. In geosciences and in particular reservoir simulation, this remains to be explored.

Increasing size and complexity of data

Extracting features of interest can become problematic when data volumes get larger, as both regular analysis methods and computing infrastructures begin to show limitations. One of the advantages of TDA techniques is their ability to define features in a hierarchical way, at multiple "levels of detail". Even though, the analysis of very large data becomes prohibitive due to hardware input/output (I/O) limitations: computing power grows faster than storage transfer rates. This calls for new ways to reduce data movement.

Understanding time-varying data

The I/O problematic is particularly important for *time-varying* data (for example coming from simulations), where a simple increase in time resolution may prohibit analyses which require the full data to be stored to disk first. Moreover, traditional tools show some limitations for the analysis of time data (unless the time sampling is very high, which would lead to I/O issues). For example, when trying to follow structures in a given simulation at successive time-steps, there may be robustness issues producing discontinuous jumps in the tracking graph.

Understanding the influence of simulation model parameters

Parametric studies aim at understanding the influence of simulation model parameters on the produced forecasts, often in order to adjust these parameters. If going from static to time-varying scalar data can be seen as adding a dimension of complexity, doing parametric studies can be seen as adding even more dimensions of complexity; such studies also need to be adapted to data movement problematics. In addition, a framework for studying ensembles of simulations using TDA methods need to account for the problems raised by time-varying data, and also for domain-specific notions when working with a topological definition of features of interest.

1.2 Contributions

In this thesis, we propose contributions for each of the aforementioned challenges.

Structure characterization and comparison with topological features

As a basis for our work, we rely on the theoretical setting of TDA, which allows to define structures of interest in scalar data, in terms of *topological features*, in a robust and hierarchical way. In particular, an advantageous aspect of TDA is its ability to measure the similarity between features across distinct data-sets, based on identifying the underlying discrete topological constructs. Such similarity measures are called *topological metrics*, and are central for data-set comparisons in our context. Our feature-oriented characterization of structures of interest in the applicative domain of reservoir simulation is documented in chapter 6.

Data compression which preserves topological features

Considering that topological features reliably capture structures of interest in static data, we introduce a new lossy compression algorithm, with guarantees on the topological loss, in an effort to address the I/O problematic. The loss can be controlled with respect to topological metrics thanks to a user-defined threshold. This allows to reduce the size of scientific data, achieving high compression factors in practice, while preserving the most salient topological structures of interest. The approach is extended to optionally enforce a maximum error bound specified by the user, and we show how it can be used in conjunction with other state-of-the-art compressors. This contribution has been documented in the publication [Sol+18b].

Fast tracking of time-varying features

In the context of time-varying data, we propose a new feature tracking framework that enables a fast tracking of structures throughout time. The approach is designed to be robust with respect to both noise and temporal resolution, which allows in practice to consider fewer time-steps and which consequently contributes to relaxing the IO usage. For that purpose, we introduce new topological metrics, as well as a new efficient algorithm to compute them, adapting and adjusting existing TDA metrics which suffered from robustness limitations. The tracking, based on the same topological definition of features, can be performed on data that was reduced with our compression method. This contribution has been documented in the publication [Sol+18a].

Topology-guided industrial parametric case study

Relying on the concepts that we introduced in the context of time-varying

data, we adapt and extend topological metrics to the needs of an industrial parametric case study involving reservoir simulation, an essential application in the oil and gas industry. Specifically, we introduce a new ranking framework which evaluates the likeliness of simulation runs in an ensemble given a reference ground truth. Our approach is designed to allow specialists to quickly determine which are the most physically adequate parameters in their simulations; its relevance is assessed with feedback from domain experts. With this case study modeling the *viscous fingering* phenomenon arising in reservoir simulation, we demonstrate how topological features can be used to capture structures of interest, and adapted to the case of large data, validating our initial motivation. In order to perform in the context of very large data and to handle I/O problematics, we deployed our analysis pipeline following the *in-situ* paradigm, a recent technique addressing such infrastructure limitations. This contribution is documented in the submitted manuscript [Sol+19].

1.3 Outline

The remainder of this manuscript is organized as follows.

In chapter 2, we introduce the scientific context of this thesis, at the crossroads between geoscience, scientific visualization and topological data analyis. In particular, we highlight the different problematics inherent to these scientific domains.

In chapter 3, we present the theoretical prerequisites regarding topology and topological data analysis, first in a very general, non-formal manner, then in a precise and formal way.

In chapter 4, we detail our lossy compression scheme for scientific data-sets, which allows the user to control the topological loss.

In chapter 5, we describe our novel feature tracking framework, designed to detect and follow topological singularities in time-varying scientific data.

In chapter 6, after having introduced some key specifics of reservoir simulation, we expose our framework for ranking simulation runs to an acquired ground truth in a *viscous fingering* parametric study.

Finally, in chapter 7, we summarize the contributions brought by this thesis and expose some open perspectives.

Scientific Context

2

Contents

In this chapter, we expose the main scientific context of our work: geoscience and reservoir simulation, and highlight some problematics inherent to this domain. We then present an overview of modern scientific data analysis and visualization; in particular, we introduce techniques related to Topological Data Analysis and *in-situ*, which were specifically designed to address such issues. This gives a general context to the tools and techniques that will be used throughout this manuscript.

2.1 Geoscience

The field of geoscience includes many scientific domains aiming at understanding phenomena and mechanisms related to the planet Earth. For instance, climatology and meteorology are interested in measuring, modeling and predicting phenomena related to the atmosphere; where geology is focused on studying the lithosphere, the outermost solid rock shell of the Earth; and geophysics is concerned with physical phenomena such as vibrations studied by seismology, or the Earth's magnetic field studied by geomagnetism. Some of these scientific domains are of central importance to the oil and gas industry, which is the primary applicative focus of this thesis and the focus of the present section.

2.1.1 Oil and gas exploration

Subsurface hydrocarbons are commonly contained in porous reservoir rocks like sandstones or limestones. In the oil industry, the process of finding and exploiting natural hydrocarbon reservoirs is done in multiple steps. The first one, called exploration, aims to determining which geographical zones are the most likely to contain such reservoirs. Exploration can go through an analysis of aerial photographs, which can be followed by seismic surveys. On land, seismographs record the ground response to waves emitted by explosives or seismic vibration trucks; at sea, seismic vessels shot air guns and record the waves reflected by geologic strata below the seabed with series of hydrophones. The recorded signals must be processed and interpreted so as to determine the geological structure of the underground, and to detect the presence of hydrocarbons: this stage is called seismic interpretation. A geological 3D model of the petroleum field may then be constructed based on seismic data (Fig. 2.1).

When potential oil and/or gas reservoirs are detected, the available quantity of hydrocarbons must be assessed. Exploration wells are drilled in order to analyze more precisely the nature of geological strata. Probes are lowered into the well to record physical properties of the rock at different depths: electrical resistivity allows to distinguish between formations containing salty waters (good conductors) and those containing hydrocarbons (poor conductors); porosity gives a measure of the fraction of pore volume (where hydrocarbons can be found) in a rock volume; gamma radiation can be used to distinguish between sandstones (non-radioactive) and shales (containing clays with radioactive isotopes of

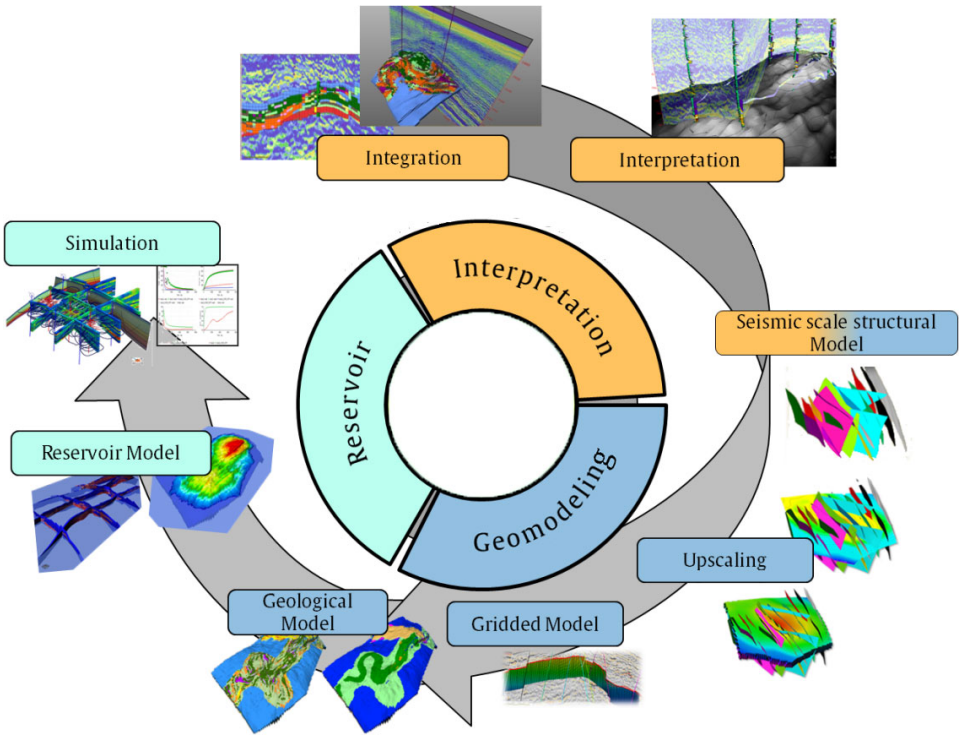

Figure 2.1 – *Modern workflow used in the oil and gas industry for the analysis of a hydrocarbon reservoir, relying on computer capabilities. During the interpretation stage, recorded seismic signals are transformed into images (integration), then processed and interpreted by experts to expose geological* horizons *and* faults*. During the geomodeling stage, a 3D model conforming to these geological features is built (structural model), associated with physical properties (upscaling and gridding), leading to a geological model. The reservoir model compatible with simulation software is built from the geological model; finally, simulations and history-match can be performed. Image from [SA14].*

potassium) [Daro5]; and many others. Core samples can also be brought to the surface to be analyzed in labs.

When multiple wells are drilled in the same geographical zone, then the physical properties recorded by logging instruments may be "extended" to fill the geological 3D model built from seismic data. *Gridded models* are then constructed: they consist of 3D meshes on which the physical equations modeling fluid displacements are discretized, serving as a basis for *simulation*. The simulation process aims to forecast and understand the dynamic behavior of subsurface fluids during the production stage. The evolution of reservoirs throughout time is recorded and compared to the predictions yielded by simulations, so as to adjust predictive models and data; this is a process called *history match*.

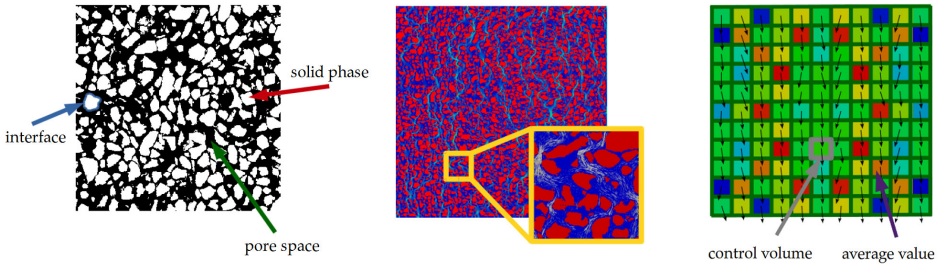

Figure 2.2 – *Two-dimensional cross-section of a sandstone (left), where the dark zones correspond to the porous space where hydrocarbons can be potentially found, and the light zone is the solid rock. The physics of flow can be captured by directly modeling the rock at pore-scale (center), or by averaging physical quantities over control volumes at the Darcy scale (right). Arrows represent velocity vectors and colors represent the volume fraction of the solid phase. Image from [Sou].*

2.1.2 Fluid simulation in porous media

Oil and gas experts are interested in understanding the displacement of fluids in porous media. These happen at multiple scales, from the pore scale (nanometers) to full field extents (tens of kilometers). There are multiple models for simulating flow in porous media, adapted to different scales. Fig. 2.2 shows two of the main modeling approaches.

As they are extremely computationally expensive and require to precisely know the pore network, pore-scale simulations cannot be run on the full scale of petroleum fields; they are typically used in order to understand flow in specific conditions. For example, PNM (pore network modeling) approaches [XBJ16] may be used to infer the parameters of Darcy-type simulations [Lou+18]. The latter methods, which work with average quantities defined on control volumes, are commonly used for larger scale simulations.

Among the physical quantities that are modeled in order to describe the evolution of a reservoir, two of the most important ones are pressure and saturation, which is the volume fraction of a certain phase (say the oil phase) at a given control volume in the field. The precise equations at play in Darcy flow are discussed further in chapter 6.

There are numerous simulation schemes for flow in porous media, which can be based on an Eulerian or a Lagrangian formulation [BB67], sometimes both. For example, the method illustrated in Fig. 2.3 first uses static properties, well locations and initial conditions (Fig. 2.3, left) to solve the pressure field (Fig. 2.3, center left); then it uses this pressure field to trace streamlines (Fig. 2.3, center right); finally, it numerically solves the

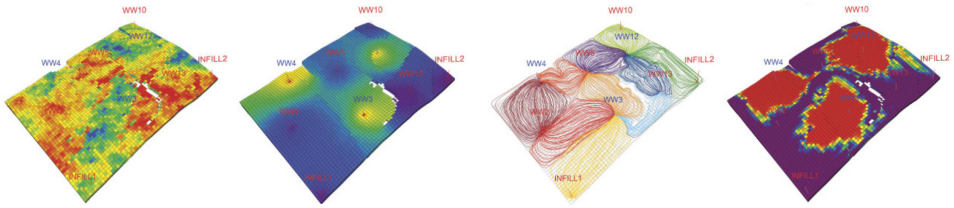

Figure 2.3 – *Successive simulation steps of a porous flow simulation solver for polymer floods. Initial conditions and properties at wells (left) are used to compute the pressure scalar field (center left); which is then used to compute the fluid velocity vector field and trace streamlines (center right). Transport equations are then solved numerically on each streamline and their solutions are reported on the static grid, yielding the saturation scalar field (right). The next time-step can then be computed. Scalar field values range from blue (low) to red (high). Image from [Thi+10].*

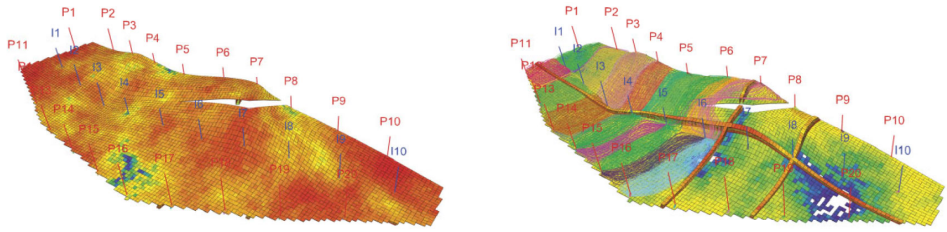

Figure 2.4 – *3D model of an oil reservoir from [Pet+10]; top layer view (left), cross-sections with streamlines colored by regions (right). Producer wells are colored in red; injector wells are colored in blue. The displayed scalar field is the permeability in the x-direction, (blue: low values to red: high values). Image from [Thi+10].*

1D transport equations numerically on the streamlines and maps them back to the Eulerian grid (Fig. 2.3, right).

Some of the most important outputs of these simulations are hydrocarbon production forecasts at wells. Other outputs, which are not necessarily saved nor fully exploited, are the scalar fields used as intermediate steps to compute production data. This is the case of the pressure and saturation scalar fields, for instance, which are defined at each cell of the gridded simulation model.

2.1.3 Inherent challenges

As mentioned in Sec. 2.2, both computing power and acquisition techniques have quickly evolved in recent years, in such a way that real world studies involving porous flow simulations kept growing in size and resolution. Typical studies involving oil and gas fields with dimensions in the order of tens of square kilometers can require large meshes displaying a complex geometry. For example, a model from the Brugge Benchmark Study [Pet+10] is shown in Fig. 2.4. The shown extent of this model only

has 60,048 cells; in recent studies, models with more than a billion of cells begin to appear. This gives rise to a number of challenges.

Data size and I/O

First, it is prohibitive to save scalar fields to the disk, at multiple time-steps, for later analyses. The classical approach for data analysis in reservoir studies is, as a matter of fact, *post-mortem*: it requires to save simulation time-steps to the disk before anything. This is due to the historical design of the engineering workflow and to limitations in traditional analysis software. Reservoir engineers are thus constrained to select a drastically reduced subset of the data generated by the simulation for later analysis.

To address this issue, data reduction methods may be required, for example compression schemes. Other solutions for limiting data movement should be investigated, as ways to overcome the limitations of simulation infrastructures.

Interpretation and extraction of structures

An additional problem arising with huge volumes of data is the need for techniques to extract the right, meaningful information. Given, for instance, a micro CT scan of a core sample, which could be a very large 3D density scalar field, it is not obvious how to extract the pore network from this field. As another example, given a time-dependent scalar field representing the local ratio of oil and water in a core sample which is submitted to an injection of water, it is unclear how to characterize the appearance and evolution of probable instabilities (as in the well-known *viscous fingering* phenomenon).

Such scientific issues are, however, exclusive to neither reservoir simulation nor geoscience. As all fields of science progressively advanced, the difficult task of making sense of an ever-growing quantity of scientific data has lead to the development of scientific visualization and data analysis, in an effort to understand increasingly complex natural phenomena. In the following section, we expose and give context to the modern solutions that were developed in order to address such problematics.

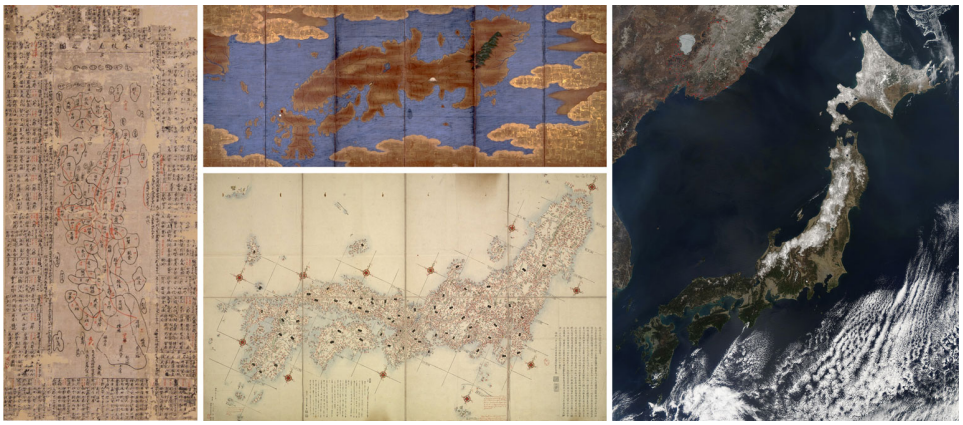

Figure 2.5 – *Maps of Japan: 14th century map from [His18] (left); ca. 1595 map from [Unn87b] (center, top); 1779 map from [Unn87a], the first to show latitude and longitude lines (center, bottom); 2011 satellite photograph from [Tea11].*

2.2 Visualization and data analysis

2.2.1 Computer science, 3D and visualization

Graphical representations have been used since the earliest ages to communicate ideas and knowledge. A reason for their efficiency as a means of capturing facts and insights about the surrounding world is the innate capability of humans to reason well with geometrical objects [Pin84]. In fact, the human brain performs remarkably when dealing with visual objects and environments, for instance for recognizing and associating shapes or reasoning about the possible interactions with a physical system. An area of study concerned with this part of human intelligence is visual cognition [Pin84]. Relying on these abilities, the idea behind scientific visualization is to enhance the understanding of abstract scientific data through the use of (possibly interactive) graphical or sensory representations.

Scientific visualization is arguably at least as old as cartography. Over the course of centuries, maps have become more and more accurate as surveying tools and geometrical methods were developed (Fig. 2.5). With the systematic use of triangulation techniques starting in the sixteenth century, and the advent of the telescope, navigational instruments and printing, quite precise maps could be obtained, long before more "direct" methods such as aerial photography were a possibility. In modern cartography, a number of map projection techniques deliberately distort space to emphasize certain aspects of the data. This is the case of the Mercator projection, which makes all lines of constant bearing (called *loxodromes*) into straight lines.

The idea of gaining visual insight into abstract data is not limited,

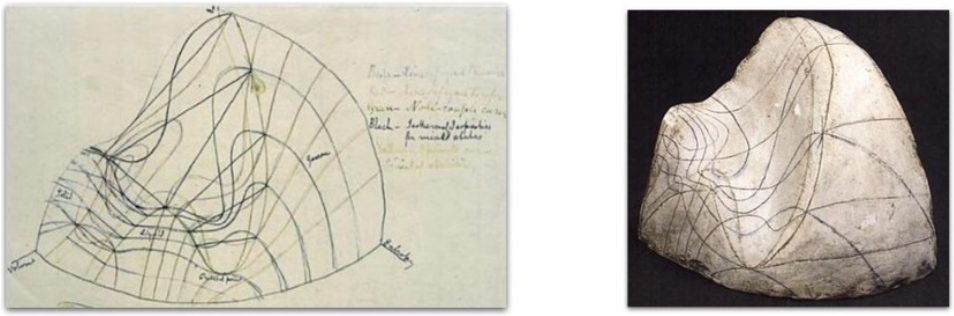

Figure 2.6 – *Maxwell's sketch of Gibbs' thermodynamic surface (left) from [Gib73], photograph of Maxwell's clay model (right) from [Max90]. According to Maxwell, this model, displaying the possible states of a water-like substance in a volume-entropy-energy 3D space, allows to represent the features of the substance on a convenient scale.*

though, to mapping the data into two-dimensional drawings or maps: historical examples include a notable sculpture of a three-dimensional surface by Maxwell in 1874 (Fig. 2.6). This would inspire later visualizations, made possible by computer graphics.

In the first half of the twentieth century, advances in the electrical sciences and engineering permitted the invention of new *interactive* visualization instruments, such as cathode-ray tubes (CRTs) in the 1950s, which were used in oscilloscopes, televisions and computer monitors. CRTs may produce images line by line on a screen by magnetically bending a focused electron beam, in a process called *raster scan*. As advances in computing led to the emergence of interactive graphical systems, the new capabilities provided by computers drew interest from the aerospace, automotive and energy industries in the early 1960s.

The late 1960s and 1970s witnessed foundational work in graphics and a growing interest of the animation and entertainment industries. Major breakthroughs notably include hidden-surface algorithms [SSS74], Gouraud [Gou71] and Phong [Pho75] shading models (see Fig. 2.7), texture [Cat74] and bump [Bli78] mapping. Though these (mostly *ad-hoc*) interactive techniques essentially concerned raster-based graphics because of hardware constraints at that time, ways to produce physically accurate images by formulating [Kaj86] and solving the global illumination problem were already investigated, which prefigures modern photo-realistic rendering (Fig. 2.7, bottom right), as well as volume rendering. In the latter, light rays are cast into volumetric data; these ray interact with matter according to a user-defined function (called a *transfer function*), allowing to visually explore large and occluded data.

In the 1980s and later decades, computer technology became available

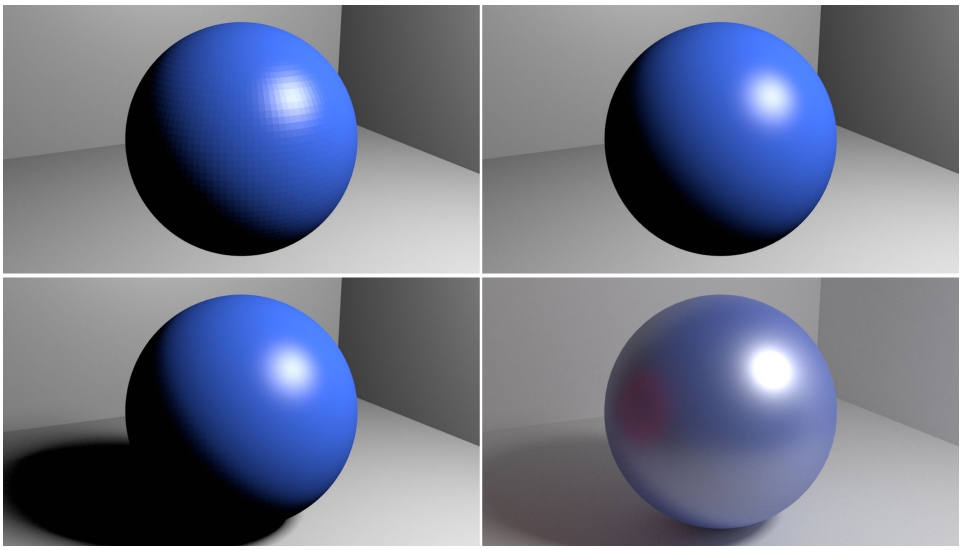

Figure 2.7 – *Computer-generated images of a sphere. Lighting can be efficiently computed with ad-hoc raster-based methods: flat shading (top left), Phong shading (top right). As the physical light interacts between objects, ray-tracing methods [Whi05] can be used to compute shadows cast by the light source (bottom left). As light bounces many times over different objects, modern path-tracing methods can be used to render physically plausible images (bottom right, obtained with Blender's Cycles renderer [Fou15]).*

to the larger public with the proliferation of home computers. The fast growth of visualization and graphical techniques, with the help of Moore's law until the very recent years, gave rise to many ways of efficiently representing abstract data, in numerous scientific domains and industries. In parallel, scientific data equally kept growing in size and intricacy, thanks to the advancement in acquisition and simulation techniques. New methods were needed for dealing with very large data, for example for extracting features from massive acquisitions (CT scans, astrophysical data, seismic acquisitions, etc.) or for assessing the influence of parameters and uncertainties in chaotic systems (in meteorology, parametric studies, etc.).

2.2.2 Topology and large data analysis

Topological data analysis (TDA) techniques [EH09; Pas+10; Hei+16; De +15] have been used over the course of recent years because of their ability to hierarchically identify features in scalar data in a generic, robust [ELZ02; CEH05] and efficient manner. They have been applied in various scientific domains, such as computational fluid dynamics [Kas+11; FGT16b], turbulent combustion [Bre+11], material sciences [Gyu+15], biological imaging [CSP04; Boc+18], chemistry [Bha+18; Gue+14], as-

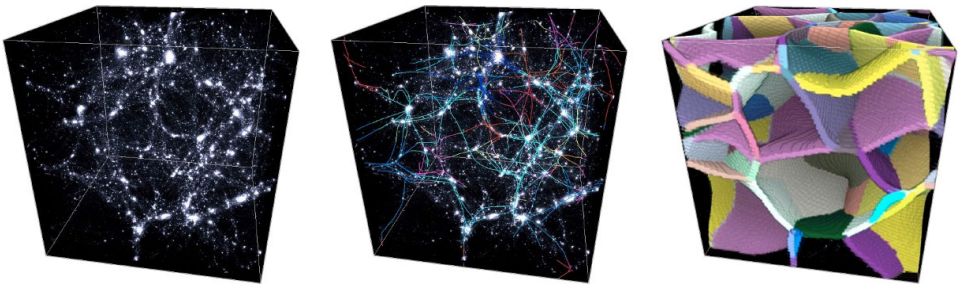

Figure 2.8 – *In a matter density data-set (left), the cosmic web can be extracted by querying the "most persistent* 1*-separatrices of the Morse-Smale complex" connected to local maxima of density (center). The "*2*-separatrices" of this topological object produce a segmentation of the space (right). Image from [Sou11].*

trophysics [Shi+16; Sou11], ensemble clustering [Fav+19], compression [Sol+18b] or feature tracking [Sol+18a].

One of the reasons for the successful applications of TDA is the possibility for experts to easily translate high-level domain-specific notions in terms of topological data structures, which are abstractions related to geometrical aspects or discrepancies in the data. Among such abstractions are persistence diagrams [EH08; ELZ02], contour trees [CSA03], Reeb graphs [Pas+07; Bia+08; Tie+09], Morse-Smale complexes [Gyu+08]. An example application in astrophysics is given in Fig. 2.8. Similar TDA applications can be found in the above examples.

Another possible application of TDA is the automatic definition of transfer functions for volume rendering [Wil17]. As a matter of fact, across different scientific domains, volumetric datasets may display a sensibly different distribution of features of interest. For large volumes of data, it is impractical to manually explore the parameter space of transfer functions to find the best possible representation. Fortunately, with TDA techniques, it is possible to automatically detect the most important regimes of features and adjust the transfer function to highlight them.

For a more detailed exposition of the underlying theoretical concepts of TDA, the reader is referred to chapter 3, which exposes in detail our formal setting.

2.2.3 Large-scale simulations and in-situ

As major industrial and academic actors display a clear ambition to reach toward exascale computing in the forthcoming years [Son+14; Com13], it is expected that the strong coupling between high performance computing and data analysis will have a significant impact over analysis and visual-

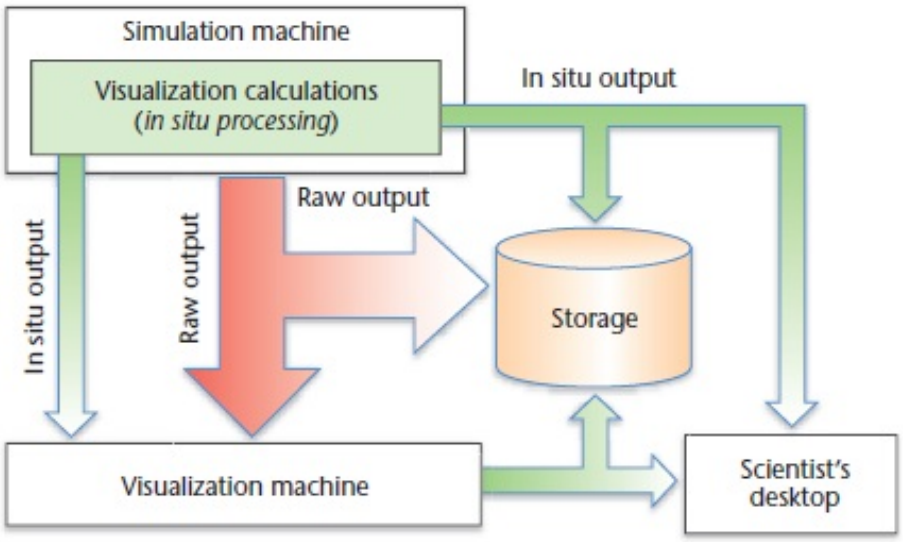

Figure 2.9 – *Data flow from simulation to visualization. Two approaches are possible for the analysis of data generated by numerical simulations:* post-mortem*, whose data flows, displayed in red, are potentially critical analysis bottlenecks if a consequent number of simulation time-steps must be saved; and* in-situ *(by extension,* in-transit*), whose data flows, displayed in green, are not an analysis bottleneck. Image from [SA16].*

ization algorithms, software infrastructures, hardware architectures and the workflow of research engineers.

For example, current trends in super-computing indicate an increase of the computing power that evolves faster than memory, IO and network bandwidth. Therefore, new paradigms for scientific simulation are needed. The simulation of flow in porous media, of key importance in the context of the work presented in this manuscript (for example for studying a phenomenon called *viscous fingering*), is particularly affected by data movement problematics, as models keep increasing in size, and high-resolution time sampling is required for producing realistic simulations.

Over recent years, solutions for limiting data movement were developed in this perspective, such as *in-situ* [Yu+10; Riv+12; Ras+11; OLe+16; Aya+16] and *in-transit* [Ben+12; Mor+11] models, described in this section.

In the *in-situ* approach, computing resources (HPC nodes) are used during the numerical simulation process to contribute to the analysis (or visualization) of the data being generated. The classical approach would require to perform the numerical simulation independently, then to save a certain number of simulated time-steps on the disk, then to perform visualization and analysis task on the saved data. This approach is called *post-mortem*.

A complementary notion is often associated to *in-situ*: *in-transit*. In this approach, the data computed using simulation nodes is transfered to computing nodes specialized in analysis and visualization tasks (hence not impacting initial simulation resources), without requiring to disk storage. Fig. 2.9 summarizes these two models.

Software infrastructures have been developed to enable in-situ. One such example is Paraview Catalyst [Aya+15; Bau+16], which was notably used for the visualization of a large-scale computational fluid dynamics simulation [Ras+14] (256,000 MPI processes on the Mira [Kum16] Blue Gene/Q supercomputer). Paraview Catalyst has also been used in the context of parametric studies and sensitivity analysis [LFR13; Ter+17].

We believe that this in-situ paradigm is of central importance to scientific studies involving large data, and that new analysis methods, for instance relying on TDA, should be developed while keeping in mind the technical challenges that a possible in-situ deployment would raise.

Theoretical background

3

Contents

This chapter introduces the concepts and modern formalisms of *topology* and *topological data analysis* (TDA), which are of central importance to this thesis. We first give some intuitive context to the versatility and convenience of topology-based tools. We then present a modern formalism of topology and proceed to introduce important TDA concepts, following and extending the elements of [Tie16] and [EH09]. The reader who is already familiar with these concepts may jump to the important definitions outlined in boxes (from Sec. 3.2 on), or directly to chapter 4 which exposes our first contributions and applications relying on TDA.

3.1 Introduction to topology

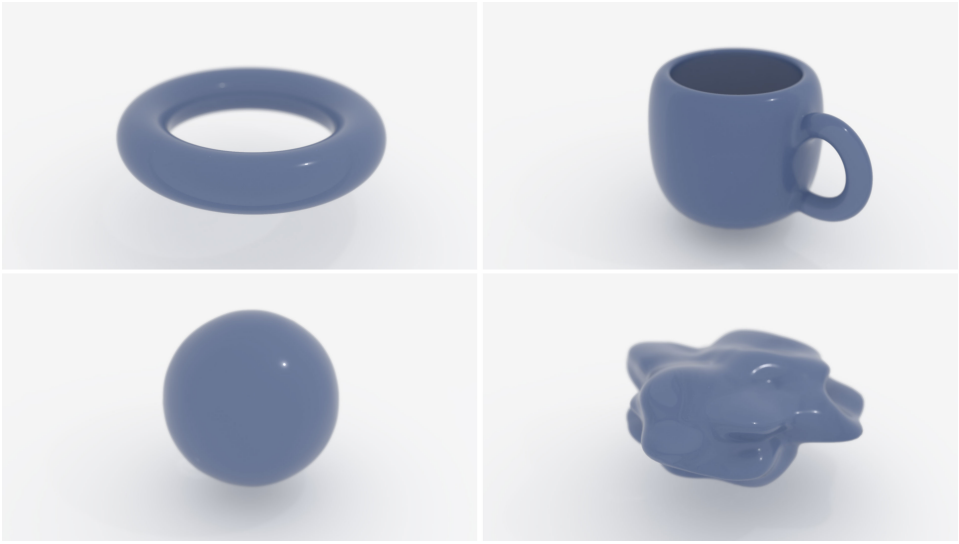

Figure 3.1 – *The surface of a doughnut (top left) can be "continuously" deformed until it resembles the surface of a mug (top right), whereas it cannot be "continuously" transformed into a two dimensional sphere (bottom left). Spaces which can be morphed into one another (top row, bottom row) are considered "equivalent".*

Extracting knowledge from large amounts of abstract data, whether of geometrical nature or not, is a difficult task. By contrast with historical scientific experiments and observations, which could simply involve notes on sheets of paper, modern science have to deal with such challenges. As underlined in chapter 2, it is common for scientists to try to visually represent their abstract data with shapes or drawings.

However, there is a certain independence between the shape of objects and some of their *structural* properties. For instance, the geometrical shape of a network can be modified without changing anything for an observer which would be within the network, and whose only observable universe would be the network itself. As the geometrical representation of objects is dependent on their embedding space, which can be distorted, and from the point of view of an external observer, it is sometimes difficult to capture structural discrepancies between them. The idea behind topology is to characterize such objects from a fundamental, structural point of view, without relying *a priori* on measurements or geometric representations.

A common image of topology in popular culture is that it states a mug is equivalent to a doughnut (Fig. 3.1), using the notion of *continuous* deformation. By continuous, it is intuitively meant that the shape is not allowed to undergo tearing or merging. In accordance to this idea, topology investigates the fundamental structural properties of objects, that can

be identified by looking at the said object regardless of any geometrical representation or measure. For that purpose, *homeomorphism* is a central concept, based on the notion of continuity, that allows to construct a topological characterization, so as to group objects, called *topological spaces*, in equivalence classes. Nonetheless, it proves quite difficult in practice to demonstrate that two objects are homeomorphic using only continuity. Topologists would rather make use of the concept of *invariants*, which are computable quantities or algebraic structures that stay the same for all objects that are homeomorphic to one another. It can be seen as a way to introduce back the concept of measure, in a more fundamental sense, to study topological objects.

The study of topology can bring out coarse truths about problems (not necessarily formulated in terms of geometry), and can lead to understand deep properties that a broad class of geometric or continuous objects can or cannot have. The interest of topology, though, is not limited to proving the existence or non-existence of solutions to abstract, theoretical problems. A recently developed field of topology involving *continuous functions*, called *Morse theory* [Bot88], popularized by Milnor [Mil63] and which found many applications in computing [EH09] proved very useful to the understanding of large data.

Topology can indeed provide insights about spaces as well as functions defined on spaces. In engineering and scientific applications, large volumes of data are often found in the form of a well-known (geometrical or topological) space, on which *scalar fields* or *vector fields* are defined. The question often asked is to analyze the properties of such fields.

Interestingly, there are deep relations between functions and the space on which they are defined. The field of Morse theory which examines this relationship, relies on the central notions of *manifold*, *continuity*, *homeomorphism*, *critical points*, which are formally introduced in the following section (Sec. 3.2). In other words, if one knows some properties concerning a given function (obeying to certain conditions), but nothing about the space where it resides, then properties of this function can be used to study the "shape" of the space it is defined on. This may sound abstract, but in some cases, scientists may know properties concerning functions but not the space where they are defined. The study of robotic arms is a notorious example [TW84; Far08], where the configuration space of robot manipulators can be investigated with the help of Morse theory [Got88; Hau91], as presented in Fig. 3.2. In this domain, one of the challenges for engineers doing inverse kinematics is to deal with singularities, or *critical points*, of

the function giving the position of the end effector from the configuration space of the robotic arm [BHB84]. Such considerations have applications to the study of molecules [GK12]. In graphics, Morse theory has also interesting applications, for example to study implicit shapes [Har98; SH05; SKK91], or for visualization and mesh compression [LLT04].

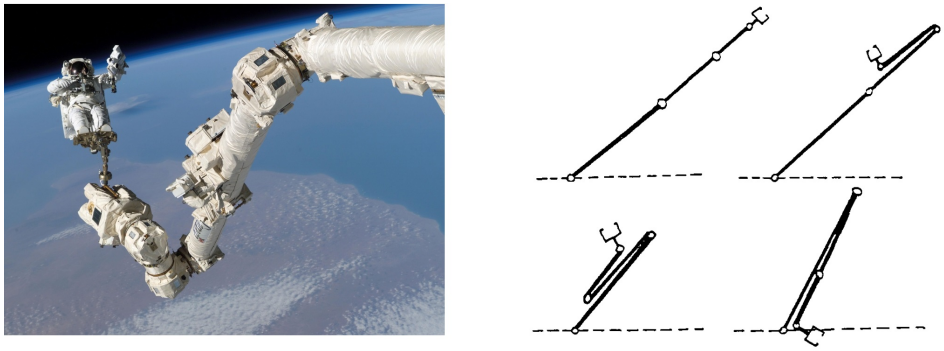

Figure 3.2 – *A robotic arm (left, Canadarm2 [Nok07] - NASA 2005). Some degenerate configurations of a 3-arm (right, from [BHB84]), corresponding to critical points of the function giving the position of the end effector from the angles of joints.*

The study of functions defined on known spaces (quite often on geometrical meshes) is, nonetheless, much more common. In recent years, topological data analysis, based on the same theoretical framework (Morse theory), has been broadly used because of its ability to perform feature extraction [ELZ02; CEH05] from functions defined on known spaces in a robust, hierarchical manner. A few of its numerous applications are briefly discussed in Sec. 2.2.2 of chapter 2.

In the following section, the topological notions illustrated here in an intuitive and non-formal manner are exposed in a more rigorous setting.

3.2 A formalism of topology

In this section, we introduce a modern formalism of topology. There is not only one valid formalism; the one presented in the following starts from the concepts of open sets, though other formalisms can build up from the concept of neighborhood. The reader can be referred to [EH09] and [DL14] for further readings on the subject.

3.2.1 Preliminary notions

Definition 3.1 *(Topological space)* Let $\mathbb{X}$ be a set. $(\mathbb{X}, T)$ is a topological space if T is a collection of subsets of $\mathbb{X}$ such that:

- $\emptyset$ and $\mathbb{X}$ belong to T;
- Any union of elements of T belongs to T;
- Any finite intersection of elements of T belongs to T.

Some reference books remark that the first condition requiring $\emptyset$ and $\mathbb{X}$ to be in T is redundant [Bou07]; this actually depends on the terminology used to define sub-collections of sets [Bou06]. We use standard definitions concerning sets and partitions.

Definition 3.2 *(Open set)* If $(\mathbb{X}, T)$ is a topological space, then elements of T are called open sets.

Similarly, elements of $\mathbb{X}$ are called *points*. For example, considering $\mathbb{R}$ the set of real numbers, and $\mathcal{B}$ the set of all open intervals of $\mathbb{R}$, $(\mathbb{R}, \mathcal{B})$ is *not* a topological space, because the union $]0,1[\cup]2,3[$ is not in $\mathcal{B}$. Instead, let $\mathcal{S}$ be the set containing all the elements of $\mathcal{B}$, and closed under the operations of arbitrary union and *finite* intersection. Then, $(\mathbb{R}, \mathcal{S})$ is a topological space and $]0,1[\in \mathcal{S}$ is an example of an open set. Open sets are used to build the concept of neighborhood:

Definition 3.3 *(Neighborhood)* Let $(\mathbb{X}, T)$ be a topological space and $x \in \mathbb{X}$. Then, $N \subset \mathbb{X}$ is a neighborhood of x if there is an open set in N that contains x.

Informally, a neighborhood of a point x is a set that contains all elements that are "arbitrarily" close to x; any set that contains a neighborhood of x is itself a neighborhood of x. An open set can be seen as a set which is a neighborhood of all of its points. In this sense, the union of any family of open sets is also open.

The next definitions introduce the notion of *homeomorphism* between topological spaces.

Definition 3.4 *(Function)* Let $(\mathbb{X}_1, T_1)$ and $(\mathbb{X}_2, T_2)$ be two topological spaces. A function $f : \mathbb{X}_1 \to \mathbb{X}_2$ associates each element of $\mathbb{X}_1$ with a unique element of $\mathbb{X}_2$.

Definition 3.5 *(Bijection)* A function $f : \mathbb{X}_1 \to \mathbb{X}_2$ is a bijection if for each element $x_2 \in \mathbb{X}_2$ there is a unique element $x_1 \in \mathbb{X}_1$ such that $f(x_1) = x_2$.

Definition 3.6 *(Injection)* A function $f : \mathbb{X}_1 \to \mathbb{X}_2$ is an injection if for each element $x_2 \in \mathbb{X}_2$ there is at most one element $x_1 \in \mathbb{X}_1$ such that $f(x_1) = x_2$.

Definition 3.7 *(Continuous function)* Let $(\mathbb{X}_1, T_1)$ and $(\mathbb{X}_2, T_2)$ be two topological spaces. The function $f : \mathbb{X}_1 \to \mathbb{X}_2$ is continuous if for each open set $t_2 \in T_2$, $f^{-1}(t_2)$ is an open set of T_1.

Definition 3.8 *(Homeomorphism)* Let $(\mathbb{X}_1, T_1)$ and $(\mathbb{X}_2, T_2)$ be two topological spaces. The function $f : \mathbb{X}_1 \to \mathbb{X}_2$ is an homeomorphism if f is a bijection and f and f^{-1} are continuous. The topological spaces are said homeomorphic.

In general topology, homeomorphisms are used to characterize the structure of topological spaces: two spaces are considered equivalent if they are homeomorphic. Quite often in the domain of scientific visualization, the data is defined on some geometrical domain. The next definitions introduce the concept of *topological manifold*, adapted in this case.

Definition 3.9 *(Unit Euclidean ball)* The unit Euclidean ball of dimension d is the set $\mathcal{B}^d = \{x \in \mathbb{R}^d, ||x||_2 < 1\}$, where $||\cdot||_2$ denotes the Euclidean (L^2) norm.

Definition 3.10 *(Unit Euclidean half-ball)* The unit Euclidean half-ball of dimension d is the set $\mathcal{B}^d_{1/2} = \{x = (x_1, \ldots, x_d) \in \mathbb{R}^d, ||x||_2 < 1 \text{ and } x_1 \geq 0\}$.

Definition 3.11 *(Manifold)* A topological space $\mathbb{M}$ is a d-manifold if every point of $\mathbb{M}$ has an open neighborhood homeomorphic to either the unit Euclidean ball of dimension d or the unit Euclidean half-ball of dimension d.

Definition 3.12 *(Manifold boundary)* Let $\mathbb{M}$ be a d-manifold. The set of points of $\mathbb{M}$ which have a neighborhood homeomorphic to $\mathcal{B}^d_{1/2}$ is called the *boundary* of $\mathbb{M}$.

Manifolds without a boundary are called *closed*. Fig. 3.3 illustrates this distinction. A d-manifold can be seen as a curved space, which is locally equivalent to the Euclidean space of dimension d (except on its boundary) but with a possibly more complicated global structure.

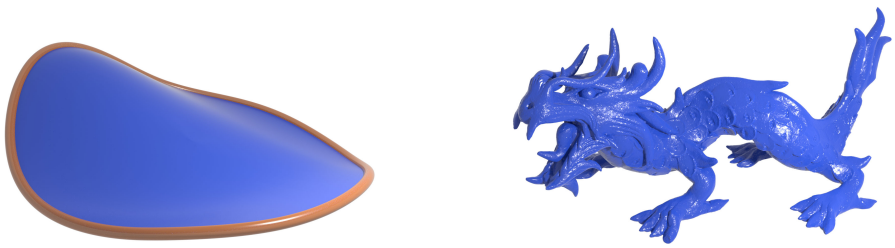

Figure 3.3 – *Two 2-manifolds (surfaces), with boundary (left, the boundary is colored in orange), and without boundary (right).*

3.2.2 Domain representation

The main focus of this manuscript is the analysis of scalar fields, which are one-dimensional scientific data defined on a geometrical domain. In this subsection, we formally introduce the domain representation that was chose in the context of this work.

Definition 3.13 *(Convex set)* A set $\mathbb{C}$ of a d-dimensional Euclidean space $\mathbb{R}^d$ is convex if, for any pair of points $x, y \in \mathbb{C}$, the point $tx + (1-t)y$ is also in $\mathbb{C}$, for all $t \in [0, 1]$.

In other words, a set is convex if all pairs of points in the set define a straight line segment which is also in the set (Fig. 3.4).

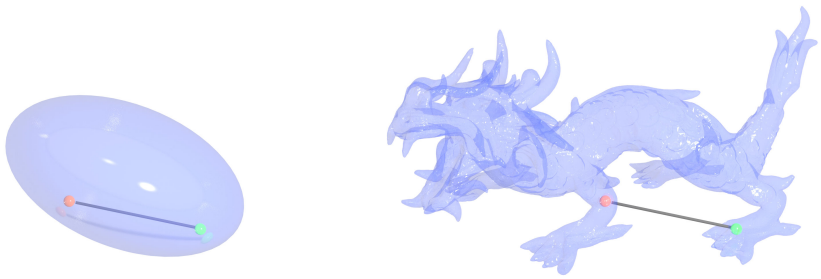

Figure 3.4 – *Two 3-manifolds (volumes), a convex one (left) where any two points can be linked by a segment that is still in the domain; and a non-convex one (right), for which it is not the case.*

Definition 3.14 *(Convex hull)* The convex hull of a set of points $\mathcal{P}$ of an Euclidean space $\mathbb{R}^n$ is the minimal convex set containing all points of $\mathcal{P}$.

Definition 3.15 *(Simplex)* A d-dimensional simplex is the convex hull of $d+1$ affinely independent points of an Euclidean space $\mathbb{R}^n$ (with $n \geq d$).

A simplex is the generalization of a triangle or a tetrahedron to any dimension. It is a basic combinatorial brick that we will be using to represent neighborhoods. Each simplex σ of dimension d contains $d+1$ simplices of dimension $d-1$ as illustrated in Fig. 3.5. The lower dimensional simplices of σ are called its *faces*.

Definition 3.16 *(Vertex)* A vertex is a 0-dimensional simplex.

Definition 3.17 *(Edge)* An edge is a 1-dimensional simplex.

Definition 3.18 *(Triangle)* A triangle is a 2-dimensional simplex.

Definition 3.19 *(Tetrahedron)* A tetrahedron is a 3-dimensional simplex.

Definition 3.20 *(Face)* A face τ of a d-dimensional simplex σ is a simplex containing a non-empty subset of the points of σ. An i-dimensional face is noted τ_i.

Definition 3.21 *(Simplicial complex)* A simplicial complex $\mathcal{K}$ in $\mathbb{R}^d$ is a set of simplices of $\mathbb{R}^d$ verifying the following two properties:

- for all $s \in \mathcal{K}$, every face of s is in $\mathcal{K}$;
- for all $s_1, s_2 \in \mathcal{K}$, the intersection $s_1 \cap s_2$ is either empty or it is face of both s_1 and s_2.

A simplicial complex is therefore a combinatorial assembly obtained by *gluing* simplices along their faces.

Definition 3.22 *(Star)* Let $\mathcal{K}$ be a simplicial complex. The star St of a simplex $\sigma \in \mathcal{K}$ is the set of simplices of $\mathcal{K}$ that contain σ as a face. The set of d-simplices of $St(\sigma)$ is noted $St_d(\sigma)$.

Definition 3.23 *(Link)* Let $\mathcal{K}$ be a simplicial complex. The link Lk of a simplex $\sigma \in \mathcal{K}$ is the set of faces of the simplices of $St(\sigma)$ that are disjoint from σ. The set of d-simplices of $Lk(\sigma)$ is noted $Lk_d(\sigma)$.

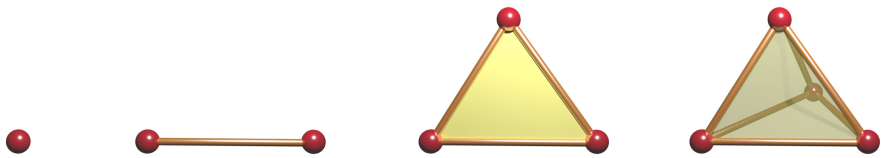

Figure 3.5 – *Some d-simplices, from left to right: vertex (red sphere, $d = 0$), edge (orange segment, $d = 1$), triangle (yellow surface, $d = 2$) and tetrahedron (translucent volume, $d = 3$).*

Definition 3.24 *(Triangulation)* Let $(\mathbb{X}, T)$ be a topological space and let $\mathcal{K}$ be a simplicial complex. Then $\mathcal{K}$ is a triangulation of $\mathbb{X}$ if there exists an homeomorphism between $\mathbb{X}$ and the union of all simplices of $\mathcal{K}$.

The set of topological data analysis techniques and algorithms that we will review in the later sections have been chosen to operate on triangulations. This is a practical basis as any usual mesh representation can indeed be transformed into a triangulation, by subdividing their cells into simplices. For regular grids (i.e. regular tessellations whose cells are quadrilaterals

in dimension 2 and cuboids in higher dimensions), an associated triangulation can be implicitly defined and adjacency relations can be retrieved on-the-fly [Tie+17]. Plus, the convexity of simplices make them easy to deal with when interpolating scalar fields, as we will see in the following subsection.

A triangulation can be efficiently represented in software, as a list of d-simplices with their stars and links for each dimension d. In the remainder of this work, we will represent the domain with triangulations of manifolds, called *piecewise-linear manifolds*.

Definition 3.25 *(Piecewise Linear manifold)* Let $\mathbb{M}$ be a manifold. A triangulation of $\mathbb{M}$ is called a piecewise linear manifold and noted $\mathcal{M}$.

3.2.3 Topological invariants

In general, it is very difficult to prove that there exists a homeomorphism between two topological spaces. For that reason, topological invariants have been introduced.

Definition 3.26 *(Topological invariant)* A topological invariant of a topological space is a property that is preserved under homeomorphism.

There are a number of topological invariants. For example, all surfaces obtained by subdivision can be structurally characterized according to their number of *connected boundaries*, their *genus* and their *orientability factor*, thanks to a theorem in [Gri81]. However, the complete characterization of objects in three dimensions and higher is much more challenging. For this, it is necessary to introduce more powerful topological invariants. The remainder of this subsection introduces and formalizes some important notions and invariants, such as *homology groups* and *Betti numbers*, which are central to the definition of powerful tools in TDA.

Definition 3.27 *(Path)* Let $(\mathbb{X}, T)$ be a topological space. A path $p : [0,1] \to \mathbb{C} \subset \mathbb{X}$ is a continuous function from the unit interval to a subset of $\mathbb{X}$.

Definition 3.28 *(Connectedness)* A topological space $(\mathbb{X}, T)$ is connected if for any pair of points in $\mathbb{X}$ there exists a path between them.

Definition 3.29 *(Connected components)* The connected components of a topological space are its largest connected subsets.

This definition is instrumental to Homology Theory [Hat02]. The idea be-

hind this theory is to associate topological spaces with algebraic structures (groups). Then, two spaces are considered equivalent if their homology groups are *isomorphic*.

Homology

Here we introduce more precisely the notions of *homology group* and *Betti numbers*, which are the topological invariants on which the work presented in this manuscript mostly relies.

Definition 3.30 *(p-chain)* Let $\mathcal{T}$ be a triangulation and $\mathcal{T}_p$ the set of its p-simplices. A p-chain of $\mathcal{T}$ is a modulo 2 formal sum f of p-simplices: $f : \mathcal{T}_p \to \mathbb{Z}$.

Property 3.1 The set of all p-chains of a triangulation is an abelian group under the operation of addition.

Definition 3.31 *(Incidence function)* Let $\mathcal{T}$ be a triangulation. An incidence function is a function γ defined by:

$$\begin{array}{rl} \gamma : \mathcal{T} \times \mathcal{T} \to & \{0,1\} \\ (\sigma,\tau) \mapsto & 1 \text{ if } \tau \text{ is a face of } \sigma \\ & 0 \text{ otherwise} \end{array}$$

Definition 3.32 *(Boundary operator)* Let $\mathcal{T}$ be a triangulation and $\mathcal{C}_p$ the set of its p-chains. A boundary operator is defined by:

$$\begin{array}{rl} \partial_p : \mathcal{C}_p \to & \mathcal{C}_{p-1} \\ c \mapsto & \sum_{\tau \in \mathcal{T}} \tau \times \gamma(c,\tau) \end{array}$$

In other words, ∂_p sends a p-simplex to the set of $(p-1)$-simplices which constitute its boundary.

Definition 3.33 *(p-cycle)* Let $\mathcal{T}$ be a triangulation and c a p-chain of T. If $\partial_p(c) = 0$ then c is a p-cycle.

Alternatively, a p-cycle is a p-chain without boundary. The boundary of a p-chain is therefore a $(p-1)$-cycle: it has no boundary.

Property 3.2 For any p-chain c, $\partial_{p-1} \circ \partial_p(c) = 0$.

Property 3.3 The set of all p-cycles of a triangulation is a subgroup of the group of its p-chains.

Definition 3.34 *(Group of p-cycles)* The group of all p-cycles of a triangulation T is noted $\mathcal{Z}_p(\mathcal{T})$.

Definition 3.35 *(p-boundary)* Let $\mathcal{T}$ be a triangulation. A p-boundary of $\mathcal{T}$ is the boundary of a $(p+1)$-chain of $\mathcal{T}$.

Property 3.4 The set of all p-boundaries of a triangulation is a subgroup of the group of its p-cycles.

Definition 3.36 *(Group of p-boundaries)* The group of all p-boundaries of a triangulation T is noted $\mathcal{B}_p(\mathcal{T})$.

Definition 3.37 *(Homology group)* Let $\mathcal{T}$ be the triangulation of a topological space. The p^{th} homology group of T is the quotient group of its p-cycles modulo its p-boundaries: $\mathcal{H}_p(\mathcal{T}) = \mathcal{Z}_p(\mathcal{T})/\mathcal{B}_p(\mathcal{T})$.

Equivalently, $\mathcal{H}_p(\mathcal{T}) = \ker(\partial_p)/\mathrm{im}(\partial_{p+1})$. More informally, p-cycles are grouped in a given homology class if they can be "continuously" transformed into one another. Classes thus obtained can be identified to a representative p-cycle, called a *generator*. By counting the number of generators of a homology group, we get topological invariants called *Betti numbers*.

Definition 3.38 *(Betti number)* Let $\mathcal{T}$ be the triangulation. The p^{th} Betti number of $\mathcal{T}$ is the rank of its p^{th} homology group, noted $\beta_p(\mathcal{T})$.

The topology of $\mathcal{T}$ can be described with its *Betti numbers* β_i, which correspond in 3D to the numbers of connected components (β_0), non collapsible cycles (β_1) and voids (β_2).

Definition 3.39 *(Reduced Betti number)* Let $\mathcal{T}$ be a triangulation and β_p its Betti numbers. The reduced Betti numbers $\tilde{\beta}_p$ are defined by:

$$\begin{aligned} \tilde{\beta}_p &= \beta_p \text{ for all } p \geq 1 \\ \tilde{\beta}_0 &= \beta_0 - 1 \text{ if } T \text{ is non-empty} \\ \tilde{\beta}_{-1} &= 1 \text{ if } T \text{ is empty} \end{aligned}$$

This comes from the definition of *reduced* homology groups [EH09], a slight modification that is useful to ensure that β_0 counts the number of components, and if there is no component, then there is no hole. This comes in handy when defining the concept of *critical point index* as we will see later in the next section.

3.2.4 Data representation

Now that we have formally introduced the domain on which scientific data may be defined, we formalize our representation of the data itself.

Definition 3.40 *(Barycentric coordinates)* Let p be a point in $\mathbb{R}^d$ and σ a d-simplex. Then p can be expressed as a linear combination of the 0-simplices of σ, with coefficients α_i, $0 \leq i \leq d$. If the coefficients α_i sum to 1, they are called the barycentric coordinates of p relative to σ.

This holds as the vertices of a d-simplex are all affinely independent.

Property 3.5 The barycentric coordinates of a point of $\mathbb{R}^d$ relative to a d-simplex are unique.

Property 3.6 A point $p \in \mathbb{R}^d$ belongs to a d-simplex σ if and only if all of its barycentric coordinates are in $[0, 1]$.

Definition 3.41 *(Piecewise linear scalar field)* Let $\mathcal{T}$ be a triangulation and h a function mapping the vertices of $\mathcal{T}$ to $\mathbb{R}$. A piecewise linear (PL) scalar field is a function mapping the *points* of $\mathcal{T}$ to $\mathbb{R}$, linearly interpolated from h.

The linear interpolation of f from h can be constructed in the following way: for all points p in a d-simplex σ, $f(p) = \sum_i \alpha_i h(\tau_i)$, where τ_i is the i^{th} vertex of σ. In this way, a PL scalar field is constructed by taking a function valued on the 0-simplices of $\mathcal{T}$ and linearly interpolating on the higher dimensional simplices of $\mathcal{T}$, as illustrated in Fig. 3.6. This linear interpolation can be computed efficiently on modern hardware.

Definition 3.42 *(Lower link)* Let f be a PL scalar field. The lower link of a simplex σ relatively to f is the subset of the link $Lk^-(\sigma) \subset Lk(\sigma)$ whose vertices v have a strictly lower value $f(v)$ than the vertices of σ.

Conversely, we define the upper link $Lk^+(\sigma)$ as the subset of σ's link whose vertices all have a strictly higher value by f.

To classify Lk without ambiguity into either lower or upper links, the restriction of f to the vertices of $\mathcal{M}$ is assumed to be injective. This can be easily enforced in practice by a variant of simulation of simplicity [EM90]. This is achieved by considering an associated injective integer offset $\mathcal{O}_f(v)$, which initially typically corresponds to the vertex position offset in memory. Then, when comparing two vertices, if these share the same value f, their order is disambiguated by their offset $\mathcal{O}_f$.

Figure 3.6 – *A piecewise linear scalar field constructed on a piecewise linear manifold $\mathcal{M}$, using linear interpolation. Values are first defined on the vertices of $\mathcal{M}$ (top left), then linearly interpolated at edges (top right), triangles (bottom left), and in tetrahedra (bottom right).*

Definition 3.43 *(Sub-level set)* The sub-level set $\mathcal{L}^-(i)$ of an isovalue i with respect to a PL scalar field $f : \mathcal{M} \to \mathbb{R}$ is the set $\{p \in \mathcal{M}, f(p) \leq i\}$.

Conversely, the *sur-level set* $\mathcal{L}^+(i)$ of i is the set $\{p \in \mathcal{M}, f(p) \geq i\}$. These two objects serve as segmentation tools in multiple analysis tasks [Bre+11; CSP04; Boc+18].

Definition 3.44 *(Level set)* The level-set $f^{-1}(i)$ of an isovalue $i \in \mathbb{R}$ with respect to a PL scalar field $f : \mathcal{M} \to \mathbb{R}$ is the preimage of i onto $\mathcal{M}$ by f: $f^{-1}(i) = \{p \in \mathcal{M}, f(p) = i\}$

Level-sets of PL scalar fields have interesting properties. In the 3-dimensional case, within every tetrahedra the level-sets of a PL function are parallel 2-dimensional surfaces. This allows to extract them efficiently in practice, for example with the *Marching Tetrahedra* algorithm [DK91].

Definition 3.45 *(Contour)* Let $f : \mathcal{T} \to \mathbb{R}$ be a PL scalar field and $i \in \mathbb{R}$ an isovalue. The connected components of the level-set $f^{-1}(i)$ are called contours.

3.3 Topological abstractions

In scientific visualization, geometrical objects such as level-sets and contours are essential, as they allow to extract meaningful or representative regions in the data.

They are fundamental objects for the segmentation of regions of interest in data where direct visualization is difficult (like compact or occluded data). One of the main ideas of Topological Data Analysis is to perform a segmentation of the data into regions where level-sets and contours are homogeneous from a topological point of view. These regions of interest then form *topological features*, which often bear a different specific meaning, depending on the scientific applicative domain from which the data is coming.

In this section, we define, in its formal context, the principal topological abstraction, for scalar fields, that will serve as a basis for the work presented in this manuscript: the *persistence diagram*. Its definition is closely related to the distribution of *critical points* of the scalar field.

Our formal setting is that of Morse theory [Mil63], which, as briefly outlined in Sec. 3.1, relates the topology of manifolds to functions with sufficiently nice properties (called Morse functions) defined on them.

3.3.1 Critical points

In multivariable calculus, the critical points of a function are defined as the locations in the domain where the gradient of the function vanishes. This formulation does not translate well to the case of piecewise-linear scalar fields.

Instead, we use a result from Morse theory, which states that the critical points of a function f are the only points $p \in \mathcal{M}$ where the Betti numbers of $f^{-1}_{-\infty}(f(p)-\epsilon)$ differs from those of $f^{-1}_{-\infty}(f(p)+\epsilon)$ for $\epsilon \to 0$. Intuitively, when one takes progressively increasing threshold values, the topology of sub-level sets with respect to the threshold changes when crossing critical points. In the piecewise-linear setting, critical points have handy properties.

Definition 3.46

(Critical point) Let f be a PL scalar field on a triangulation $\mathcal{T}$ and v a vertex of $\mathcal{T}$. If $Lk^+(v)$ and $Lk^-(v)$ are simply connected, v is a regular point. Otherwise, v is a critical point of f and $f(v)$ is called a critical isovalue.

Property 3.7 Let f be a PL scalar field on a triangulation $\mathcal{T}$. If the restriction of f to the vertices of $\mathcal{T}$ is injective, then the set of critical points of f is finite and only contains isolated critical points.

Definition 3.47 *(Maximum and minimum)* Let v be a critical point of a PL scalar field f. If $Lk^+(v)$ is empty, then v is a maximum. If $Lk^-(v)$ is empty, then v is a minimum.

Definition 3.48 *(Saddle)* Let v be a critical point of a PL scalar field f. If v is neither a minimum nor a maximum, then it is a saddle.

Fig. 3.7 illustrates this distinction between critical points. For increasing threshold values a, the number of connected components of the level set, given by $\beta_0(f^{-1}(a))$, augments by one when passing a minimum, decreases by one when passing a maximum, and does not change when passing regular points.

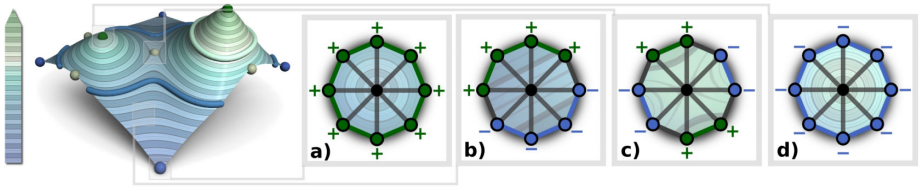

Figure 3.7 – *Points of the domain of a 2-dimensional PL scalar field (left), classified according to the connectivity of their lower links (blue) and upper links (green). The links of a minimum (a), regular point (b), saddle (c), maximum (d) are shown. From [TP12].*

Definition 3.49 *(Saddle multiplicity)* Let v be a saddle of a PL scalar field f, and k is the maximum value of $\beta_0(Lk^-(v))$ and $\beta_0(Lk^+(v))$. The multiplicity of a saddle is $k-1$.

A saddle whose multiplicity is 1 is called a *simple* saddle; a saddle whose multiplicity is higher than 1 is called a *multi-saddle*, or a degenerate critical point. Fig. 3.8 illustrates this case.

Definition 3.50 *(Critical point index)* Let v be a non-degenerate critical point of a PL scalar field f. v is a critical point of index $\mathcal{I}(v) = p$ if the only non-zero reduced Betti number of its lower link is $\tilde{\beta}_{p-1}$.

Remember that the reduced Betti numbers are given by $\tilde{\beta_p} = \beta_p$ and $\tilde{\beta}_0 = \beta_0 - 1$, except when the triangulation is empty (in which case $\tilde{\beta}_0 = \beta_0$ and $\tilde{\beta}_{-1} = 1$). For example, given a PL scalar field f defined on a 2-triangulation, maxima have index 2; saddles have index 1; and minima

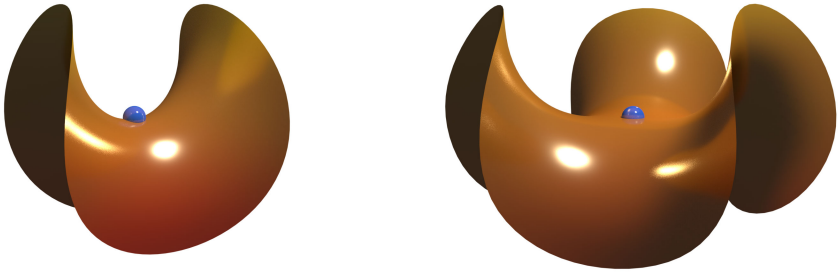

Figure 3.8 – *Saddle points (blue spheres) of a 2-dimensional PL scalar field. A regular saddle (left) has an upper (resp. lower) link consisting of two connected components; whereas a multi-saddle may have more (right, three in this case).*

have index 0. In practice, the computation and classification of critical points according to their index can be done efficiently.

Definition 3.51 *(Critical points)* Let f be a PL scalar field on a triangulation $\mathcal{T}$. The set of critical points of index i of f is noted $\mathcal{C}_f^i$.

Definition 3.52 *(PL Morse scalar field)* Let f be a PL scalar field. f is a PL Morse scalar field if:

- f has no degenerate critical point;
- all the critical points of f have a distinct f value.

Considering PL Morse scalar fields instead of PL scalar fields potentially containing multi-saddles allows to classify its critical points in a robust and consistent way. A PL scalar field can be made into a PL Morse scalar field with a slight numerical perturbation. The first condition in Def. 3.52 can be enforced with a process called multi-saddle unfolding [EH09]; the second condition can be enforced by requiring the restriction of f to the vertices of $\mathcal{T}$ to be injective, with simulation of simplicity [EM90].

The basis for that is an important result of Morse theory in the smooth setting, which states that all smooth functions $f : \mathcal{M} \to \mathbb{R}$ (for which PL scalar fields are a special case), can be approximated by a smooth function g which has no degenerate critical point (corollary 6.8 in [Mil63]). More formally, the set of Morse functions form a dense subset of all smooth functions. In the discrete case, however, the class of PL Morse scalar fields is not dense among the class of all PL scalar fields; therefore, it may be sometimes required to locally alter the triangulation before the scalar field can be Morse.

Other important results of Morse theory in the smooth case still hold in the piecewise-linear case. For example, as discussed in the Sec. 3.1, the critical points of a PL scalar field are related to the topology of the domain; this is illustrated by notorious results such as the Morse-Euler relation [Ban70].
In many scientific applications, the critical points of scalar fields are themselves features of interest. In 2D fluid flow, for instance, extrema of the vertical component of the curl can be used to determine geometrical zones where the flow is turbulent. However, in many practical cases, acquired or simulated scientific data can be noisy. This means that scalar fields can be subject to small oscillations due to numerical or acquisition artifacts, potentially leading to an explosive increase in the number of critical points.

This observation is the main motivation of the framework developed by *Persistent Homology*, which associates a measure of "importance" with each critical point, called *persistence*.

3.3.2 Persistent Homology

The critical points extracted from scientific data cast into a PL Morse scalar field can be important features of interest. However, as the data may be noisy, it is important to distinguish between the critical points which are due to small, insignificant oscillation, and those which are actual features. In the present subsection, we introduce the framework of *Persistent Homology*, developed in this perspective.

The Elder Rule

Let $\mathcal{T}$ be a (connected) triangulation and f a PL Morse scalar field on a triangulation $\mathcal{T}$. The sub-level sets of f form a family of nested sets $\mathcal{T}_a \subseteq \mathcal{T}_b$ for real values $a \leq b$. This family of sets can be pictured by considering the sub-level set $\mathcal{T}_a$, parametrized by a threshold value a, which evolves as the value of a increases.

Now the evolution of the connectedness of $\mathcal{T}_a$ can be visualized, by drawing each connected component of $\mathcal{T}_a$ (for a given value of a) as a point. As the threshold a continuously takes all the possible values of f, a 1-dimensional graph $\mathcal{G}(f)$ is progressively drawn. For example, this graph can be drawn by taking a from the lowest to the highest value of f. As the value of a increases, the connected components of $\mathcal{T}_a$ get larger, in such a way that the arcs on the corresponding graph can merge but never

split. When a reaches the maximum value of f, we are left with a single component (recall that $\mathcal{T}$ is connected).

Therefore $\mathcal{G}(f)$ does not contain any cycle, and it is referred to as a *merge tree*. Constructing the merge tree from the bottom up, $\mathcal{G}(f)$ can be decomposed into paths that grow monotonically with the value of f, and that merge at points called *junctures*. When two paths merge at a juncture, we consider the one whose other endpoint has a lower value to be the *oldest* one; the other is the *youngest* one.

Definition 3.53 *(Elder Rule)* At a juncture between merging paths, the older path continues and the younger path ends.

The Elder Rule generates a unique path decomposition for (PL) Morse scalar fields. Considering $\mathcal{T}_a \subseteq \mathcal{T}_b$, for any two thresholds $a \leq b$, this decomposition is the only one for which the number of paths spanning $[a, b]$ is the number of components of $\mathcal{T}_a$ that have a non-empty intersection with $\mathcal{T}_b$.

Persistent homology group

The concept of *persistence* arises from the formulation of the Elder Rule for the set of homology groups. This is the (more general) algebraic counterpart of the fact illustrated with *merge trees*. We first need to proceed with the definition of filtrations and group homomorphisms.

Definition 3.54 *(Filtration)* Let $\hat{f} : \mathcal{T} \to \mathbb{R}$ be an injective scalar field defined on a triangulation $\mathcal{T}$, so that for each face τ of each simplex $\sigma \in \mathcal{T}$, $\hat{f}(\tau) < \hat{f}(\sigma)$. Let n be the number of simplices in $\mathcal{T}$; let $\mathcal{L}_i^-$ the sub-level set of f by the i^{th} value in the sorted set of simplex values. Then $\mathcal{L}_0^- \subset \mathcal{L}_1^- \subset \cdots \subset \mathcal{L}_{n-1}^- = \mathcal{T}$ is a sequence of sub-complexes of $\mathcal{T}$, called the filtration of $\hat{f}$.

The filtration of $\hat{f}$ can be seen as a progressive construction of $\mathcal{T}$ which adds simplices by chunks. The criterion $\hat{f}(\tau) < \hat{f}(\sigma)$ is enforced so that the sequence of $\mathcal{L}^-$ consists only of sub-complexes. For that the function $\hat{f}$ also associates simplices of dimension higher than 0 to a scalar value.

In practice, for PL scalar fields, the filtration we will be using is the *lower star filtration*, described in the following. Recall that the star of a vertex v is $St(v) = \{\sigma \in \mathcal{T}, v \text{ is a face of } \sigma\}$. By extension, the lower star with respect to a PL scalar field f is $St^-(v) = \{\sigma \in St(v), f(\sigma) \leq f(v)\}$. The lower star is generally not a simplicial complex; however, by adding the missing faces, on obtains the *closed lower star*, which is a sub-complex of $\mathcal{T}$. Provided the restriction of f to the vertices of $\mathcal{T}$ is injective, each

simplex has a unique vertex of maximum value and then belongs to a unique lower star. Therefore, the set of all lower stars form a cover of $\mathcal{T}$. Furthermore, if $\mathcal{K}_i$ is defined as the set of all simplices of $\mathcal{T}$ whose vertices all have a lower value than the i^{th} vertex (in the ordering induced by f), then $\mathcal{K}_i$ is the union of the first i lower stars. This defines a filtration $\mathcal{K}_0 \subset \mathcal{K}_1 \subset \cdots \subset \mathcal{K}_{n-1} = \mathcal{T}$, called the *lower star filtration*. The upper star filtration can be defined symmetrically (by taking the upper star).

Definition 3.55 *(Homomorphism)* Given two groups $(G, *)$ and $(H, \cdot)$, a homomorphism is a map $h : G \to H$ such that for all $u, v \in G$, $h(u * v) = h(u) \cdot h(v)$.

A homomorphism is a map between groups that preserves the group operation. Now a filtration is a sequence of nested complexes, each of which is associated with homology groups H_p (for each dimension p). A filtration induces a sequence of homology groups linked by homomorphisms:

$$H_p(\mathcal{L}_0) \to H_p(\mathcal{L}_1) \to \cdots \to H_p(\mathcal{L}_{n-1}) = H_p(\mathcal{T}) \tag{3.1}$$

New homology classes appear when going from $\mathcal{L}_{i-1}$ to $\mathcal{L}_i$ (corresponding, for instance, to newly created connected components), and some disappear (when, for instance, two classes merge with each other). The idea is then to group together classes that are present between two threshold values $i \leq j$, with the help of *persistent homology groups*.

Definition 3.56 *(Persistent homology group)* The p^{th} persistent homology groups induced by a filtration are the images of the homomorphisms induced by inclusion, denoted $H_p^{i,j}$, for $0 \leq i \leq j \leq n-1$.

Property 3.8 *(Persistent homology group)* Let $H_p^{i,j}$ be the p^{th} persistent homology group induced by a filtration $\mathcal{L}_0 \subset \cdots \subset \mathcal{L}_n$. Then, $H_p^{i,j} = \mathcal{Z}_p(\mathcal{L}_i)/(\mathcal{B}_p(\mathcal{L}_j) \cap \mathcal{Z}_p(\mathcal{L}_i))$.

Thus, for $i \leq j$, the persistent homology groups $H_p^{i,j}$ consist of the homology classes of $\mathcal{L}_i$ that are still present, or *alive*, at $\mathcal{L}_j$. The ranks of persistent homology groups are called *persistent Betti numbers*.

Definition 3.57 *(Persistent Betti number)* The p^{th} persistent Betti numbers are the ranks of the persistent homology groups: $\beta_p^{i,j} = rank(H_p^{i,j})$.

As an example, let us examine the 0^{th} persistent Betti number. Recall that the 0^{th} homology group describes connected components. Let $\mathcal{L}(i) \subset \mathcal{L}(j)$ be two nested complexes with $i < j$. Now let $\mathcal{L}(i, j)$ be the complex con-

taining the connected components of $\mathcal{L}(j)$ having non-empty intersection with the connected components of $\mathcal{L}(j)$. Then the 0^{th} homology group of $\mathcal{L}(i,j)$ consists of the classes of the 0^{th} homology group which existed at the i^{th} filtration step and have *persisted to exist* at the j^{th} filtration step. It is a persistence homology group, noted $H_0^{i,j}$.

Fig. 3.9 shows this mechanism. The 0^{th} Betti numbers of the j^{th} and k^{th} filtration steps in this figure are $\beta_0(j) = 3$ and $\beta_0(k) = 3$. A new class has appeared in $\mathcal{L}(k)$. The 0^{th} persistent Betti number on the interval $[j,k]$ is $\beta_0(i,j) = 2$: there are two classes which have *persisted* from $\mathcal{L}(j)$ to $\mathcal{L}(k)$.

Figure 3.9 – *Some nested complexes induced by the lower-star filtration of a PL scalar field (valuing from red to yellow) defined on a PL 3-manifold (the dragon's head). Sub-complexes are shown in opacity at three filtration levels i (top right) < j (bottom left) < k (bottom right). A new connected component has appeared from $\mathcal{L}(i)$ to $\mathcal{L}(j)$, just before it disappears at $\mathcal{L}(k)$. Another new connected component appears in $\mathcal{L}(k)$.*

The appearance or disappearance of classes as the filtration unfolds correspond to changes in the Betti numbers of sub-level sets of a scalar field f. As outlined in Sec. 3.3.1, these changes precisely occur at the location of critical points of f. Persistent homology classes can then be associated with *pairs* of critical points of f, the critical point with the lowest f value corresponding to the *birth* of the homology class, and the one with the highest f value corresponding to its *death*. Such pairs are called *persis-*

tence pairs; the absolute difference of the two critical point values through f is called the *persistence* of the pair. More practically speaking, we show some of the most prominent persistence pairs associated to filtrations induced by a height function in Fig. 3.10.

When two classes merge, one of the two classes *dies* at the advantage of the other. By applying the Elder Rule (Def. 3.53), we choose the most persistent class (the one with the highest persistence value) to subsist. Once this is fixed, then all critical points of a PL Morse scalar field can be put into persistence pairs without ambiguity.

In this way, Persistent Homology offers a sound framework for defining topological features (in the case of the 0^{th} homology group, these are connected components, but this is applicable for all p^{th} homology groups), and for associating them with a measure of importance: persistence.

3.3.3 Persistence diagrams

In this subsection, we present the topological abstraction called a *persistence diagram*, closely related to the notion of persistence pairs. Let f be a PL Morse scalar field from which we want to analyze critical points.

From this point on, we will consider critical point pairs induced by the star filtrations (Def. 3.54) of f; the lower star filtration yields minimum-saddle pairs and and the upper star filtration gives us saddle-maximum pairs.

As a reminder, by applying the Elder Rule (Def. 3.53), critical points can be arranged in a set of pairs, such that each critical point appears in only one pair (c_i, c_j) with $f(c_i) < f(c_j)$ and $\mathcal{I}(c_i) = \mathcal{I}(c_j) - 1$. Such a pairing indicates that a topological feature of $f^{-1}_{-\infty}(i)$ (connected component, cycle, void, etc.) created at critical point c_i dies at the critical point c_j. For example, as the value i increases, if two connected components of $f^{-1}_{-\infty}(i)$ meet at a given saddle c_j of f, the *youngest* of the two (the one with the highest minimal value, c_i) *dies* at the advantage of the oldest (the one with the lowest minimal value). Critical points c_i and c_j form a *persistence pair*.

Then, the distribution of critical points of f can be represented visually by the *persistence diagram* [ELZ02; CEH05]. The persistence diagram $\mathcal{D}(f)$ embeds each pair (c_i, c_j) in the plane such that its horizontal coordinate equals $f(c_i)$, and the vertical coordinate of c_i and c_j is $f(c_i)$ and $f(c_j)$, corresponding respectively to the *birth* and *death* of the pair. The height of the pair $P(c_i, c_j) = |f(c_j) - f(c_i)|$ is called the *persistence* and denotes the life-span of the topological feature created at c_i and destroyed at c_j. Thus,

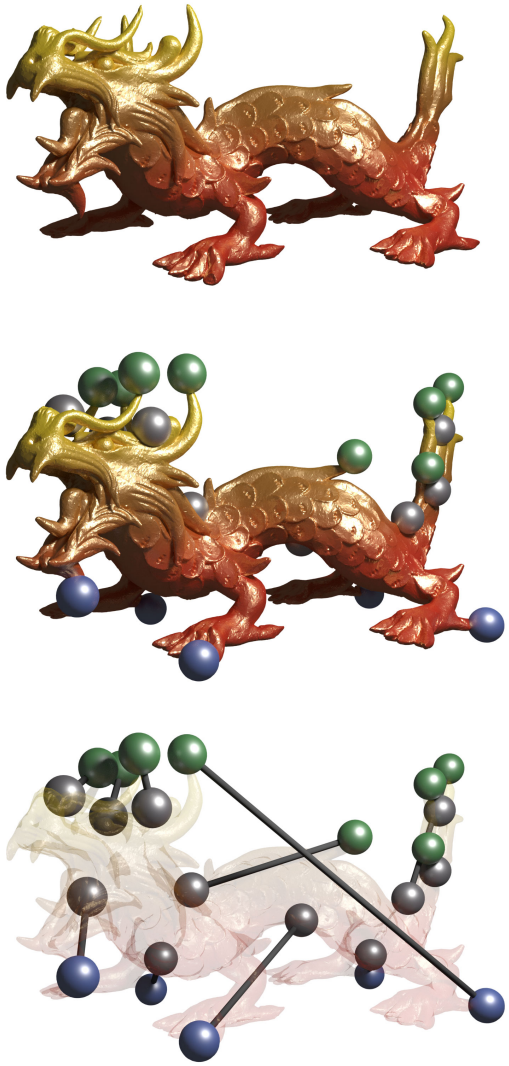

Figure 3.10 – *A PL scalar field defined on a PL 2-manifold (the dragon's surface) as a height function ranging from red to yellow (top). Some critical points are shown as spheres (center; blue: minima, green: maxima, gray: saddles). Using the Elder Rule, critical points are associated in persistence pairs (bottom).*

features with a short life span (for example, noise) will appear in $\mathcal{D}(f)$ as low persistence pairs near the diagonal (Fig. 3.11).

In three dimensions, the persistence of the pairs linking critical points of index $(0,1)$, $(2,3)$ and $(1,2)$ denotes the life-span of connected components, voids and non-collapsible cycles of $f^{-1}_{-\infty}(i)$.

The practical interest of this visual representation is that it quickly hints at the distribution and relative importance of critical points. Small oscillations due to noise in the input data are typically represented by pairs with low persistence, in the vicinity of the diagonal. In contrast, the most prominent topological features are associated with large vertical bars (Fig. 3.11, b). In many applications, persistence diagrams help users as a visual guide to interactively tune simplification thresholds in topology-based, multi-scale data segmentation tasks based on other topological abstraction, such as the Reeb graph [Ree46; CSP04; Pas+07; Tie+09; Gue+16; TC16], or the Morse-Smale complex [Gyu+08; Gyu+14; RWS11]. These two topological data structures are discussed in Sec. 3.4.

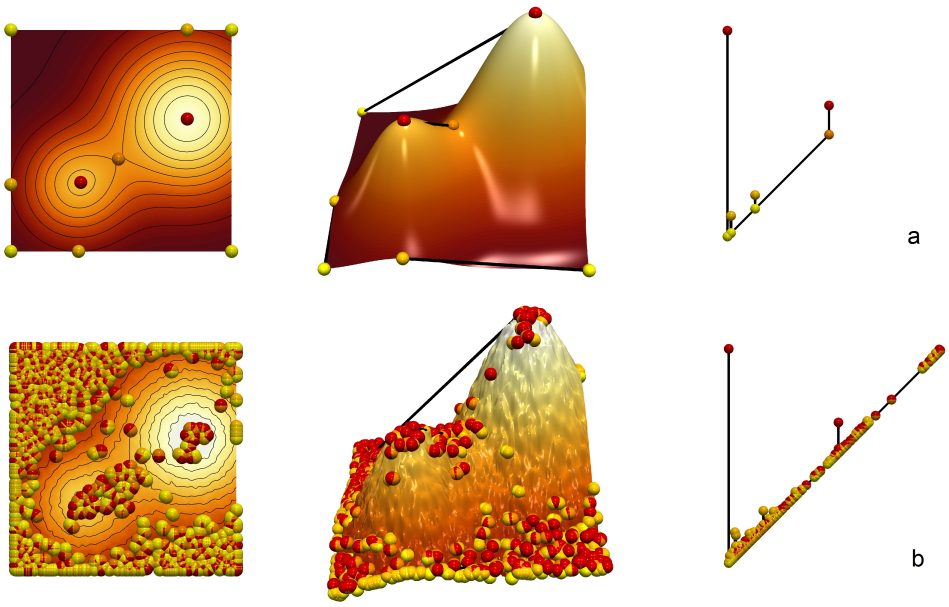

Figure 3.11 – *A smooth (top row) and a noisy (bottom row) scalar field, defined on a 2D domain (left), with their 3D terrain representation (middle) and persistence diagrams (right). Critical points are represented as spheres (red: maxima, orange: saddles, yellow: minima). The largest pairs in the diagrams correspond to the two main hills.*

3.3.4 Metrics between Persistence diagrams

In practical applications it is useful to evaluate the distance between two scalar fields $f, g : \mathcal{M} \rightarrow \mathbb{R}$. Multiple metrics have been defined in this perspective; the L^p-norm $||f - g||_p$ is a classical example.

Reflecting the idea of comparing scalar fields, multiple metrics [CEH05; Cha+09] have been introduced to compare two persistence diagrams $\mathcal{D}(f)$ and $\mathcal{D}(g)$. In the context of this thesis, such metrics are key to:

- Assessing to what extent the topology of a scalar field has been decimated through compression (chapter 4);
- Identifying zones in the data which are similar to one another (chapter 5);
- Ordering a set of scalar fields with respect to a reference *via* a quantitative measure of similarity (chapter 6).

Wasserstein and bottleneck distances

Critical point pairs in persistence diagrams can be associated with a pointwise distance, noted d_p, inspired by the L^p-norm. Given two persistence pairs $a = (a_x, a_y) \in \mathcal{D}(f)$ and $b = (b_x, b_y) \in \mathcal{D}(g)$, d_p can be defined as:

$$d_p(a,b) = \left(|a_x - b_x|^p + |a_y - b_y|^p\right)^{1/p} \tag{3.2}$$

The *Wasserstein* distance [Mon81; Kan42] or Wasserstein metric, noted W_p, between persistence diagrams $\mathcal{D}(f)$ and $\mathcal{D}(g)$ is then defined as:

$$W_p\big(\mathcal{D}(f), \mathcal{D}(g)\big) = \min_{\phi \in \Phi} \left(\sum_{a \in \mathcal{D}(f)} d_p\big(a, \phi(a)\big)^p \right)^{1/p} \tag{3.3}$$

where Φ is the set of all possible assignments ϕ mapping each persistence pair $a \in \mathcal{D}(f)$ to a persistence pair $b \in \mathcal{D}(g)$ with identical critical indices $\mathcal{I}$ or to its diagonal projection, noted $\mathrm{diag}(a)$ – which corresponds to the removal of the corresponding feature from the assignment, with a cost $d_p(a, \mathrm{diag}(a))$. It is illustrated in Fig. 3.12.

Taking the limit as p goes to infinity, one obtains the *bottleneck* distance, $W_\infty\big(\mathcal{D}(f), \mathcal{D}(g)\big)$, given by:

$$W_\infty\big(\mathcal{D}(f), \mathcal{D}(g)\big) = \min_{\phi \in \Phi} \left(\max_{a \in \mathcal{D}(f)} \Big(d_\infty\big(a, \phi(a)\big) \Big) \right) \tag{3.4}$$

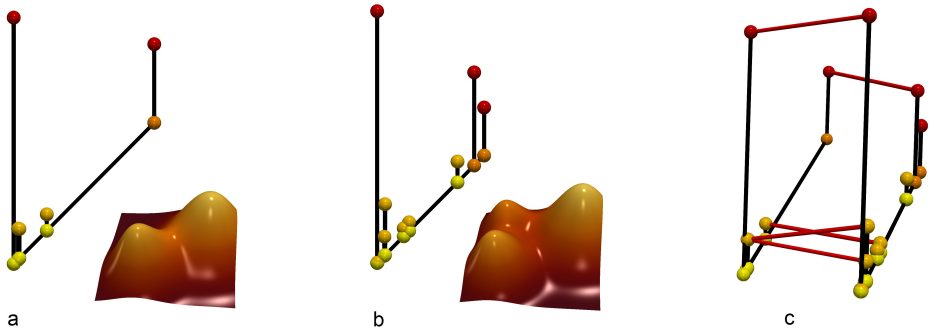

Figure 3.12 – *Persistence diagrams of two distinct 2D scalar fields (a, b), whose persistence pairs are associated by the Wasserstein metric (c, matching pairs are linked together with red segments). The third hill of (b), captured by the rightmost persistence pair, is discarded by the matching.*

An important stability result states that the bottleneck distance between two persistence diagrams is bounded by the maximum norm between the two functions [CEH05]:

$$W_\infty(\mathcal{D}(f), \mathcal{D}(g)) \leq ||f - g||_\infty \tag{3.5}$$

This stability result implies that a small perturbation of amplitude ϵ of the function ($||f - g||_\infty = \epsilon$) will at most imply a bottleneck distance of ϵ between the two persistence diagrams. This further motivates the practical usage of persistence diagrams as a stable and compact representation of the topological features of a scalar field. This important result also motivated the investigation of the reciprocal question, which addresses the problem of reconstructing a function g from the diagram of f, from which persistence pairs below ϵ would have been removed. This problem is generally called *combinatorial reconstruction* [EMP06; Att+13]; some approaches for solving it are discussed in Sec. 3.3.5. Note that the Wasserstein distance does not have the same strong stability properties as the bottleneck distance.

3.3.5 Computational aspects

In this subsection we discuss the practical implications of computing and comparing persistence diagrams.

Persistence diagram

Given a PL scalar field f defined on a PL d-manifold $\mathcal{M}$, the persistence

diagram of f is usually computed in two steps: *(i)* critical point extraction and *(ii)* association of critical points in pairs.

Step *(i)* can be done in a single pass over all the stars of simplices of $\mathcal{M}$. As this processing is local, it can be trivially parallelized. Step *(ii)* can be done with the algorithm by Edelsbruner [ELZ02]. If n is the number of simplices of $\mathcal{M}$, then this is done in $O(n^3)$.

If we restrain to extrema-saddle pairs, then step *(ii)* can be done more efficiently. The classical approach for doing this amounts to computing merge trees as described in Sec. 3.3.2. It requires to perform a sequence of breadth-first searches to find critical point pairs, for instance based on a union-find data structure; the resulting complexity is $O(n \log n + m\alpha m)$ with n the number of vertices of $\mathcal{M}$, m its number of edges and α the inverse Ackermann function [TV98; CSA03].

For the case where $d = 2$ or $d = 3$, the algorithm can be parallelized by tasks [Gue+17] and extended to non-linear interpolants [Nuc+17]. In the studies presented in the remainder of this manuscript, we use the parallel implementation by Gueunet et al. [Gue+17].

Bottleneck distance

Finding the bottleneck distance between two diagrams $\mathcal{D}(f)$ and $\mathcal{D}(g)$, with $|\mathcal{D}(f)| \leq |\mathcal{D}(g)|$, can be seen as assigning tasks (i.e. persistence pairs from $\mathcal{D}(f)$) to parallel machines (i.e. persistence pairs from $\mathcal{D}(g)$) so as to minimize the latest completion time (i.e. the highest pairwise distance between assigned pairs). This is called the *bottleneck assignment problem*, a special case of the *assignment problem*. The time required for a machine to perform a task models the pairwise distance between pairs of the two diagrams.

Times, or *assigment costs*, can be summarized in a rectangular cost matrix (r_{ij}). The problem then amounts to choose a set of $|\mathcal{D}(f)|$ elements from this matrix (r_{ij}), with no two elements sharing the same row or the same column (they are called *independent* elements), so that the largest of these elements is minimal.

There are multiple approaches for solving the bottleneck assignment problem, which can be grouped into *thresholding* and *augmenting path* algorithms [BC99]. Thresholding algorithms consider an increasing threshold cost c^*, build an intermediary matrix $(\bar{r_{ij}})$ containing the elements of (r_{ij}) smaller than c^*, and ∞ instead of elements larger than c^*; then, check whether there is a perfect matching in the graph with adjacency matrix $(\bar{r_{ij}})$ (for instance by using the Hopcroft-Karp algorithm [HK73]).

The concept of a *perfect matching* in the associated graph of a cost matrix is illustrated in Fig. 3.13. The bottleneck distance is obtained for the minimal threshold cost c^* yielding a perfect matching in the associated graph.

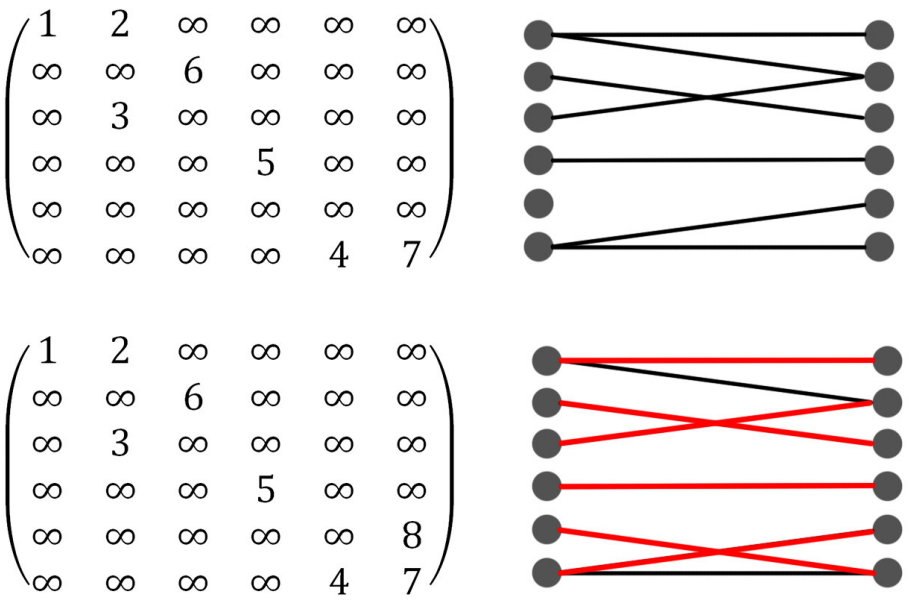

Figure 3.13 – *A cost matrix whose seven lowest elements are associated to a graph (top). The fifth dot on the left, corresponding to the fifth row of the matrix, is not linked to any other dot, so there can be no perfect matching. Taking the eight lowest elements (bottom), one can find a perfect matching in the associated graph (shown in red), so the bottleneck distance in this case is* 8.

An efficient exact implementation of this strategy is the Gabow-Tarjan algorithm [GT88], which combines Hopcroft-Karp rounds in a binary search. If the cost matrix is a square of $n \times n$ elements, and $m \leq n^2$ is the number of finite entries in the matrix, then its complexity is $O(m\sqrt{n}\log n)$. This is the algorithm we implemented and used during the experiments conducted in chapter 4.

Wasserstein distance

The Wasserstein distance can be computed by solving a modified *assignment problem* between diagrams. Namely, the modification comes from the possibility for a pair to be mapped to itself by the assignment with a cost (corresponding to the distance to the diagonal). This optimization problem is discussed in detail in chapter 5.

The seminal exact approach for computing an optimal assignment is the Kuhn-Munkres algorithm [Mun57]; adaptations to the case of persistence diagrams have been proposed [Mor10]. In chapter 5, we revisit this approach and further optimize it with arguments based on sparsity,

achieving speedups of 1 to 2 orders of magnitude. Especially, our adaptation is faster than even approximate methods for small persistence diagrams.

However, the Kuhn-Munkres algorithm is $O(n^3)$, which can quickly become prohibitive for very large persistence diagrams (more than a few thousands of pairs). Approximate methods have been proposed to ease this constraint. Among these, there is the Auction algorithm [BC89], which can be easily implemented and parallelized. Adaptations of the Auction algorithm to the case of persistence diagrams have also been proposed [KMN17]; their performance can be further increased by using lookup acceleration structures, such as KD trees.

Combinatorial reconstruction

Stability results regarding persistence diagrams raise the question of *combinatorial reconstruction*. Concretely, let $\mathcal{D}(f)$ be the persistence diagram of a scalar field f and $\mathcal{D}_\epsilon(f)$ the persistence diagram obtained by removing all persistence pairs from $\mathcal{D}(f)$ whose persistence is less than a given threshold ϵ. Then, combinatorial reconstruction is the problem of computing a function g, so that:

- g is sufficiently close to f (i.e. $||f-g||_\infty$ is small);
- $\mathcal{D}(g) = \mathcal{D}_\epsilon(f)$.

Several approaches to the *combinatorial reconstruction* problem have been proposed for the case of persistence pairs of index $(0,1)$ and $(d-1,d)$, which is precisely our setting. Such approaches address the PL case [EMP06; TP12], filtrations [Att+09] or discrete Morse functions [BLW12].

The algorithm by Tierny and Pascucci [TP12] is noted by dint of its ease of implementation. Typically, this algorithm is given as an input the list of minima and maxima to maintain (in our setting, extrema involved in persistence pairs larger than ϵ) and it produces a function g, along with its corresponding vertex integer offset function $\mathcal{O}_g$, which admits the simplified version of the persistence diagram of f, $W_\infty\big(\mathcal{D}(f),\mathcal{D}(g)\big) \leq \epsilon$, and therefore, thanks to the stability result of Eq. 3.5, which is close to the input function f, $||f-g||_\infty \leq \epsilon$. This procedure can be seen as a function reconstruction process, from critical point constraints with combinatorial guarantees. As discussed in the following, it plays a key role in our compression scheme (presented in chapter 4).

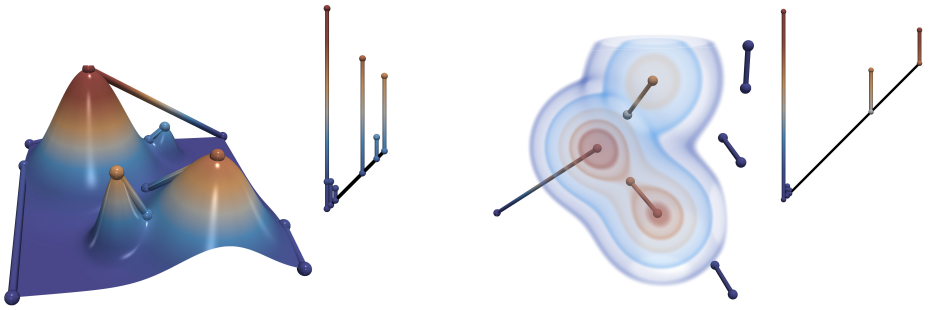

Figure 3.14 – *Four gaussians defined on a 2D plane (top left) and the associated persistence diagram (right); three gaussians defined on a 3D volume (bottom left) with persistence diagram (right). Critical point pairs belonging to the diagrams are also embedded in the domain.*

3.4 Other topological abstractions and extensions

In this section, we briefly present some other tools from TDA which arise from the definition of Persistent Homology, and have important practical applications.

Embedded persistence diagram

In Sec. 3.3.3, we saw that persistence diagrams are combinatorial structures establishing a robust, hierarchical relation between critical points in a scalar field. These critical points initially lie in the definition domain of the scalar function. By plotting persistence pairs using the initial coordinates of critical points (instead of locating them in the *birth-death* space), one obtains another possible representation of the persistence diagram (Fig. 3.14). We call this representation the *embedded persistence diagram*.

No computational overhead is required to obtain this embedding from the persistence diagram (except the cost of storing the coordinates of critical points).

Persistence curve

When challenged to analyze data with a rich underlying topology (which is typically the case of noisy data), often it is not possible to have a clear, comprehensive overview of the distribution of critical points by looking only at the persistence diagrams. In practical cases, it may be difficult to estimate the number of small persistence pairs; moreover, persistence pairs can often be stacked on top of one another in the *birth-death* space. Using the embedded persistence diagram is not a convenient solution either, when the dimension of the domain is greater than two.

To easily visualize populations of persistence pairs, an alternate rep-

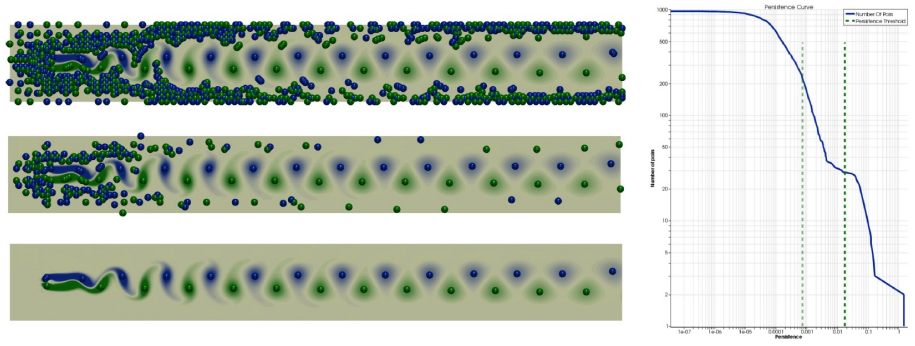

Figure 3.15 – *Von Kármán Vortex street phenomenon, showing the turbulence of a fluid flow behind a solid obstacle [Tie16]. Extrema (spheres, blue: minima, green: maxima) of the the orthogonal component of the flow curl are shown (left). The persistence curve (right) shows populations of critical point pairs in the field; by selecting the most persistent ones (those on the right of vertical cuts of the curve), we can define a hierarchy of critical point sets (left, from top to bottom).*

resentation, called the *persistence curve*, is preferred. It is a 2D plot of the function whose input is a threshold value, τ, and whose output is the number of persistence pairs with persistence smaller than τ. This is shown in Fig. 3.15. Persistence curves are used in practice to detect discontinuities in the distribution of topological features and, consequently, to separated the regimes corresponding to noise from those corresponding to actual features of interest.

The topological abstractions arising from the concepts discussed in this chapter constitute the fundamental toolbox of topological data analysis. They have been widely used, in a number of scientific domains (see Sec. 2.2.2 of chapter 2 for some examples).

In the following chapters, we will show how to capture characteristic structures, in scientific datasets, by making use of these topological abstractions. In particular, the advantages provided by the persistence diagram, namely its robustness, its ability to capture features in a concise and hierarchical way, and the possibility to compute similarity measures, are of critical interest for the applications we are targeting.

Topologically controlled data compression

4

Contents

In a first effort to address the problematic growth in size of scientific data, we present in this chapter a new algorithm for the lossy compression of scalar data defined on 2D or 3D regular grids, with topological control. Certain techniques allow users to control the point-wise error induced by the compression. However, in many scenarios, it is desirable to control in a similar way the preservation of higher-level notions, such as topological features, in order to provide guarantees on the outcome of post-hoc data analyses.

This chapter presents the first compression technique for scalar data which supports a strictly controlled loss of topological features. It provides users with specific guarantees both on the preservation of the important features and on the size of the smaller features destroyed during compression. In particular, we present a simple compression strategy based on a topologically adaptive quantization of the range. Our algorithm provides strong guarantees on the bottleneck distance between persistence diagrams of the input and decompressed data, specifically those associated with extrema. A simple extension of our strategy additionally enables a control on the pointwise error. We also show how to combine our approach with state-of-the-art compressors, to further improve the geometrical reconstruction.

Extensive experiments, for comparable compression rates, demonstrate the superiority of our algorithm in terms of the preservation of topological features. We show the utility of our approach by illustrating the compatibility between the output of post-hoc topological data analysis pipelines, executed on the input and decompressed data, for simulated or acquired data sets. We also provide a lightweight VTK-based C++ implementation of our approach for reproduction purposes. This contribution has been documented in the publication [Sol+18b].

4.1 Scientific issues

Data compression is an important tool for the analysis and visualization of large data sets. In particular, in the context of high performance computing, current trends and predictions [Son+14] indicate increases of the number of cores in super-computers which evolve faster than their memory, IO and network bandwidth. This observation implies that such machines tend to compute results faster than they are able to store or transfer them. Thus, data movement is now recognized as an important bottleneck which challenges large-scale scientific simulations. This challenges even further post-hoc data exploration and interactive analysis, as the output data of simulations often needs to be transferred to a commodity workstation to conduct such interactive inspections. Not only such a transfer is costly in terms of time, but data can often be too large to fit in the memory of a workstation. In this context, data reduction and compression techniques are needed to reduce the amount of data to transfer.

While many lossless compression techniques are now well established [Huf52; ZL77; ZL78; LI06], scientific data sets often need to be compressed at more aggressive rates, which requires lossy techniques [Lak+11; Lin14] (i.e. compression which alters the data). In the context of post-hoc analysis and visualization of data which has been compressed with a lossy technique, it is important for users to understand to what extent their data has been altered, to make sure that such an alteration has no impact on the analysis. This motivates the design of lossy compression techniques with error guarantees.

Several lossy techniques with guarantees have been documented, with a particular focus on pointwise error [LI06; IKK12; DC16]. However, pointwise error is a low level measure and it can be difficult for users to apprehend its propagation through their analysis pipeline, and consequently its impact on the outcome of their analysis. Therefore, it may be desirable to design lossy techniques with guarantees on the preservation of higher-level notions, such as the features of interest in the data. However, the definition of features primarily depends on the target application, but also on the type of analysis pipeline under consideration. This motivates, for each possible feature definition, the design of a corresponding lossy compression strategy with guarantees on the preservation of the said features. In this work, we introduce a lossy compression technique that guarantees the preservation of features of interest, defined with topological notions,

hence providing users with strong guarantees when post-analyzing their data with topological methods.

For instance, we detail in subsection 4.5.3 two analysis pipelines based on topological methods for the segmentation of acquired and simulated data. In the first case, features of interest (bones in a medical CT scan) can be extracted as the regions of space corresponding to the arcs of the split tree [CSA03] which are attached to local maxima of CT intensity. In this scenario, it is important that lossy compression alters the data in a way that guarantees to preserve the split tree, to guarantee a faithful segmentation despite compression and thus, to enable further measurement, analysis and diagnosis even after compression. Thus, it is necessary, for all applications involving topological methods in their post-hoc analysis, to design lossy compression techniques with topological guarantees.

This chapter presents, to the best of our knowledge, the first lossy compression technique for scalar data with such topological guarantees. In particular, we introduce a simple algorithm based on a topologically adaptive quantization of the data range. We carefully study the stability of the persistence diagram [ELZ02; CEH05] of the decompressed data compared to the original one. Given a target feature size to preserve, which is expressed as a persistence threshold ϵ, our algorithm *exactly* preserves the critical point pairs with persistence greater than ϵ and destroys all pairs with smaller persistence. We provide guarantees on the bottleneck and Wasserstein distances between the persistence diagrams, expressed as a function of the input parameter ϵ. Simple extensions to our strategy additionally enable to include a control on the pointwise error and to combine our algorithm with state-of-the-art compressors to improve the geometry of the reconstruction, while still providing strong topological guarantees. Extensive experiments, for comparable compression rates, demonstrate the superiority of our technique for the preservation of topological features. We show the utility of our approach by illustrating the compatibility between the output of topological analysis pipelines, executed on the original and decompressed data, for simulated or acquired data (subsection 4.5.3). We also provide a VTK-based C++ implementation of our approach for reproduction purposes.

4.1.1 Related work

Related existing techniques can be classified into two main categories, addressing lossless and lossy compression respectively.

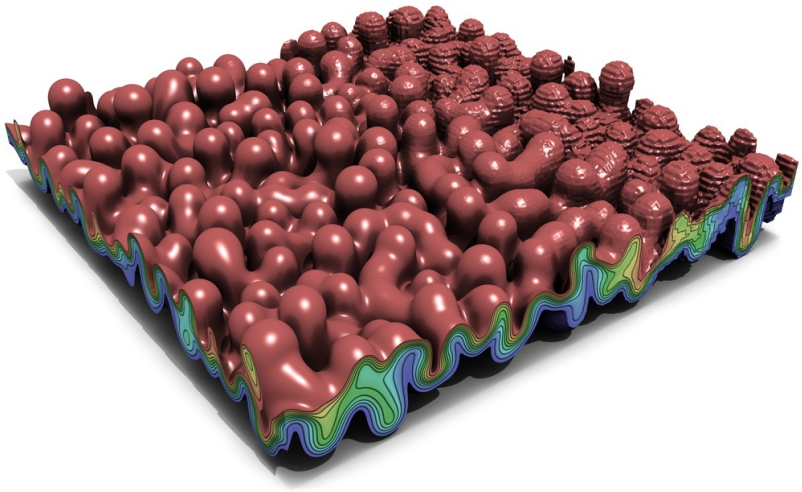

Figure 4.1 – *Rayleigh-Taylor instability compressed with ZFP. Compression factors vary from* 1 *(left) to* 64 *(right); the provided guarantees are not topological, which means the connectivity of bubbles can change undesirably through compression. From [Lin14].*

Regarding lossless compression, several general purpose algorithms have been documented, using entropy encoders [HV91; Gol66; BW94; Huf52], dictionaries [ZL77; ZL78] and predictors [BR07; CW84]. For instance, the compressors associated with the popular file format *Zip* rely on a combination of the LZ77 algorithm [ZL77] and Huffman coding [Huf52]. Such compressors replace recurrent bit patterns in the data by references to a single copy of the pattern. Thus, these approaches reach particularly high compression rates when a high redundancy is present in the data. Several statistical [ILS05; LI06] or non-statistical [RKB06] approaches have been proposed for volume data but often achieve insufficient compression rates in applications (below two [Lin14]), hence motivating lossy compression techniques.

Regarding lossy compression, many strategies have been documented. Some of them are now well established and implemented in international standards, such as GIF or JPEG. Such approaches rely for instance on vector quantization [SW03] or discrete cosine [LM97] and related block transforms [Lin14]. However, relatively little work, mostly related to scientific computing applications, has yet focused on the definition of lossy compression techniques with an emphasis on error control, mostly expressed as a bound on the pointwise error. For instance, though initially introduced for lossless compression, the *FPZIP* compressor [LI06] supports truncation of floating point values, thus providing an explicit relative error control. The *Isabela* compressor [Lak+11] supports predictive temporal compression by B-spline fitting and analysis of quantized error. The fixed

rate compressor *ZFP* (see Fig. 4.1), based on local block transforms, supports maximum error control by not ignoring transform coefficients whose effect on the output is more than a user defined error threshold [LCL16]. More recently, Di and Cappello [DC16] introduced a compressor based on curve fitting specifically designed for pointwise error control. This control is enforced by explicitly storing values for which the curve fitting exceeds the input error tolerance. Iverson et al. [IKK12] also introduced a compressor, named *SQ*, specifically designed for absolute error control. It supports a variety of strategies based on range quantization and/or region growing with an error-based stopping condition. For instance, given an input error tolerance ϵ, the quantization approach segments the range in contiguous intervals of width ϵ. Then, the scalar value of each grid vertex is encoded by the identifier of the interval it projects to in the range. At decompression, all vertices sharing a unique interval identifier are given a common scalar value (the middle of the corresponding interval), effectively guaranteeing a maximum error of ϵ (for vertices located in the vicinity of an interval bound).

Such a range quantization strategy is particularly appealing for the preservation of topological features, as one of the key stability results on persistence diagrams states that the bottleneck distance between the diagrams of two scalar functions is bounded by their maximum pointwise error (Eq. 3.5), meaning that all critical point pairs with persistence higher than ϵ in the input will still be present after a compression based on range quantization. However, a major drawback of this strategy is the constant quantization step size, which implies that large parts of the range, possibly devoid of important topological features, will still be decomposed into contiguous intervals of width ϵ, hence drastically limiting the compression rate in practice.

In contrast, our approach is based on a topologically adaptive range quantization which precisely addresses this drawback, enabling superior compression rates. We additionally show how to extend our approach with absolute pointwise error control. As detailed in Sec. 4.3.3, this strategy preserves persistence pairs with persistence larger than ϵ, *exactly*. In contrast, since it snaps values to the middle of intervals, simple range quantization [IKK12] may alter the persistence of critical point pairs in the decompressed data, by increasing the persistence of smaller pairs (noise) and/or decreasing that of larger pairs (features). Such an alteration is particularly concerning for post-hoc analyses, as it degrades the separation of noise from features and prevents a reliable post-hoc multi-scale analysis,

as the preservation of the persistence of critical point pairs is no longer guaranteed. Finally, note that a few approaches also considered topological aspects [BS98; BPZ99; TR98] but for the compression of meshes, not of scalar data.

4.1.2 Contributions

This chapter presents the following contributions:

1. **Approach:** We present the first algorithm for data compression specifically designed to enforce topological control. We present a simple strategy and carefully describe the stability of the persistence diagram of the output data. In particular, we show that, given a target feature size (i.e. persistence) to preserve, our approach minimizes both the bottleneck and Wasserstein distances between the persistence diagrams of the input and decompressed data.

2. **Extensions:** We show how this strategy can be easily extended to additionally include control on the maximum pointwise error. Further, we show how to combine our compressor with state-of-the-art compressors, to improve the average error.

3. **Application:** We present applications of our approach to post-hoc analyses of simulated and acquired data, where users can faithfully conduct advanced topological data analysis on compressed data, with guarantees on the maximal size of missing features and the *exact* preservation of the most important ones.

4. **Implementation:** We provide a lightweight VTK-based C++ implementation of our approach for reproduction purposes.

4.2 Preliminaries

This section briefly recalls our formal setting and presents an overview of our approach.

4.2.1 Background

Our compression method is based on persistence diagrams. One of the essential property of this topological abstraction, namely its stability, is illustrated in Fig. 4.2.

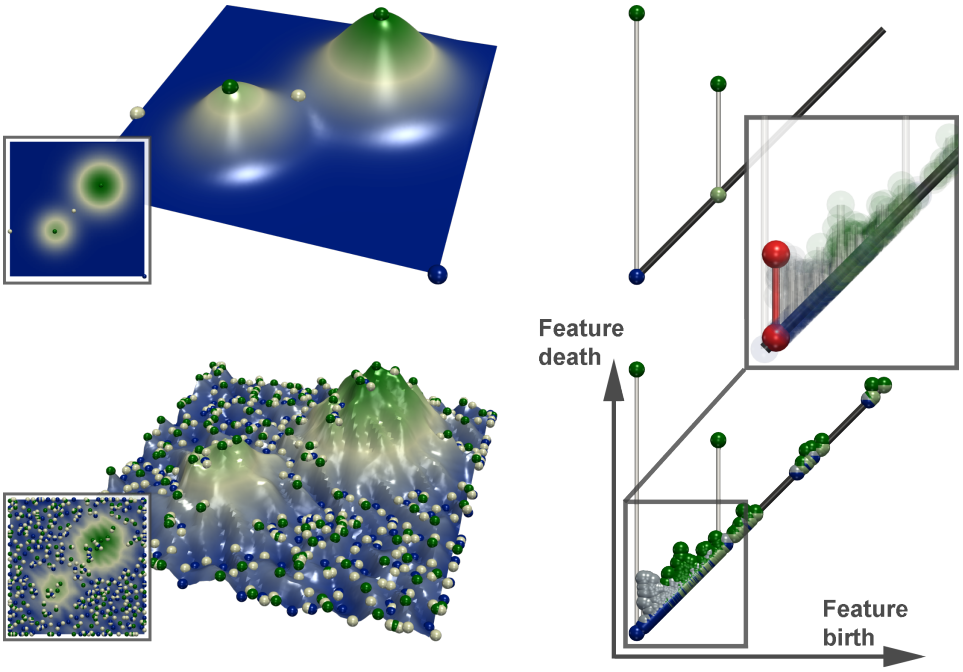

Figure 4.2 – *Critical points (spheres, blue: minima, white: saddles, green: maxima) and persistence diagrams of a clean (top) and noisy (bottom) 2D scalar field (from blue to green). From left to right: original 2D data, 3D terrain representation, persistence diagram. The diagrams clearly exhibit in both cases two large pairs, corresponding to the two main hills. In the noisy diagram (bottom), small bars near the diagonal correspond to noisy features in the data. In this scenario, the bottleneck distance between the diagrams is exactly equal to the persistence of the largest unmatched feature (red pair in the zoomed inset, center right) while the Wasserstein distance is the sum of the persistence of all unmatched pairs.*

In the rest of the chapter, when discussing persistence diagrams, we will only consider critical point pairs of index $(0, 1)$ and $((d-1), d)$. The impact of this simplification is discussed in Sec. 4.5.4.

Distances

In order to evaluate the quality of compression algorithms, several metrics have been defined to evaluate the distance between the decompressed data, noted $g : \mathcal{M} \to \mathbb{R}$, and the input data, $f : \mathcal{M} \to \mathbb{R}$. The p-norm, noted $||f - g||_p$, is a classical example:

$$||f - g||_p = \Big(\sum_{v \in \mathcal{M}} |f(v) - g(v)|^p \Big)^{\frac{1}{p}} \tag{4.1}$$

Typical values of p with practical interests include $p = 2$ and $p \to \infty$. In particular, the latter case, called the *maximum norm*, is used to estimate the maximum pointwise error:

$$||f - g||_\infty = \max_{v \in \mathcal{M}} |f(v) - g(v)| \tag{4.2}$$

In the compression literature, a popular metric is the the *Peak Signal to Noise Ratio* (PSNR), where $|\sigma_0|$ is the number of vertices in $\mathcal{M}$:

$$PSNR = 20\ log_{10}\Big(\frac{\sqrt{|\sigma_0|}}{2} \times \frac{\max_{v\in\mathcal{M}} f(v) - \min_{v\in\mathcal{M}} f(v)}{||f-g||_2}\Big) \tag{4.3}$$

As highlighted in chapter 3, in the context of topological data analysis, several metrics [CEH05] have been introduced too, in order to compare persistence diagrams. In our context, such metrics will be instrumental to evaluate the preservation of topological features after decompression. The *bottleneck* distance [CEH05], noted $W_\infty\big(\mathcal{D}(f), \mathcal{D}(g)\big)$, is a popular example.

Intuitively, in the context of scalar data compression, the bottleneck distance between two persistence diagrams can be usually interpreted as the maximal size of the topological features which have not been maintained through compression (Fig. 4.2). A simple variant of the bottleneck distance, that is slightly more informative in the context of data compression, is the *Wasserstein* distance (see Eq. 3.3). In contrast to the bottleneck distance, the Wasserstein distance will take into account the persistence of *all* the pairs which have not been maintained through compression (not only the largest one).

4.2.2 Overview

An overview of our compression approach is presented in Fig. 4.3. First, the persistence diagram of the input data $f : \mathcal{M} \to \mathbb{R}$ is computed so as to evaluate noisy topological features to later discard. The diagram consists of all critical point pairs of index $(0, 1)$ and $(d-1, d)$. Next, given a target size for the preservation of topological features, expressed as a persistence threshold ϵ, a simplified function $f' : \mathcal{M} \to \mathbb{R}$ is reconstructed [TP12] from the persistence diagram of f, $\mathcal{D}(f)$, from which all persistence pairs below ϵ have been removed (Fig. 4.3(a)). Next, the image of $\mathcal{M}$, $f'(\mathcal{M})$, is segmented along each critical value of f'. A new function $f'' : \mathcal{M} \to \mathbb{R}$ is then obtained from f' by assigning to each vertex the mid-value of the interval it maps to. This constitutes a topologically adaptive quantization of the range (Fig. 4.3(c)). This quantization can optionally be further subdivided to enforce a maximal pointwise error (Fig. 4.3(d)). At this point, the data can be compressed by storing the list of critical values of f'' and storing for each vertex the identifier of the interval it maps to. Optionally, the input data $f$can be compressed independently by state-of-the-art compressors, such as ZFP [Lin14] (Fig. 4.3(e)).

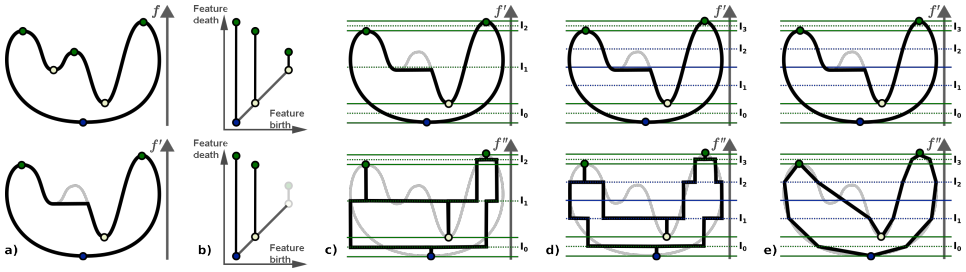

Figure 4.3 – *Overview of our topologically controlled lossy compression scheme on a 2D elevation example. First the input data $f : \mathcal{M} \to \mathbb{R}$ is pre-simplified into a function f' ((a), from top to bottom) to remove all topological features below a user persistence tolerance ϵ (as illustrated by the persistence diagram (b)). The compression is achieved by a topologically adaptive quantization of the range, which is segmented along the critical values of f' (c). A quantized function f'' is constructed ((c), bottom) to only use a finite set of possible data values for regular vertices, hence guaranteeing data compression, while still enforcing original values at critical points. This approach can be extended with point wise error control ((d)), by refining each quantization interval of f' larger than a target width ((d), bottom). Moreover, our approach can be combined with any third party compressor (e) to further improve the geometry of the compressed data.*

At decompression, a first function $g' : \mathcal{M} \to \mathbb{R}$ is constructed by re-assigning to each vertex the mid-value of the interval it maps to. Optionally, if the data has been compressed with a third-party compressor, such as ZFP [Lin14], at decompression, each vertex value is cropped to the extent of the interval it should map to. Last, a function $g : \mathcal{M} \to \mathbb{R}$ is reconstructed from the prescribed critical points of f' [TP12], to remove any topological feature resulting from compression artifacts.

4.3 Data compression

This section presents our topologically controlled compression scheme. In addition to topological control (Sec. 4.3.1), our approach can optionally support pointwise error control (Sec. 4.3.3) as well as combinations with existing compressors (Sec. 4.3.4). The format of the files generated by our compressor is described in Sec. 4.3.2.

4.3.1 Topological control

The input of our algorithm is the input data, $f : \mathcal{M} \to \mathbb{R}$, as well as the size of the topological features to preserve through compression. This size is expressed as a persistence threshold ϵ.

First, the persistence diagram of the input data, noted $\mathcal{D}(f)$ is computed. Next, a simplified version of the input data, noted $f' : \mathcal{M} \to \mathbb{R}$, is

constructed such that f' admits a persistence diagram which corresponds to that of f, but from which the critical point pairs with persistence smaller than ϵ have been removed. This simplification is achieved by using the algorithm by Tierny and Pascucci [TP12], which iteratively reconstructs sub-level sets to satisfy topological constraints on the extrema to preserve. In particular, this algorithm is given as constraints the extrema of f to preserve, which are in our current setting the critical points involved in pairs with persistence larger than ϵ. In such a scenario, this algorithm has been shown to reconstruct a function f' such that $||f - f'||_\infty \leq \epsilon$ [TP12]. At this point, f' carries all the necessary topological information that should be preserved through compression.

In order to compress the data, we adopt a strategy based on range quantization. By assigning only a small number n of possible data values on the vertices of $\mathcal{M}$, only $log_2(n)$ bits should be required in principle for the storage of each value (instead of 64 for traditional floating point data with double precision). Moreover, encoding the data with a limited number of possible values is known to constitute a highly favorable configuration for post-process lossless compression, which achieves high compression rates for redundant data.

The difficulty in our context is to define a quantization that respects the topology of f', as described by its persistence diagram $\mathcal{D}(f')$. To do so, we collect all critical values of f' and segment the image of $\mathcal{M}$ by f', noted $f'(\mathcal{M})$, into a set of contiguous intervals $I = \{I_0, I_1, \dots I_n\}$, all delimited by the critical values of f' (Figure 4.3, second column, top). Next, we create a new function $f'' : \mathcal{M} \to \mathbb{R}$, where all critical points of f' are maintained at their corresponding critical value and where all regular vertices are assigned to the mid-value of the interval I_i they map to. This constitutes a topologically adaptive quantization of the range: only n possible values will be assigned to regular vertices. Note that although we modify data values in the process, the critical vertices of f' are still critical vertices (with identical indices) in f'', as the lower and upper links (Sec. 4.2.1) of each critical point are preserved by construction.

4.3.2 Data encoding

The function f'' is encoded in a two step process. First a topological index is created. This index stores the identifier of each critical vertex of f'' as well as its critical value, and for each of these, the identifier i of the interval I_i immediately above it if and only if some vertices of $\mathcal{M}$ indeed project

to I_i through f''. This strategy enables the save of identifiers for empty intervals.

The second step of the encoding focuses on data values of regular vertices of f''. Each vertex of $\mathcal{M}$ is assigned the identifier i of the interval I_i it projects to through f''. For n_v vertices and n_i non-empty intervals between n_c critical points, we store per-vertex interval identifiers (n_v words of $\log_2(n_i)$ bits), critical point positions in a vertex index (n_c words of $\log_2(n_v)$ bits), critical types (n_c words of 2 bits) and critical values (n_c floats).

Since it uses a finite set of data values, the buffer storing interval assignments for all vertices of $\mathcal{M}$ is highly redundant. Thus, we further compress the data (topological index and interval assignment) with a standard lossless compressor (*Bzip2* [Sew17]).

4.3.3 Pointwise error control

Our approach has been designed so far to preserve topological features thanks to a topologically adaptive quantization of the range. However, this quantization may be composed of arbitrarily large intervals, which may result in large pointwise error.

Our strategy can be easily extended to optionally support a maximal pointwise error with regard to the input data $f : \mathcal{M} \to \mathbb{R}$, still controlled by the parameter ϵ. In particular, this can be achieved by subdividing each interval (Sec. 4.3.1) according to a target maximal width w, prior to the actual quantization and data encoding (Sec. 4.3.2). Since the topologically simplified function f' is guaranteed to be at most ϵ-away from f ($||f - f'||_\infty \leq \epsilon$, Sec. 4.3.1), further subdividing each interval with a maximum authorized width of w will result in a maximum error of $\epsilon + w/2$ when quantizing the data into f''. For instance, a local maximum of f of persistence lower than ϵ can be pulled down by at most ϵ when simplifying f into f' [TP12] (Fig. 4.4, center) and then further pulled down by up to $w/2$ when being assigned the mid-value of its new interval (of width w, Fig. 4.4, right). In practice, for simplicity, we set $w = \epsilon$. Hence, a maximum pointwise error of $3\epsilon/2$ is guaranteed at compression ($||f - f''||_\infty \leq 3\epsilon/2$).

4.3.4 Combination with state-of-the-art compressors

The compression approach we presented so far relies on a topologically adaptive quantization of the range (with optional pointwise error control). The compression is achieved by only allowing a small number of possi-

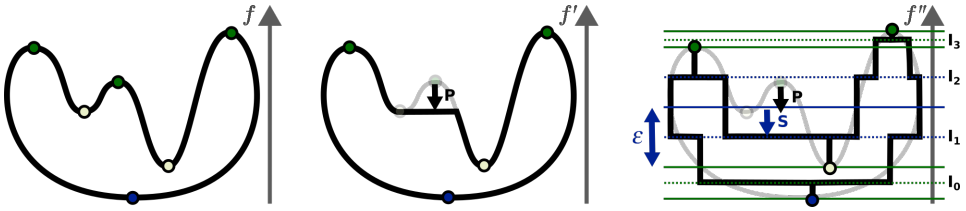

Figure 4.4 – *Topologically controlled compression with pointwise error control. When pre-simplifying the input data $f : \mathcal{M} \to \mathbb{R}$ (left) into f' (center), the value variation of each simplified extremum e equals the persistence P of the pair it belongs to, which is bounded by construction by ϵ: $|f(e) - f'(e)| = P \leq \epsilon$ [CEH05; TP12]. When adding pointwise error control, each interval is subdivided such that its width does not exceed ϵ (right). Thus, when constructing the quantized function f'' which maps each vertex to the middle of its interval, each simplified extremum e of f may further move by a* snapping *distance s to the middle of its interval, which is itself bounded by half the width of the interval ($\epsilon/2$). Thus, $|f(e) - f''(e)| = P + s \leq \epsilon + \epsilon/2$.*

ble scalar values in the compressed data, which may typically result in noticeable *staircase* artifacts. To address this, our method can be optionally combined seamlessly with any state-of-the-art lossy compressor. For our experiments, we used *ZFP* [Lin14]. Such a combination is straightforward at compression time. In particular, in addition to the topological index and the compressed quanta identifiers (subsection 4.3.2), the input data $f : \mathcal{M} \to \mathbb{R}$ is additionally and independently compressed by the third-party compressor (*ZFP*).

4.4 Data decompression

This section describes the decompression procedure of our approach, which is symmetric to the compression pipeline described in the previous section (Sec. 4.3). This section also further details the guarantees provided by our approach regarding the bottleneck (W_∞) and Wasserstein (W_2) distances between the persistence diagrams of the input data and the decompressed data, noted $g : \mathcal{M} \to \mathbb{R}$ (Sec. 4.4.4).

4.4.1 Data decoding

First, the compressed data is decompressed with the lossless decompressor *Bzip2* [Sew17]. Next, a function $g' : \mathcal{M} \to \mathbb{R}$ is constructed based on the topological index and the interval assignment buffer (Sec. 4.3.2). In particular, each critical vertex is assigned its critical value (as stored in the topological index) and regular vertices are assigned the mid-value of the interval they project to, based on the interval assignment buffer (Sec. 4.3.2).

4.4.2 Combination with state-of-the-art decompressors

If a state-of-the-art compression method has been used in conjunction with our approach (Sec. 4.3.4), we use its decompressor to generate the function $g' : \mathcal{M} \to \mathbb{R}$. Next, for each vertex v of $\mathcal{M}$, if $g'(v)$ is outside of the interval I_i where v is supposed to project, we snap $g'(v)$ to the closest extremity of I_i. This guarantees that the decompressed data respects the topological constraints of $\mathcal{D}(f')$, as well as the optional target pointwise error (Sec. 4.3.3).

4.4.3 Topological reconstruction

The decompressed function g' may contain at this point extraneous critical point pairs, which were not present in $\mathcal{D}(f')$ (Sec. 4.3.1). For instance, if a state-of-the-art compressor has been used in conjunction with our approach, arbitrary oscillations within a given interval I_i can still occur and result in the apparition of critical point pairs in $\mathcal{D}(g')$ (with a persistence smaller than the target interval width w, Sec. 4.3.3) which were not present in $\mathcal{D}(f')$. The presence of such persistence pairs impacts the distance metrics introduced in Sec. 4.2.1, and therefore impacts the quality of our topology controlled compression. Thus, such pairs need to be simplified in a post-process.

Note that, even if no third-party compressor has been used, since our approach is based on a topologically adaptive quantization of the range, large flat plateaus will appear in g'. Depending on the vertex offset $\mathcal{O}_{g'} : \mathcal{M} \to \mathbb{R}$ (used to disambiguate flat plateaus, Sec. 4.2.1), arbitrarily small persistence pairs can also occur. Therefore, for such flat plateaus, $\mathcal{O}_{g'}$ must be simplified to guarantee its monotonicity everywhere except at the desired critical vertices (i.e. those stored in the topological index, Sec. 4.3.2).

Thus, whether a state-of-the-art compressor has been used or not, the last step of our approach consists in reconstructing the function $g : \mathcal{M} \to \mathbb{R}$ from g' by enforcing the critical point constraints of f' (stored in the topological index) with the algorithm by Tierny and Pascucci [TP12]. Note that this algorithm will automatically resolve flat plateaus, by enforcing the monotonicity of $\mathcal{O}_g$ everywhere except at the prescribed critical points [TP12]. Therefore, the overall output of our decompression procedure is the scalar field $g : \mathcal{M} \to \mathbb{R}$ as well as its corresponding vertex integer offset $\mathcal{O}_g : \mathcal{M} \to \mathbb{N}$.

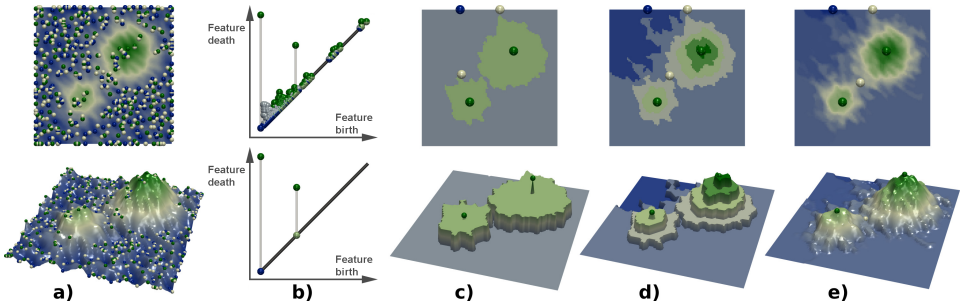

Figure 4.5 – *Compression of the noisy 2D data set from Fig. 4.2 ((a), 80,642 bytes, top: 2D data, bottom: 3D terrain). In all cases (c-e), our compression algorithm was configured to maintain topological features more persistent than 20% of the function range, as illustrated with the persistence diagrams ((b), top: original noisy data $\mathcal{D}(f)$, bottom: decompressed data $\mathcal{D}(g)$). Our topology controlled compression (c), augmented with pointwise error control (d), and combined with ZFP [Lin14] ((e), one bit per scalar) yields compression rates of 163, 50 and 14 respectively.*

4.4.4 Topological guarantees

The last step of our decompression scheme, topological reconstruction (Sec. 4.4.3), guarantees that $\mathcal{D}(g)$ admits no other critical points than those of $\mathcal{D}(f')$ (specified in the topological index). Moreover, the corresponding critical values have been strictly enforced (Sec. 4.4.1). This guarantees that $W_\infty\big(\mathcal{D}(g),\mathcal{D}(f')\big) = 0$, and thus:

$$\begin{aligned} W_\infty\big(\mathcal{D}(g),\mathcal{D}(f)\big) &\leq W_\infty\big(\mathcal{D}(g),\mathcal{D}(f')\big) + W_\infty\big(\mathcal{D}(f'),\mathcal{D}(f)\big) \\ &\leq W_\infty\big(\mathcal{D}(f'),\mathcal{D}(f)\big) \end{aligned} \tag{4.4}$$

Since we know that f' is ϵ-away from the original data f (Sec. 4.3.1) and due to the stability of persistence diagrams [CEH05], we then have:

$$W_\infty\big(\mathcal{D}(g),\mathcal{D}(f)\big) \leq ||f - f'||_\infty \leq \epsilon \tag{4.5}$$

Thus, the bottleneck distance between the persistence diagrams of the input and decompressed data is indeed bounded by ϵ, which happens to precisely describe the size of the topological features that the user wants to preserve through compression.

Since $\mathcal{D}(f') \subset \mathcal{D}(f)$ and $W_\infty\big(\mathcal{D}(g),\mathcal{D}(f')\big) = 0$, we have $\mathcal{D}(g) \subset \mathcal{D}(f)$. This further implies that:

$$W_\infty\big(\mathcal{D}(g),\mathcal{D}(f)\big) = \max_{(p,q)\in\big(\mathcal{D}(g)\,\Delta\,\mathcal{D}(f)\big)} P(p,q) \tag{4.6}$$

where $P(p,q)$ denotes the persistence of a critical point pair (p,q) and where $\mathcal{D}(g)\,\Delta\,\mathcal{D}(f)$ denotes the symmetric difference between $\mathcal{D}(g)$ and

$\mathcal{D}(f)$ (i.e. the set of pairs present in $\mathcal{D}(f)$ but not in $\mathcal{D}(g)$). In other words, the bottleneck distance between the persistence diagrams of the input and decompressed data exactly equals the persistence of the most persistent pair present in $\mathcal{D}(f)$ but not in $\mathcal{D}(g)$ (in red in Fig. 4.2). This guarantees the *exact* preservation of the topological features selected with an ϵ persistence threshold.

As for the (2-)Wasserstein distance, with the same rationale, we get:

$$W_2\big(\mathcal{D}(g), \mathcal{D}(f)\big) = \sum_{(p,q)\in\left(\mathcal{D}(g)\,\Delta\,\mathcal{D}(f)\right)} \big(P(p,q)\big)^2 \tag{4.7}$$

In other words, the Wasserstein distance between the persistence diagrams of the input and decompressed data will be exactly equal to sum of the persistence of all pairs present in $\mathcal{D}(f)$ but not in $\mathcal{D}(g)$ (small bars near the diagonal in Fig. 4.2, bottom), which corresponds to all the topological features that the user precisely wanted to discard.

Finally, for completeness, we recall that, if pointwise error control was enabled, our approach guarantees $||f - g||_\infty \leq 3\epsilon/2$ (Sec. 4.3.3).

4.5 Results

This section presents experimental results obtained on a desktop computer with two Xeon CPUs (3.0 GHz, 4 cores each), with 64 GB of RAM. For the computation of the persistence diagram and the topological simplification of the data, we used the algorithms by Tierny and Pascucci [TP12] and Gueunet et al. [Gue+17], whose implementations are available in the Topology ToolKit (TTK) [Tie+17]. The other components of our approach (including bottleneck and Wasserstein distance computations) have been implemented as TTK modules. Note that our approach has been described so far for triangulations. However, we restrict our experiments to regular grids in the following as most state-of-the-art compressors (including *ZFP* [Lin14]) have been specifically designed for regular grids. For this, we use the triangulation data-structure from TTK, which represents implicitly regular grids with no memory overhead using a 6-tet subdivision.

Fig. 4.5 presents an overview of the compression capabilities of our approach on a noisy 2D example. A noisy data set is provided on the input. Given a user threshold on the size of the topological features to preserve, expressed as a persistence threshold ϵ, our approach generates decompressed data-sets that all share the same persistence diagram $\mathcal{D}(g)$ (Figure 4.5(b), bottom), which is a subset of the diagram of the input data $\mathcal{D}(f)$

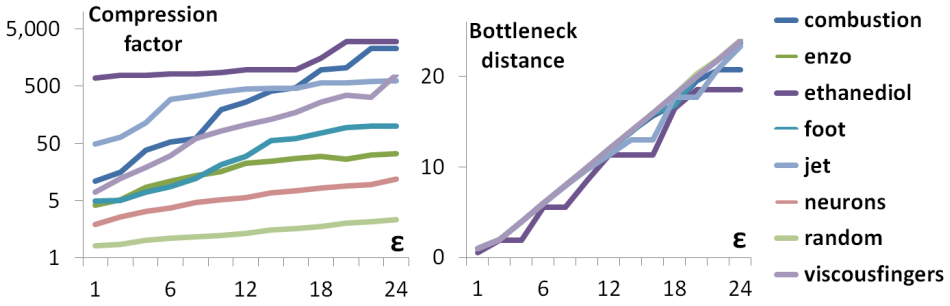

Figure 4.6 – *Performance analysis of our compression scheme (topological control only). Left: Compression rate for various 3D data sets, as a function of the target persistence threshold ϵ (percentage of the function range). Right: bottleneck distance between the persistence diagrams of the input and decompressed data, $W_\infty(\mathcal{D}(f), \mathcal{D}(g))$, for increasing target persistence thresholds ϵ.*

(Figure 4.5(b), top) from which pairs with a persistence lower than ϵ have been removed, and those above ϵ have been *exactly* preserved. As shown in this example, augmenting our approach with pointwise error control or combining it with a state-of-the-art compressor allows for improved geometrical reconstructions, but at the expense of much lower compression rates. Note that the Figure 4.5(e) shows the result of the compression with *ZFP* [Lin14], which has been augmented with our topological control. This shows that our approach can enhance any existing compression scheme, by providing strong topological guarantees on the output.

4.5.1 Compression performance

We first evaluate the performance of our compression scheme with topological control only on a variety of 3D data sets all sampled on 512^3 regular grids. Fig. 4.6 (left) presents the evolution of the compression rates for increasing target persistence thresholds ϵ expressed as percentages of the data range. This plot confirms that when fine scale structures need to be preserved (small ϵ values, left), smaller compression rates are achieved, while higher compression rates (right) are reached when this constraint is relaxed. Compression rates vary among data sets as the persistence diagrams vary in complexity. The Ethane Diol dataset (topmost curve) is a very smooth function coming from chemical simulations. High compression factors are achieved for it, almost irrespective of ϵ. On the contrary, the random dataset (bottom curve) exhibits a complex persistence diagram, and hence lower compression rates.

In between, all data sets exhibit the same increase in compression rate for increasing ϵ values. Their respective position between the two extreme

Table 4.1 – *Detailed computation times on* 512^3 *regular grids* ($\epsilon = 5\%$)*, with and without compression-time simplification. P, S, Q and L stand for the persistence diagram, topological simplification, topological quantization and lossless compression efforts (%).*

Data-set	With compression-time simplification						No simplification		Decompr.
	P (%)	S (%)	Q (%)	L (%)	Total (s)	Compr. Rate	Total (s)	Compr. Rate	Time (s)
Combustion	8.4	89.3	0.7	1.6	**593.9**	121.3	**64.1**	111.1	213.3
Elevation	14.6	84.1	1.2	0.1	**157.0**	174,848.0	**25.3**	174,848.0	211.5
EthaneDiol	12.6	86.7	0.5	0.2	**490.0**	2,158.6	**63.0**	2,158.6	228.9
Enzo	9.5	86.7	1.0	2.7	**695.6**	24.5	**91.8**	19.8	204.0
Foot	12.6	81.9	1.6	3.8	**380.6**	12.1	**68.4**	7.75	205.7
Jet	22.2	75.7	0.6	1.5	**451.3**	315.6	**111.1**	287.6	220.3
Random	15.9	76.0	2.8	5.3	**1357.1**	1.5	**307.7**	1.5	101.2

configurations of the spectrum (elevation and random) depend on their input topological complexity (number of pairs in the persistence diagram).

Fig. 4.6 (right) plots the evolution of the bottleneck distance between the input and decompressed data, $W_\infty\big(\mathcal{D}(f), \mathcal{D}(g)\big)$, for increasing target persistence threshold ϵ, for all data sets. This plot shows that all curves are located below the identity diagonal. This constitutes a practical validation of our guaranteed bound on the bottleneck distance (Eq. 4.5). Note that, for a given data set, the proximity of its curve to the diagonal is directly dependent on its topological complexity. This result confirms the strong guarantees regarding the preservation of topological features through compression.

Table 4.1 provides detailed timings for our approach and shows that most of the compression time (at least 75%, *S* column) is spent simplifying the original data f into f' (section 4.3). If desired, this step can be skipped to drastically improve time performance, but at the expense of compression rates (Table 4.1, right). Indeed, as shown in Figure 4.7, skipping this simplification step at compression time results in quantized function f'' that still admits a rich topology, and which therefore constitutes a less favorable ground for the post-process lossless compression (higher entropy). Note however that our implementation has not been optimized for execution time. We leave time performance improvement for future work.

4.5.2 Comparisons

Next, we compare our approach with topological control only to the *SQ* compressor [IKK12], which has been explicitly designed to control pointwise error. Thus, it is probably the compression scheme that is the most related to our approach. *SQ* proposes two main strategies for data compression, one which is a straight range quantization (with a constant step

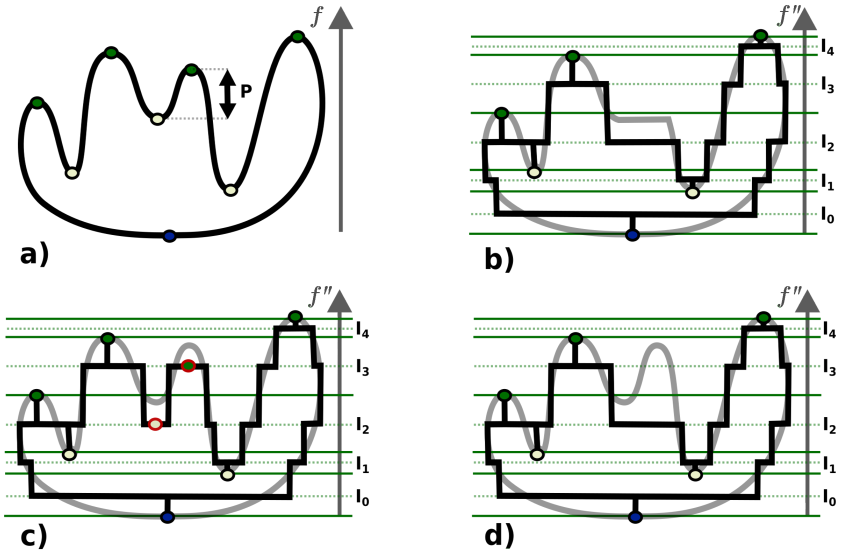

Figure 4.7 – *Topologically controlled compression with (b) and without (c) compression-time simplification. Topological simplification (c to d) removes all critical point pairs not present in the topological index (red circles) and exactly maintains the others [TP12]. Thus, simplifying the data only at decompression (d) yields identical decompressed data (d vs b). The quantized function then admits a richer topology (c vs b), which deteriorates compression rates.*

size ϵ, *SQ-R*) and the other which grows regions in the 3D domain, within a target function interval width ϵ (*SQ-D*). Both variants, which we implemented ourselves, provide an explicit control on the resulting pointwise error ($||f - g||_\infty \leq \epsilon$). As such, thanks to the stability result on persistence diagrams [CEH05], *SQ* also bounds the bottleneck distance between the persistence diagrams of the input and decompressed data. Each pair completely included within one quantization step is indeed flattened out. Only the pairs larger than the quantization step size ϵ do survive through compression. However, the latter are snapped to admitted quantization values. In practice, this can artificially and arbitrarily reduce the persistence of certain pairs, and increase the persistence of others. This is particularly problematic as it can reduce the persistence of important features and increase that of noise, which prevents a reliable multi-scale analysis after decompression. This difficulty is one of the main motivations which led us to design our approach.

To evaluate this, we compare *SQ* to our approach in the light of the Wasserstein distance between the persistence diagrams of the input and decompressed data, $W_2\big(\mathcal{D}(f), \mathcal{D}(g)\big)$. As described in Sec. 4.2.1, this distance is more informative in the context of compression, since not only does it track all pairs which have been lost, but also the changes of the

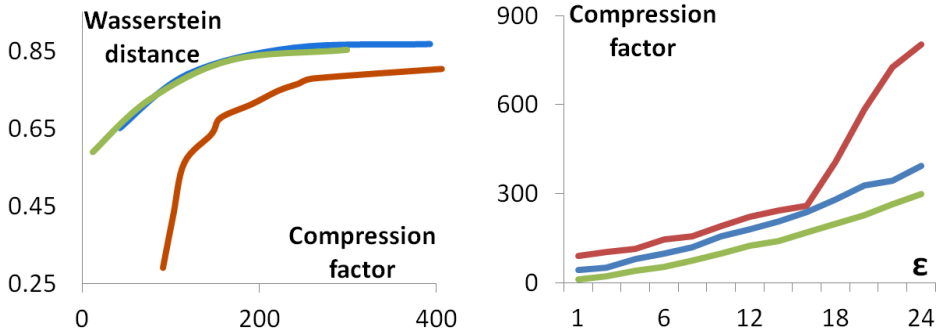

Figure 4.8 – *Comparison to the* SQ *compressor [IKK12] (green: SQ-D, blue: SQ-R, red: our approach). Left: average normalized Wasserstein distance between the persistence diagrams of the input and decompressed data, for increasing compression rates. Right: average compression factors for increasing target persistence thresholds ϵ.*

pairs which have been preserved. Fig. 4.8 (left) presents the evolution of the Wasserstein distance, averaged over all our data sets, for increasing compression rates. This plot shows that our approach (red curve) achieves a significantly better preservation of the topological features than *SQ*, for all compression rates, especially for the lowest ones. As discussed in the previous paragraph, given a quantization step ϵ, *SQ* will preserve all pairs more persistent than ϵ but it will also degrade them, as shown in the above experiment. Another drawback of *SQ* regarding the preservation of topological features is the compression rate. Since it uses a constant quantization step size, it may require many quantization intervals to preserve pairs above a target persistence ϵ, although large portions of the range may be devoid of important topological features. To illustrate this, we compare the compression rates achieved by *SQ* and our approach, for increasing values of the parameter ϵ. As in Fig. 4.6, increasing slopes can be observed. However, our approach always achieves higher compression rates, especially for larger persistence targets.

Next, we study the capabilities offered by our approach to augment a third party compressor with topological control (Secs. 4.3.4 and 4.4.2). In particular, we augmented the *ZFP* compressor [Lin14], by using the original implementation provided by the author (with 1 bit per vertex value). Fig. 4.9 (left) indicates the evolution of the compression rates as the target persistence threshold ϵ increases. In particular, in these experiments, a persistence target of 100% indicates that no topological control was enforced. Thus, these curves indicate, apart from this point, the overhead of topological guarantees over the *ZFP* compressor in terms of data storage. These curves, as it could be expected with Fig. 4.6, show that compres-

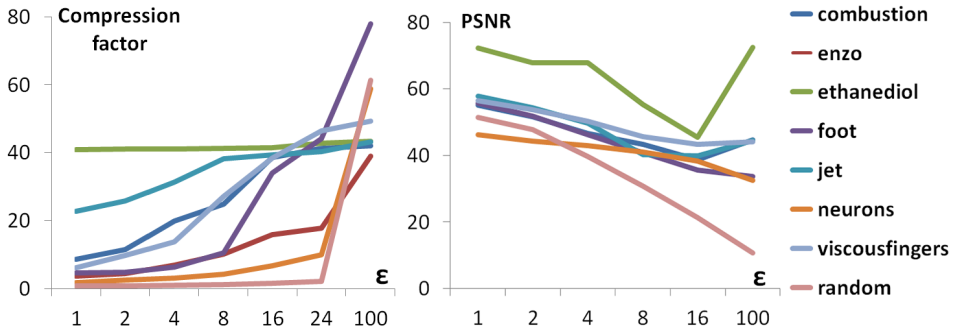

Figure 4.9 – *Augmenting a third party compressor, here* ZFP *[Lin14] (1 bit per vertex value), with topological control. Left: Compression factors for increasing persistence targets ϵ. Right: PSNR for increasing persistence targets ϵ. In these plots (left and right), a persistence target of* 100% *indicates that no topological control was enforced.*

sion rates will rapidly drop down for topologically rich data sets (such as the random one). On the contrary, for smoother data sets, such as Ethane Diol or Jet, high compression rates can be maintained. This shows that augmenting a third party compressor with topological control results in compression rates that adapt to the topological complexity of the input. Fig. 4.9 (right) shows the evolution of the PSNR (Sec. 4.2.1) for decreasing persistence targets ϵ. Surprisingly, in this context, the enforcement of topological control improves the quality of the data decompressed with *ZFP*, with higher PSNR values for little persistence targets. This is due to the rather aggressive compression rate which we used for *ZFP* (1 bit per vertex value) which tends to add noise to the data. Thanks to our topological control (Sec. 4.4.3), such compression artifacts can be cleaned up by our approach.

We evaluate in Table 4.2 the advantage of our topology aware compression over a standard lossy compression, followed at decompression by a topological cleanup (which simplifies all pairs less persistent than ϵ [TP12]). In particular, this table shows that augmenting a third party compressor (such as ZFP) with our topological control (second line) results in more faithful decompressed data (lower Wasserstein distances to the original data) than simply executing the compressor and topologically cleaning the data in a post-process after decompression (first line). This further motivates our approach for augmenting existing compressors with topological control.

Finally, Table 4.3 provides a comparison of the running times (for comparable compression rates) between our approach and SQ and ZFP. ZFP has been designed to achieve high throughput and thus delivers the best time performances. The running times of our approach are on par with

Table 4.2 – *Wasserstein distance between the persistence diagrams of the original and decompressed data ($\epsilon = 1\%$). First line: ZFP 1bit/scalar, followed by a topology cleanup procedure. Second line: ZFP 1bit/scalar, augmented with our approach.*

Data-set	W_2					
	Combustion	Elevation	EthaneDiol	Enzo	Foot	Jet
ZFP + Cleanup	18.08	0.00	1.53	189.66	520,371	351.97
Topology-aware ZFP	13.73	0.00	0.40	131.11	506,714	153.45

Table 4.3 – *Time performance comparison between ZFP, SQ and our approach on* 512^3 *regular grids.*

Data-set	Time (s).			
	ZFP	SQ-R	SQ-D	Ours
Combustion	4.6	37.6	242.9	64.1
Elevation	7.2	31.4	204.2	25.3
EthaneDiol	4.7	34.4	197.1	63.0
Enzo	4.7	33.0	229.5	91.8
Foot	2.9	18.2	198.0	68.4
Jet	4.7	31.4	203.4	111.1
Random	4.1	31.6	182.7	307.7

other approaches enforcing strong guarantees on the decompressed data (SQ-R and SQ-D).

4.5.3 Application to post-hoc topological data analysis

A key motivation to our compression scheme is to allow users to faithfully conduct advanced topological data analysis in a post-process, on the decompressed data, with guarantees on the compatibility between the outcome of their analysis and that of the same analysis on the original data. We illustrate this aspect in this sub-section, where all examples have been compressed by our algorithm, without pointwise error control nor combination with ZFP.

We first illustrate this in the context of medical data segmentation with Fig. 4.10, which shows a foot scan. The persistence diagram of the original data counts more than 345 thousands pairs (top left). The split tree [Bre+11; CSA03] is a topological abstraction which tracks the connected components of sur-level sets of the data. It has been shown to excel at segmenting medical data [CSP04]. In this context, users typically compute multi-scale simplifications of the split tree to extract the most important features. Here, the user simplified the split tree until it counted only 5 leaves, corresponding to 5 features of interest (i.e. the 5 toes). Then, the

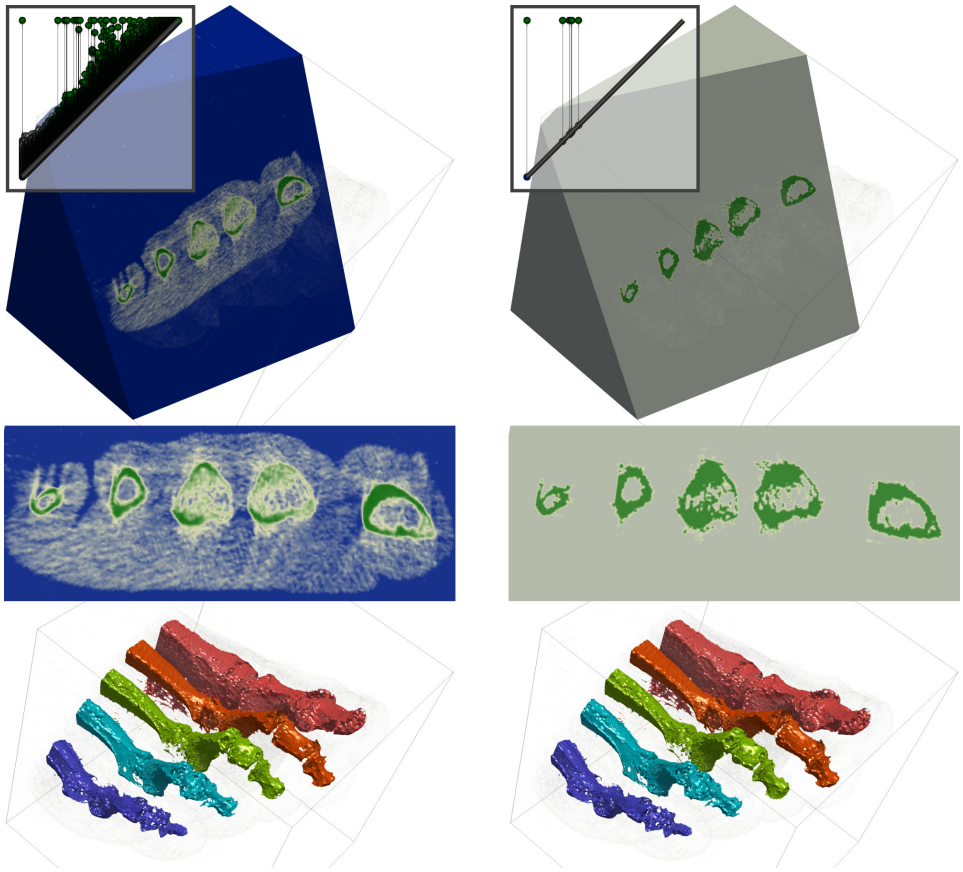

Figure 4.10 – *Topology driven data segmentation with multi-scale split trees, on the original data (left) and the data compressed with our approach (right). Top to bottom: persistence diagrams, sliced views of the data, output segmentations. The analysis yields compatible outcomes with and without compression, as shown with the bottom row, which exhibits identical segmentations (compression rate: 360).*

segmentation induced by the simplified split tree has been extracted by considering regions corresponding to each arc connected to a leaf. This results immediately in the sharp segmentation of toe bones. (Fig. 4.10, bottom left). We applied the exact same analysis pipeline on the data compressed with our approach. In particular, since it can be known a priori that this data has only 5 features of interest (5 toes), we compressed the data with a target persistence ϵ such that only 5 pairs remained in the persistence diagram (top right). Although such an aggressive compression greatly modifies data values, the outcome of the segmentation is identical (Table 4.4), for an approximate compression rate of 360.

Next, we evaluate our approach on a more challenging pipeline (Fig. 4.11), where features of interest are not explicitly related to the persistence diagram of the input data. We consider a snapshot of a simulation run of viscous fingering and apply the topological data analysis pipeline

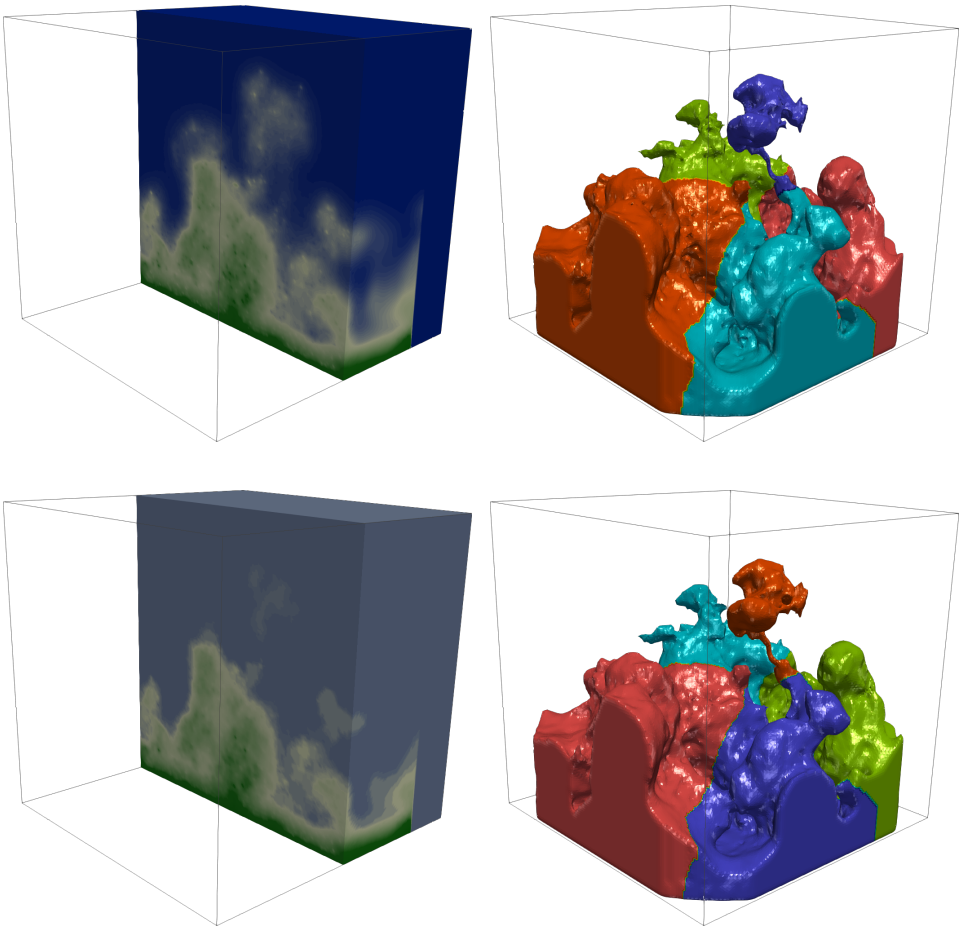

Figure 4.11 – *Topology driven data segmentation (right) on a viscous fingering simulation (left) on the original data (top) and the data compressed with our approach. Compatible fingers are extracted after compression (compression rate: 56).*

described by Favelier et al. [FGT16a] for the extraction and tracking of viscous fingers.

This pipeline first isolates the largest connected component of sur-level set of salt concentration. Next it considers its height function, on which persistent homology is applied to retrieve the deepest points of the geometry (corresponding to finger tips). Finally, a distance field is grown from the finger tips and the Morse complex of this distance field is computed to isolate fingers. In contrast to the previous example, the original data undergoes many transformations and changes of representation before the extraction of topological features. Despite this, when applied to the data compressed with our scheme, the analysis pipeline still outputs consistent results with the original data (Fig. 4.11, compression rate: 56). Only slight variations can be perceived in the local geometry of fingers, but their number is unchanged and their overall geometry compatible. Table 4.4 provides a quantitative estimation of the similarity between segmentations,

Table 4.4 – *Rand index between the outputs of a data segmentation pipeline based on topological methods (subsection 4.5.3), before and after compression, for several methods at compatible compression rates.*

Experiment	Rand index			
	Ours	SQ-R	SQ-D	ZFP
Foot scan	1.000	0.913	0.895	0.943
Viscous fingers	0.985	0.977	0.977	0.973

before and after compression with several algorithms. This table shows that our approach enables the computation of more faithful topological segmentations (higher rand index) compared to SQ and ZFP, which further underlines the superiority of our strategy at preserving topological features.

4.5.4 Limitations

Like all lossy techniques, our approach is subject to an input parameter that controls the loss, namely the persistence threshold ϵ above which features should be strictly preserved. While we believe this parameter to be intuitive, prior domain knowledge about the size of the features to preserve may be required. However, conservative values (typically 5%) can be used by default, as they already achieve high compression rates while preserving most of the features.

In some applications, ad-hoc metrics [CSP04] may be preferred over persistence. Our approach can be used in this setting too as the simplification algorithm that we use [TP12] supports an arbitrary selection of the critical points to preserve. However, it becomes more difficult then to express clear guarantees on the compression quality in terms of the bottleneck and Wasserstein distances between the persistence diagrams of the input and decompressed data.

When pointwise error control is enabled, the ∞-norm between the input and decompressed data is guaranteed by our approach to be bounded by $3\epsilon/2$. This is due to the topological simplification algorithm that we employ [TP12], which is a flooding-only algorithm. Alternatives combining flooding and carving [BLW12; TGP14] could be considered to reach a guaranteed ∞-norm of ϵ.

Finally, our approach only considers persistence pairs corresponding to critical points of index $(0, 1)$ and (d-1, d). However $(1, 2)$ pairs may have a practical interest in certain 3D applications and it might be interesting to enforce their preservation throughout compression. This would require

an efficient data reconstruction algorithm for $(1,2)$ pairs, which seems challenging [Att+13].

4.6 Summary

In this chapter, we presented the first compression scheme, to the best of our knowledge, which provides strong topological guarantees on the decompressed data. In particular, given a target topological feature size to preserve, expressed as a persistence threshold ϵ, our approach discards all persistence pairs below ϵ in order to achieve high compression rates, while *exactly* preserving persistence pairs above ϵ. Guarantees are given on the bottleneck and Wasserstein distances between the persistence diagrams of the input and decompressed data. Such guarantees are key to ensure the reliability of any post-hoc, possibly multi-scale, topological data analysis performed on decompressed data. Our approach is simple to implement; we provide a lightweight VTK-based C++ reference implementation.

Experiments demonstrated the superiority of our approach in terms of topological feature preservation in comparison to existing compressors, for comparable compression rates. Our approach can be extended to include pointwise error control. Further, we showed, with the example of the *ZFP* compressor [Lin14], how to make any third-party compressor become *topology-aware* by combining it with our approach and making it benefit from our strong topological guarantees, without affecting too much isosurface geometry. We also showed that, when aggressive compression rates were selected, our topological approach could improve existing compressors in terms of PSNR by cleaning up topological compression artifacts. We finally showed the utility of our approach by illustrating, qualitatively and quantitatively, the compatibility between the output of post-hoc topological data analysis pipelines, executed on the input and decompressed data, for simulated or acquired data sets. Our contribution enables users to faithfully conduct advanced topological data analysis on decompressed data, with guarantees on the size of missed features and the *exact* preservation of most prominent ones.

In the future, practical aspects of our algorithm could be improved, for in-situ deployment and to handle time-varying datasets. Runtime limitations should be investigated, with the objective to mitigate the effects of using (or not) a sequential topological simplification step, and to determine how many cores are necessary to outperform raw storage. A streaming version of the algorihm, which would not require the whole dataset to

be loaded at once would be of great interest in this framework. Finally, as our approach focuses on regular grids, a possible extension to the case of unstructured meshes ought to be investigated.

In the following chapter, we show a new tracking approach, demonstrating how persistence diagrams can be used to perform efficient feature tracking, and motivating further the utility of enforcing guarantees on the topological loss when performing data compression.

5 Fast and robust topology tracking

Contents

In this chapter, we increase the dimensionality of the targeted scientific data by considering *time-varying* scalar data. In this case, the I/O bottleneck is much more problematic, as a high temporal resolution would mean a high read/write throughput. One solution would be to decrease this temporal resolution, which would make later analyses, for example structure tracking, more difficult.

We then present, in this chapter, a robust and efficient method for tracking topological features in time-varying scalar data. Structures are tracked based on the optimal matching between persistence diagrams with respect to the Wasserstein metric. This fundamentally relies on solving the assignment problem, a special case of optimal transport, for all consecutive timesteps. Our approach relies on two main contributions.

First, we revisit the seminal assignment algorithm by Kuhn and Munkres which we specifically adapt to the problem of matching persistence diagrams in an efficient way. Second, we propose an extension of the Wasserstein metric that significantly improves the geometrical stability of the matching of domain-embedded persistence pairs. We show that this geometrical lifting has the additional positive side-effect of improving the assignment matrix sparsity and therefore computing time. The global framework computes persistence diagrams and finds optimal matchings in parallel for every consecutive timestep. Critical trajectories are constructed by associating successively matched persistence pairs over time. Merging and splitting events are detected with a geometrical threshold in a post-processing stage.

Extensive experiments on real-life datasets show that our matching approach is up to two orders of magnitude faster than the seminal Munkres algorithm. Moreover, compared to a modern approximation method, our approach provides competitive runtimes while guaranteeing exact results. We demonstrate the utility of our global framework by extracting critical point trajectories from various time-varying datasets and compare it to the existing methods based on associated overlaps of volumes. Robustness to noise and temporal resolution downsampling is empirically demonstrated. This contribution has been documented in the publication [Sol+18a].

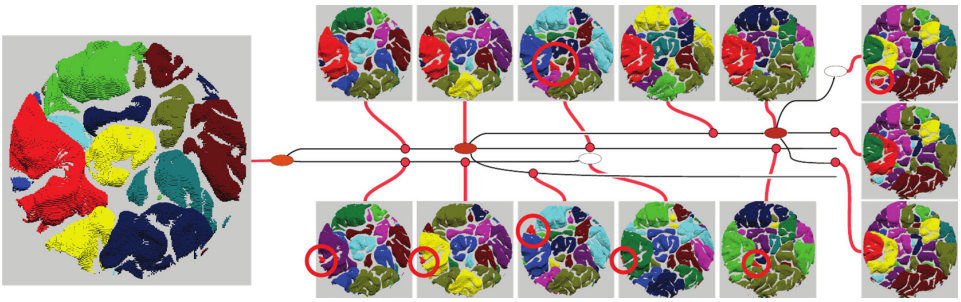

Figure 5.1 – *Topological segmentation of burning regions in a combustion simulation at multiple time-steps; then identification and tracking of these regions with a tracking graph. The burning cell displayed in red is shown to prorgressively split into smaller cells throughout the process. Image from [Bre+11].*

5.1 Scientific issues

Performing feature extraction and object tracking is an important topic in scientific visualization, for it is key to understanding time-varying data. Specifically, it allows to detect and track the evolution of regions of interest over time, which is central to many scientific domains, such as combustion (see Fig. 5.1), aerodynamics [HT94], oceanography [Rin+13] or meteorology [ZFL07]. With the increasing power of computational resources and resolution of acquiring devices, efficient methods are needed to enable the analysis of large datasets.

Topological data analysis has been used in the last decades as a robust and reliable setting for hierarchically defining features in scalar data [EH09]. In particular, its successful application to time-varying data [SB06; Bre+10] makes it a prime candidate for tracking. Both topological analysis and feature tracking have been applied *in-situ* [Zha+12; Lan+14], which demonstrates their interest in the context of large-scale data. Nonetheless, major bottlenecks of state-of-the art topology tracking methods are still the high required computation cost as well as the need for high temporal resolution.

In this chapter, we propose a novel feature-tracking framework, which correlates topological features in time-varying data in an efficient and meaningful way. It is the first approach, to the best of our knowledge, combining the setting of topological data analysis with optimal transport for the problem of feature tracking. More precisely, the key idea is to use combinatorial optimization for matching topological structures (namely, *persistence diagrams*) according to a fine-tuned metric. After exposing our formal setting (Sec. 5.2), we introduce an extension of the exact assignment algorithm by Kuhn and Munkres [KY55; Mun57] that we adapt in an effi-

cient way to the case of persistence diagrams (Sec. 5.3). We highlight the issues raised by the classical Wasserstein metric between diagrams, and propose a robust *lifted* metric that overcomes these limitations (Sec. 5.4). We then present the detailed tracking framework (Sec. 5.5). Extensive experiments demonstrate the utility of our approach (Sec. 5.6).

5.1.1 Related work

Our framework encompasses the definition, correlation and tracking of topological features in scalar fields. As such, it is related to topological data analysis of scalar fields, tracking techniques, the definition of metrics and combinatorial optimization.

Feature tracking

Topology has been used for feature extraction and tracking in the context of vector fields [Tri+02; Pos+03; Rei+12], mostly relying on stream lines, path lines [TS03; The+04; The+05; K S+06; Shi+09], or tracking punctual singularities [KE07]. For the latter, a forward streamline integration of critical points is performed in a specific scale space, which adapted for time-varying data would require knowledge about the evolution of the field, and for instance to compute time-derivatives.

For scalar data, features are defined based on attributes that are either geometric (isosurfaces, thresholded regions), or topological (contour trees, Reeb graphs). Similarly, tracking approaches either rely on geometrical (volume overlaps, distance between centers of gravity) or topological extracts (Jacobi set, segment overlap).

Geometrical approaches are based on thresholded connected components [SW99], glyphs [RPS01], cluster tracking [Gro+07], petri nets [Oze+14], or propose a hierarchical representation [GW11]. Similarly, the core methodology for associating topological features for tracking is often based on overlaps of geometrical domains along with other attributes [SW17; Bre+10; Bre+11; SB06; Sil95; SW96; SW97; SW98], on tracking Jacobi sets [EH04], or matching isosurfaces in higher-dimensional spaces [JS04; JSW03].

Such approaches usually test features in two consecutive timesteps against one another for potential overlaps, then draw the best correspondence between features according to some criterion. Typical criteria include optimizing the overlapping volume, mass, distance between centroids, or a combination of these [Sam+94; RPS01]. For this to work, the

temporal sampling rate of the underlying data must be such that features in two consecutive timesteps effectively overlap. This first criterion is thus not very robust to temporal downsampling.

Other approaches rely on global optimization [JS06], using the Earth Mover's distance [LB01] between geometrical features with various attributes such as centroid position, volume and mass. This does not, however, benefit from the natural definition of features offered by topological data analysis, nor from the possibility to simplify features in a hierarchical way. This is a real drawback in the context of noisy data as it implies dealing with large, computation-intensive optimization problems between every pair of timesteps.

Once features have been defined, and a methodology established to associate them in consecutive timesteps, the tracking representation is quite independent of whether geometrical or topological arguments have been used. Most often, graphs are used [RJR00; Wid+15; Lan+06], such as Reeb graphs [Web+11; Ede+04] and nested tracking graphs [Luk+17b]. Many popular graph structures are accounted for in [WT17]. An inconvenience of extracting rich tracking structures such as these without taking careful attention to potential noise is that it makes the interpretation quite difficult. In [Bre+11], the tracking graph is dense and intricate, making exploration impractical. It is therefore mandatory to do filtering and simplification in a post-processing stage, or to cleverly discard noisy events beforehand.

Assignment problems

Since we revisit the original algorithm by Kuhn and Munkres, we discuss here some related work in combinatorial optimization. The assignment problem is the discrete optimization problem consisting of finding a perfect matching of optimal cost in a weighted bipartite graph [Mun57; Ber98; BDM09]. In other terms, the problem is to find an optimal one-to-one correspondence between discrete entities (such as singularities in a scalar field at two different timesteps), with a cost associated to each possible correspondence. It can be solved with the seminal Kuhn-Munkres algorithm [Mun57]. The auction algorithm [Ber98; BC89; KMN17] is another popular approach for solving the assignment problem with a user-defined error threshold on the resulting assignment cost. In practice, this threshold is often set to 1% of the scalar range. A more general, continuous formulation of this problem is at the heart of Transportation theory [Mon81; Kan42; Vil08]. Modern techniques [Cut13] have attracted acute interest for

making this theoretical setup central to shape correlation [Sol+16] and interpolation [Sol+15], which do bear resemblance to feature tracking.

Metrics

Since we introduce a new metric for the matching of persistence diagrams, we discuss in the following existing metrics traditionally used in topological data analysis. The bottleneck and the interleaving distances have been widely investigated to study the stability of persistence diagrams. These metrics have been notably adapted in the context of kernel methods [Rei+15; CCO17] in machine learning. We discussed in chapter 3 the bottleneck distance, and the more general Wasserstein distance, applied to diagram points. The standard approach for computing this discrete Wasserstein metric relies on solving the associated assignment problem, either with an exact Kuhn-Munkres approach [Mor10; Wea13] or with an auction-based approximation [KMN17]. However, as discussed in Sec. 5.4, when these methods (metric-based [Cha+09; CEH05] or kernel-based [Rei+15; CCO17]) are applied as-is for tracking purposes, a high geometrical instability occurs which impairs the tracking robustness, as already observed in the case of *vineyards* [CEM06]. Our work (see Sec. 5.4) addresses this issue.

5.1.2 Contributions

This chapter presents the following contributions:

1. **Approach:** We present a sound and original framework, which is the first combining topology and transportation for feature tracking, comparing favorably to other state-of-the-art approaches, both in terms of speed and robustness.

2. **Metric:** We extend traditional topological metrics, for the needs of time-varying feature tracking, notably enhancing geometrical stability and computing time.

3. **Algorithm:** We extend the assignment method by Kuhn and Munkres to solve the problem of persistence matchings in a fast and exact way, taking advantage of our metric.

5.2 Preliminaries

This section gives some background on how the Wasserstein distance can be computated by solving the assignment problem, and presents an overview of our approach.

5.2.1 Assignment problem

The assignment problem is the problem of choosing an optimal assignment of n workers $w \in W$ to n jobs $j \in J$, assuming numerical ratings are given for each worker's performance on each job.

Given ratings $r(w_x, j_y)$ are summed up in a cost matrix (r_{xy}), finding an optimal assignment means choosing a set of n *independent* entries of the matrix so that the sum of these elements is optimal. *Independent* means than no two such elements should belong to the same row or column (i.e. no two workers should be assigned to the same job and no worker should be given more than one job). In other words, one must find a map $\sigma : W \to J$ of workers and jobs for which the sum $\sum_x (r(w_x, \sigma(w_x)))$ is optimal. There are $n!$ possible assignments, of which several may be optimal, so that an exhaustive search is impractical as n gets large.

Similarly, the unbalanced assignment problem is the problem of finding an optimal assignment of n workers to m jobs, where some jobs or workers might be left unassigned. This is the case of assignments between sets of persistence pairs; where costs are defined for leaving specific pairs unassigned.

The Hungarian algorithm [KY55; Mun57] is the first polynomial algorithm proposed by Kuhn to solve the assignment problem. It is an iterative algorithm based on the following two properties:

Theorem 1 *If a number is added or subtracted from all the entries of any one row or column of a cost matrix, then an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.*

This means that the cost matrix (r_{ij}) can be replaced with $(r_{ij}) - u_i - v_j$ where u_i (resp. v_j) is an arbitrary number which is fixed for the i^{th} row (resp. the j^{th} column).

Theorem 2 *If R is a matrix and m is the maximum number of independent zeros of R (i.e. number of entries valued at 0), then there are m lines (row or columns) which contain all the zeros of R.*

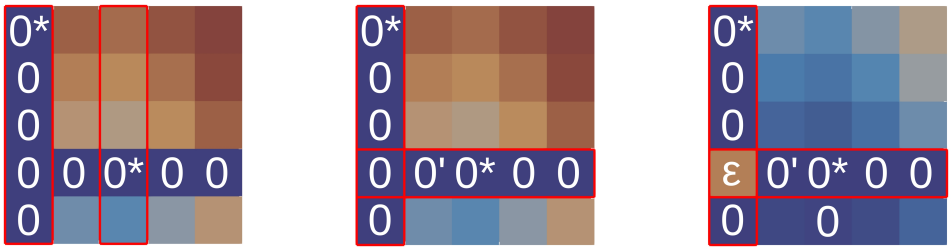

Figure 5.2 – *Matrix reduction phase. Subtracting the minimum element from each of the n rows and columns might not be sufficient to make a set of n* independent *zeros appear. In the above example, initially detected independent zeros are first starred. All columns containing a o* are then covered (left). An uncovered zero which has a o* in its row is found and primed; its row is covered and the column of the o* is uncovered (center). At this point, all zeros are covered by construction. Let ϵ be the smallest uncovered value. Add ϵ to every covered row; subtract ϵ from every uncovered column. This amounts to decreasing uncovered elements by ϵ and increasing twice-covered elements by ϵ. The sum of the elements of the matrix has been decreased and a new zero has appeared in an uncovered zone.*

This allows to determine whether an optimal assignment has been found and thus constitutes the stop criterion.

The algorithm iteratively performs additions and subtractions on lines and columns of the cost matrix, in a way that globally decreases the matrix cost, until the optimal assignment has been found, that is, until the matrix contains a set of $min(m, n)$ independent zeros.

In the remainder we consider the $O(min(m, n)^2 max(m, n))$ unbalanced Kuhn-Munkres algorithm [Mun57; BL71], an improvement over Kuhn's original version which follows the same principles, with an enhanced strategy for finding independent elements. The goal is always to reduce the cost matrix and find a maximal set of independent zeros. These independent zeros are marked with a *star*: they are candidates for the optimal assignment. Zeros which are candidates for being swapped with a starred zero are marked with a *prime*. Throughout the algorithm, rows and columns of the matrix are marked as *covered* to restrict the search for candidate zeros.

The algorithm can be seen as two alternating phases: a matrix reduction phase (Fig. 5.2) which makes new zeros appear, and an augmenting path phase (Fig. 5.3) which augments the number of marked (*starred*) independent zeros.

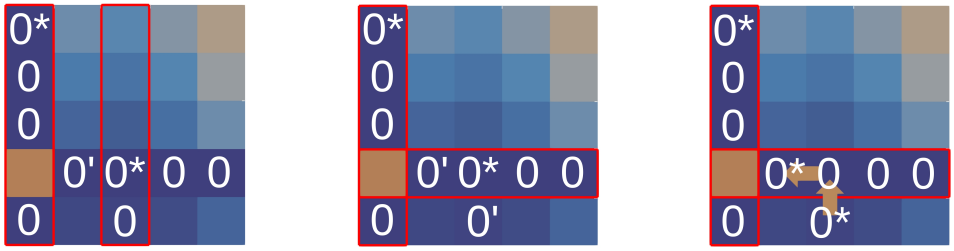

Figure 5.3 – *Augmenting path phase. After the first non-covered zero (left) is primed and covers updated, there is one non-covered zero Z_1 in the matrix (center), which is then primed. Let Z_2 be the o* in the column of Z_1 (if any), let Z_3 be the o′ in the row of Z_2 (there is one). Consider the series consisting of o* (Z_{2i}) and o′ (Z_{2i+1}) until it ends at a o′ that has no o* in its column. Unstar each o*, star each o′ of the series. The number of starred zeros has increased by one.*

5.2.2 Persistence assignment problem

The assignment problem for persistence pairs is very similar to the standard unbalanced assignment problem, except additional costs are defined for not assigning elements (*i.e.* matching persistence pairs with the diagonal). The assignment between diagrams P and Q then involves r_{ij} the numerical rating associated with assigning $p_i \in P$ with $q_j \in Q$, along with $r_{i,-1}$ (resp. $r_{-1,j}$) the numerical rating associated with matching p_i (resp. q_j) with the diagonal.

If P and Q are sets of persistence pairs such that $card(P) = n$ and $card(Q) = m$, then it is possible to solve the persistence assignment problem using the standard Kuhn-Munkres algorithm with the $(n+m) \times (n+m)$ cost matrix described by Eq. 5.1, as proposed in [Mor10]:

$$r_{ij} = \begin{cases} d_v(p_i, q_j) & \text{if } 0 < i \leq n, 0 < j \leq m \\ d_v(q_j, \text{diag}(q_j)) & \text{if } n < i \leq m+n, 0 < j \leq m \\ d_v(p_i, \text{diag}(p_i)) & \text{if } 0 < i \leq n, m < j \leq m+n \\ 0 & \text{if } n < i \leq m+n, m < j \leq n+m \end{cases} \tag{5.1}$$

The first line corresponds to matching pairs from P to pairs from Q; the second one corresponds to the possibility of matching pairs from P to the diagonal; the third one is for matching pairs of Q to the diagonal and the last one completes the cost matrix. The drawback of this approach is that it requires to solve the assignment problem on a $(n+m)^2$ cost matrix (that potentially contains two non-sparse blocks where persistence elements are located, see Fig. 5.4), though the number of distinct elements is at most $(n+1) \times (m+1)$. As seen in Sec. 5.3, our algorithm addresses this issue.

5.2.3 Overview

This section presents a quick overview of our tracking method, which is illustrated in Fig. 5.9. The input data is a time-varying PL scalar field f defined on a PL d-manifold $\mathcal{M}$ with $d \leq 3$.

1. First, we compute the persistence diagram of the scalar field for every available timestep.

2. Next, for each pair of two consecutive timesteps t and $t+1$, we consider the two corresponding persistence diagrams $\mathcal{D}(f_t)$ and $\mathcal{D}(f_{t+1})$. For each couple of persistence pairs $(p_i, q_j) \in \mathcal{D}(f_t) \times \mathcal{D}(f_{t+1})$, we define a distance metric corresponding to the similarity of these pairs: $d_v(p_i, q_j)$ (see Sec. 5.4).

3. For each pair of consecutive timesteps, we compute a *matching function* M. Every persistence pair p_i of $\mathcal{D}(f_t)$ is associated to to $M(p_i)$, which is either a persistence pair q_j in $\mathcal{D}(f_{t+1})$ or $\text{diag}(p_i)$ so as to minimize the total distance $\sum_i d(p_i, M(p_i))$. Finding the optimal M involves solving a variant of the classical Assignment Problem, as presented in Sec. 5.2.2. Only persistence pairs involving critical points of the same index are taken into account.

4. We compute tracking trajectories starting from the first timestep. If at timestep t the matching associates p_i with $M_t(p_i) = q_j$, then a segment is traced between p_i and q_j. If $M_t(p_i) = \text{diag}(p_i)$, the current trajectory ends. Trajectories are grown following this principle throughout all timesteps. Properties are associated to trajectories (time span, critical index), and to trajectory segments (matching cost, scalar value).

5. Finally, trajectories are post-processed to detect feature merging or splitting events with a user-defined geometric threshold.

5.3 Optimized persistence matching

This section presents our novel extension of the Kuhn-Munkres algorithm, which has been specifically designed to address the computation time bottleneck described in Sec. 5.2.1.

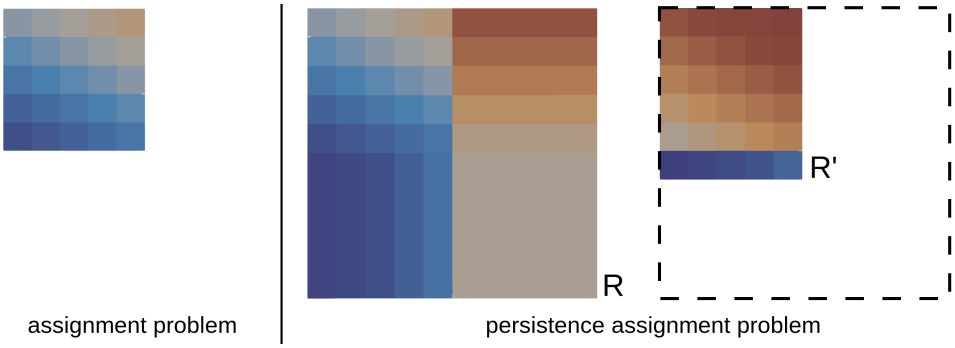

Figure 5.4 – *Cost matrices for a balanced assignment problem (left, $n \times n$ elements); for a persistence assignment problem with [Mor10] (center, R with $2n \times 2n$ elements – Eq. 5.1); and for the same persistence assignment problem with our proposed approach (right, R' with $(n+1) \times n$ elements – Eq. 5.2). Persistence elements in R induce two redundant non-sparse blocks (top-right and bottom-left).*

5.3.1 Reduced cost matrix

The classical persistence assignment algorithm based on Kuhn-Munkres considers R, a $(n+m)^2$ cost matrix. We propose to work instead with R', a reduced $(n+1) \times m$ matrix defined in Eq. 5.2, where every zero appearing in the last row is considered independent. This amounts to considering that persistence pairs corresponding to rows are not assigned by default. Fig. 5.4 summarizes the matrices considered by each assignment method.

$$r'_{ij} = \begin{cases} d_v(p_i, q_j) - d_v(\mathrm{diag}(p_i), p_i) & \text{if } 0 < i \leq n, 0 < j \leq m \\ d_v(\mathrm{diag}(q_j), q_j) & \text{if } i = n+1, 0 < j \leq m \end{cases} \tag{5.2}$$

This last row, emulating the diagonal blocks of Fig. 5.4-b requires a specific handling in the optimization procedure. In particular, it requires the first step of the algorithm to subtract minimum elements from columns (and not rows) so as not to have negative elements in the matrix.

As a reminder, the original algorithm proceeds iteratively in two alternating phases: matrix reduction that makes new zeros appear, and augmenting path that finds a maximal set of independent zeros. At the i^{th} iteration, the current maximal set of independent zeros is made of *starred* zeros. After a matrix reduction, new zeros appeared that can potentially belong to the new maximal set of independent zeros. Such candidates are *primed*. A single augmenting path (as in Fig. 5.3) replaces a set of n *starred* zeros with $n+1$ *primed* zeros, forming a new set of independent zeros with one more element. Rows and columns of the matrix are marked as *covered* to restrict the search for candidates in the augmenting path phase. Blue blocks of Algorithm 1 indicate our extension of the Kuhn-Munkres algorithm.

In this novel extension, an augmenting path constructed in the corresponding phase can start from a starred zero in the last row (and then potentially find a primed zero in its column), but such a path can never access a starred zero in the last row at another step, for the corresponding column would have been covered prior to this (and thus cannot contain a primed zero, see Algorithm 1). A starred zero in the last row can then never be unstarred.

The Kuhn-Munkres approach has the property to only increase row values (and only decrease column values). When our algorithm working on the reduced matrix R' ends, it is therefore not possible that the elements on the top-right corner of the corresponding full matrix R be negative. Furthermore, given Theorem 1, the resulting matrix corresponds to the same assignment problem.

5.3.2 Optimality

Working with the reduced matrix R', however, does not necessarily yield an optimal assignment. When assignments are found in the bottom row, if there has been additions to the matrix rows, then the corresponding R matrix would contain a top-right block that is not zero, and a top-left block that is not zero either. Thus, the stop criterion stated by Theorem 2 may not be respected when $k = \min(m, n)$ lines are covered (as the real number of independent zeros in R is $m + n$). Moreover, in our setup, a starred zero in the last column can never be unstarred; this is allowed in the approach on R, owing to the bottom-right block, initially filled with zeros.

We therefore use the criterion stated in Eq. 5.3 to ensure that if, at any given iteration of the algorithm, a zero is starred in the last row of column j, the cost of assigning the corresponding persistence pair to any other pair is higher than the cost of leaving both unassigned (0 for the j^{th}-column pair and the *residual value* ρ_i for i^{th}-row pairs – see Algorithm 1). This specificity is illustrated in Fig. 5.5.

$$\forall i \in [\![0, n]\!], r_{i,j} > \rho_i \Rightarrow r_{n+1,j} = 0^* \tag{5.3}$$

The (Eq. 5.3) criterion is checked whenever a zero appears on the last row after a subtraction is performed on a column by the algorithm. If it is observed, the corresponding column is removed from the problem and the persistence pair is set unassigned.

If the criterion is not respected, we have to report back the reduced problem onto the full matrix (missing banned columns and with reported

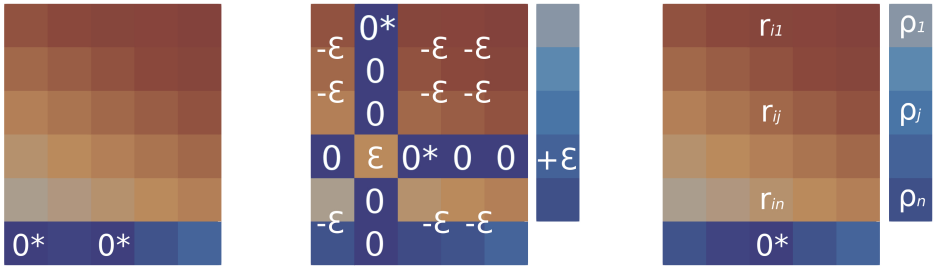

Figure 5.5 – *In our setup, every element in the last row is considered independent, so that it can contain multiple starred zeros (left). This emulates the behavior of the bottom-left matrix block in the classical approach. During an ϵ-reduction phase (center), we keep track of the (always positive) quantities that were added to matrix rows, hence increasing the top-right block in the classical approach, initially filled with only zeros. If a zero is starred in the last row and j^{th} column, let ρ_i be the sum of quantities added to row i throughout the algorithm (right). If for all i, $r_{ij} > \rho_i$, then the persistence pair associated with column i is assigned to the diagonal. If not (which never happened in our experiments), row residuals ρ_i and the equivalent residuals for columns ρ_j are used to report the partial optimization onto the matrix of the exact classical approach.*

found *residuals* ρ). For this, we need to keep track of residuals, that is, values that have been added or subtracted from each row and column throughout the course of the algorithm. Once these residuals have been reported onto the full matrix, there can be no negative element, and all of the optimization work has been reported (so that we do not start all over again from the beginning, but we start from the optimized output of the first phase).

In practice for persistence diagrams, we always observed that the first phase is sufficient to find an optimal assignment. Using this approach prevents from working with two potentially large blocks of persistence elements, typically occurring with the complete matrix for $i \in [\![n+1, m+n]\!]$ and $j \in [\![m+1, m+n]\!]$. This property is further motivated by the use of geometrical lifting (Sec. 5.4). The approach is detailed in Algorithm 1.

5.3.3 Sparse assignment

In practice, it is often observed that some assignments are not possible, and that reordering columns in the associated cost matrix would enable faster lookups and modifications [Cui+16], using sparse matrices. With persistence diagrams, the following simple criterion (Eq. 5.4) can be used to discard lookups for potential matchings.

$$d_v(p,q) > d_v(p, \mathrm{diag}(p)) + d_v(q, \mathrm{diag}(q)) \tag{5.4}$$

Algorithm 1: Our algorithm for sparse persistence matching. Blue sections allow to emulate the behavior of the three original non-sparse blocks on one single row, while ensuring optimality thanks to the residuals column. Black sections are common with the unbalanced Kunk-Munkres algorithm.

Data: $R' = (r_{ij})$, an $(n+1) \times m$ persistence cost matrix,
R the full $(n+m)^2$ matrix with non-sparse blocks.
Result: S a set of starred independent zeros
$\forall i, \rho_i \leftarrow 0$ // row residuals
$\forall j, \rho_j \leftarrow 0$ // column residuals
$B \leftarrow \emptyset$ // banned columns
Subtract the persistence element from every row and ρ_i
Transpose R' if $n > m$ and let $k = \min(m, n)$
Subtract the min element from every column of R' and ρ_j
Star independent zeros and cover their columns
while *number of covered columns* $< k$ **do**
    Find a non-covered zero Z_1 and prime it
    if Z_1 *is in the last row or there is no o* in its row* **then**
        Augmenting path phase (Fig. 5.3)
        Erase all primes, reset all covers
        Cover each column of containing a starred zero
    else
        Let Z_1' be the o* in the row of Z_1
        Cover this row and uncover the column of Z_1'
    if *there is no uncovered zero left* **then**
        Matrix ϵ-reduction phase (Fig. 5.2)
        $\rho_i \leftarrow \rho_i + \epsilon$ for modified rows i
        $\rho_j \leftarrow \rho_j - \epsilon$ for modified columns j
        if $\exists j | r_{n+1,j} = 0$ *and* $\forall i \in [\![1, n]\!], r_{i,j} > \rho_i$ **then**
            $r_{n+1,j}$ is starred
            $B \leftarrow B \cup j$
if $\exists j \notin B | r_{n+1,j} = 0^*$ *and* $\exists i | r_{i,j} < \rho_i$ **then**
    Kuhn-Munkres($R''_{ij} = R_{ij} + \rho_i + \rho_j$) with $j \notin B$.

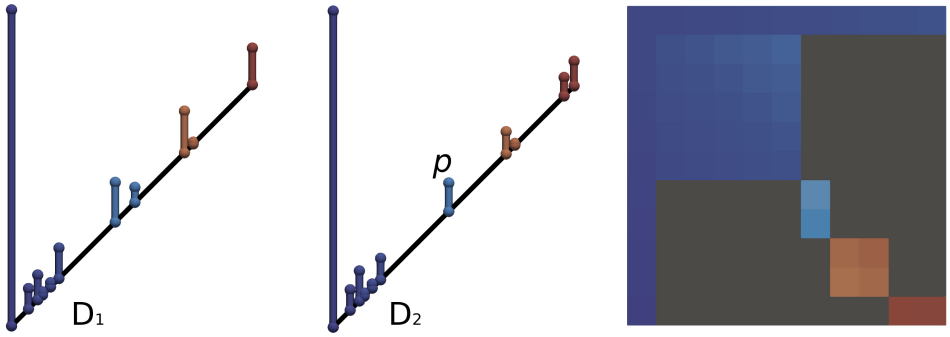

Figure 5.6 – *Persistence diagrams D_1 and D_2 showing in color small persistence pairs that will never be assigned in an optimal matching. The light blue pair $p \in D_2$ is such that $d(p, diag(p)) + d(q, diag(q)) < d(p, q)$ for any $q \in D_1$ which is neither light blue nor the first large persistence pair. This results in the cost matrix (right, D_1 pairs are rows and D_2 pairs are columns), where gray elements correspond to pairs (p, q) s.t. $d(p, q) > d(p, diag(p)) + d(q, diag(q))$.*

Working with our version of the Kuhn-Munkres algorithm then becomes interesting for many assignments verify Eq. 5.4 (Fig. 5.6), hence greatly reducing the lookup time for zeros, minimal elements, and the access time for operations performed on rows or columns.

On the contrary, the original full-matrix version of Kuhn-Munkres deals with non-sparse blocks which have to be accessed and modified constantly throughout the course of the algorithm.

5.4 Lifted persistence Wasserstein metric

This section highlights the limitations of the natural Wasserstein metric applied to time-varying persistence diagrams and presents an extension that enhances its geometrical stability. Geometrical considerations are motivated, in terms of accuracy and performance.

Persistence diagrams can be embedded into the geometrical domain (Fig. 3.14). Doing so, one easily sees how different embeddings can correspond to similar persistence diagrams in the *birth-death* space. Working in this 2D space does yield irrelevant matchings: as can be seen in Fig. 5.7, when only the *birth-death* coordinates of persistence pairs are considered, a matching can be optimal even if it happens between geometrically distant zones. As a consequence, the distance metric between persistence pairs must be augmented with geometrical considerations.

To address this, we propose instead of d_v (Eq. 3.2) to use the distance defined in Eq. 5.5:

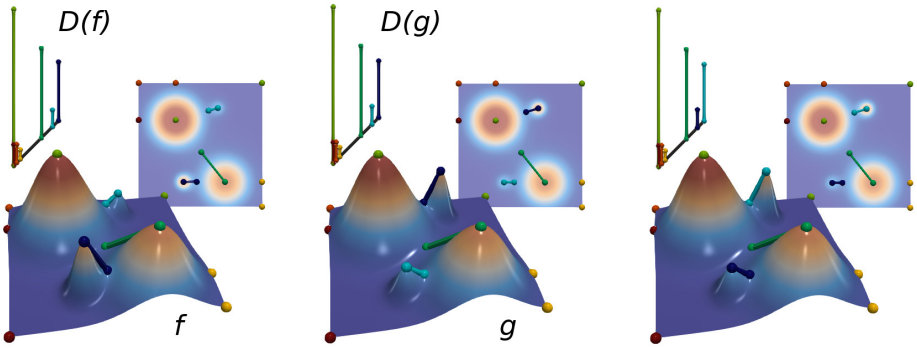

Figure 5.7 – *Scalar field f with persistence diagram $D(f)$ (left), matched with a scalar field g with a similar persistence diagram $D(g)$, but an embedding that swaps the position of the light blue pair with that of the dark blue pair. Matched pairs are displayed with the same color using the non-geometric (center) and geometric Wasserstein metric (right). The latter takes the geometrical embedding into account, preventing similar pairs (regarding persistence) to be assigned if they are geometrically distant.*

$$d_{lift,\nu}(p,q) = (\alpha\delta_{birth}^{\nu} + \beta\delta_{death}^{\nu} + \gamma_1\delta_x^{\nu} + \gamma_2\delta_y^{\nu} + \gamma_3\delta_z^{\nu})^{1/\nu} \tag{5.5}$$

where δ_x, δ_y and δ_z correspond to geometric distances between the extrema involved in the persistence pairs on a given axis. We process diagonal projections as follows (Eq. 5.6):

$$\begin{aligned} d_{lift,\nu}(p,\mathrm{diag}(p)) =& (\alpha\left|p_x\right|^{\nu} + \beta\left|p_y\right|^{\nu} + \\ & \gamma_1(\delta_x^p)^{\nu} + \gamma_2(\delta_y^p)^{\nu} + \gamma_3(\delta_z^p)^{\nu})^{1/\nu} \end{aligned} \tag{5.6}$$

where the terms δ_x^p, δ_y^p and δ_z^p correspond to the geometric distance between the critical points of a given pair p. Intuitively, it accounts for the distance between the critical points to cancel.

A *lifted* distance is considered by augmenting the geometric distance with coefficients α, β, γ_i. This aims at giving more or less importance to the birth, death or some of the x, y, z coordinates during the matching, depending on applicative contexts. For instance, in practice it is desirable to give less importance to the birth coordinate when dealing with d-$(d-1)$ persistence pairs (in other words, for tracking local maxima, see Fig. 5.8). For the remainder of the chapter and the experiments, we used $(\alpha, \beta, \gamma_i) = (0.1, 1, 1)$ for maxima and $(\alpha, \beta, \gamma_i) = (1, 0.1, 1)$ for minima, for normalized geometrical extent and scalar values. We observed that using a lifted metric further increases the cost matrix sparsity, resulting in extra speedups.

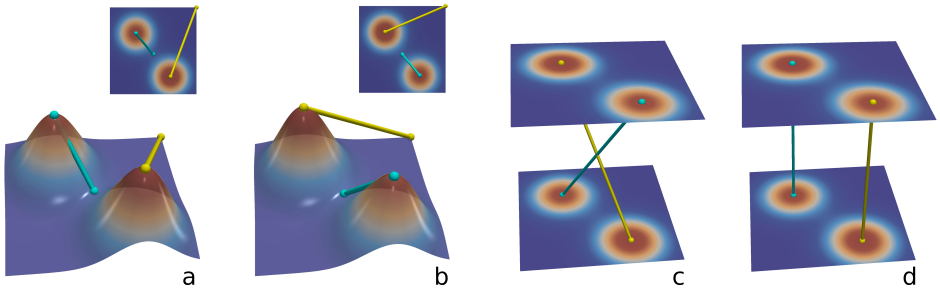

Figure 5.8 – *Lifting the birth coordinate. 2D scalar fields with two gaussians (a, b), where the bottom (resp. top) gaussian has the maximum value (a) (resp. b). Using the geometrical metric alone (c) is not sufficient, as the* birth *coordinate p_x misleadingly equalizes the persistence term of pairs of the same color in (a, b): $\delta^a_{p,yellow} = \delta^b_{p,yellow}$, $\delta^a_{p,blue} = \delta^b_{p,blue}$, potentially overcoming the geometrical factor. Lifting the birth coordinate with a small coefficient for associating maxima yields the correct matching (d).*

5.5 Feature tracking

This section describes the four main stages of our tracking framework, relying on the discussed theoretical setup. Without loss of generality, we assume that the input data is a time-varying 2D or 3D scalar field defined on a PL-manifold. Topological features are extracted for all timesteps (Fig. 5.9, a-b), then matched (Fig. 5.9, c); trajectories are built from the successive matchings (Fig. 5.9, d) and post-processed to detect merging and splitting events.

5.5.1 Feature detection

First, we compute persistence diagrams for each timestep. We propose using the algorithm by Gueunet et al. [Gue+17], in which only 0-1 and d-$(d-1)$ persistence pairs are considered.

When the data is noisy, it is possible to discard pairs of low persistence (typically induced by noise) by applying a simple threshold. In practice, this amounts to only considering the most prominent features. Using such a threshold accelerates the matching process, where for approaches based on overlaps, removing topological noise would require a topological simplification of the domain (for example using the approach in [TP12]), which is computationally expensive.

5.5.2 Feature matching

If P_1, P_2 are two sets of persistence pairs taken at timesteps t and $t+1$, then we use the algorithm described in Sec. 5.3, with the appropriate distance

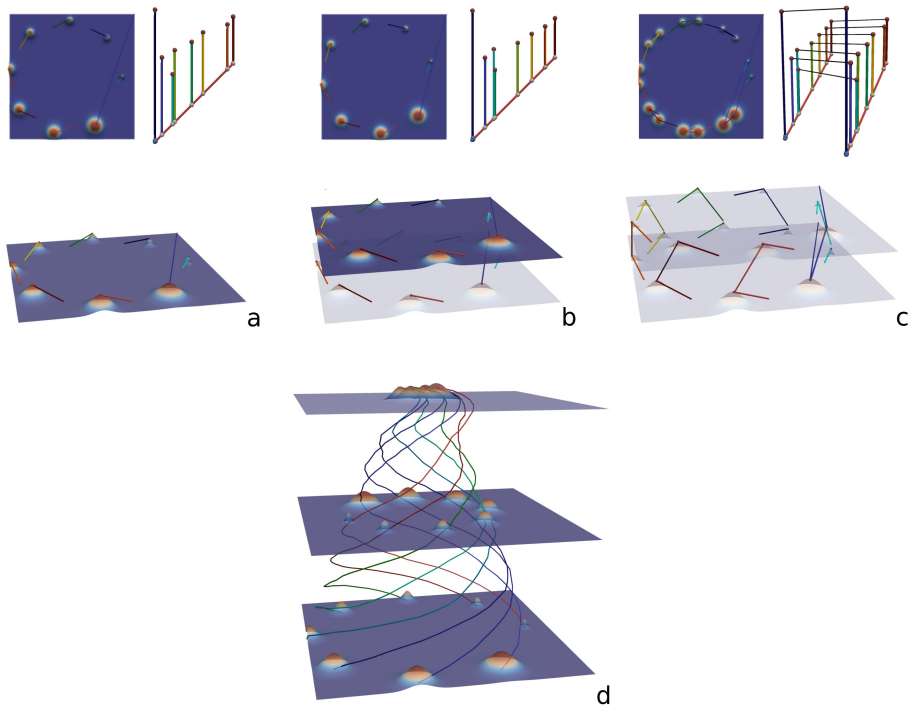

Figure 5.9 – *Overview of our tracking approach on a dataset consisting of eight whirling gaussians: persistence diagram computations for two consecutive timesteps (a) and (b); matching of persistence pairs of two timesteps (c), propagation of matchings and construction of a trajectory (d).*

metric, as discussed in Sec. 5.4, to associate pairs in P_1 and P_2. A given pair $p_1 \in P_1$ might be associated to one pair $p_2 \in P_2$ at most, or not associated, and symmetrically.

5.5.3 Trajectory extraction

Trajectories are constructed by simply attaching successively matched segments. For all timesteps t, if the feature matching associates p_i with $M_t(p_i) = q_j$, then a segment is traced between q_j and p_i, and is potentially connected back to the previous segment of p_i's trajectory. If $M_t(p_i) = \text{diag}(p_i)$, the current trajectory ends. Properties are associated to trajectories (time span, critical point index) and to trajectory segments (matching cost, scalar value, persistence value, embedded volume).

5.5.4 Handling merging and splitting events

Given a user-defined geometrical threshold ϵ, we propose to detect events of *merging* or *splitting* along trajectories in the following manner. If $T_1, T_2 : I \subset \mathbb{N} \rightarrow \mathbb{R}^3$ are two trajectories spanning throughout $[t_i, t_{i+n}]$ and $[t_j, t_{j+m}]$ respectively, and if for some $k \in [i, i+n] \cap [j, j+m]$,

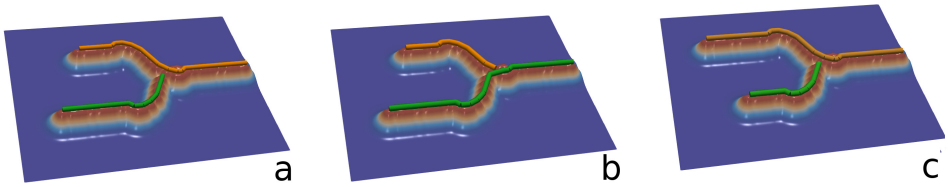

Figure 5.10 – *Merging process. Tracking is performed on two gaussians moving from left to right (a). The post-process detects a merging event with a geometrical threshold, then propagates the component identifier of the oldest component (b) and properly reconnects the matching segment. If the oldest component has already the right identifier (c) nothing is done. This process is proposed by analogy with building persistence diagrams.*

$d_{lift,\nu}(T_1(t_k), T_2(t_k)) < \epsilon$, where $d_{lift,\nu}$ is a lifted distance, then an event of merging (or splitting) is detected. We consider that a merging event occurs between T_1 and T_2 at time k, when neither T_1 nor T_2 start at t_k. We then consider that the *oldest* trajectory takes over the *youngest*. For example, T_1 and T_2 meet (according to the ϵ criterion) at t_k the last timestep of T_2, and T_2 started *before* T_1, then we disconnect the remainder of T_1 from the trajectory before t_k and we connect it so as to continue T_2 until T_1's original end. Similarly, a splitting event occurs between T_1 and T_2 at time k, when neither T_1 nor T_2 end at t_k. The process is illustrated in Fig. 5.10. It is done separately for distinct critical point types: minima, maxima and saddles are not mixed.

5.6 Results

This section presents experimental results obtained on a desktop computer with two Xeon CPUs (3.0 GHz, 4 cores each), with 64 GB of RAM. We report experiments on 2D and 3D time-varying datasets, that were either simulated with Gerris [Popo3] (von Kármán Vortex street, Boussinesq flow, starting vortex), or acquired (Sea surface height, Isabel hurricane). Persistence diagrams are computed with the implementation of [Gue+17] available in the Topology ToolKit [Tie+17]; the tracking is restricted to 0-1 and $(d-1)$-d pairs. We implemented our matching (Sec. 5.3) and tracking approaches (Sec. 5.5) in C++ as a Topology ToolKit module.

5.6.1 Application to simulated and acquired datasets

We applied our tracking framework to both simulated and acquired time-varying datasets to outline specific phenomena.

In Fig. 5.11, we present the results of the tracking framework applied

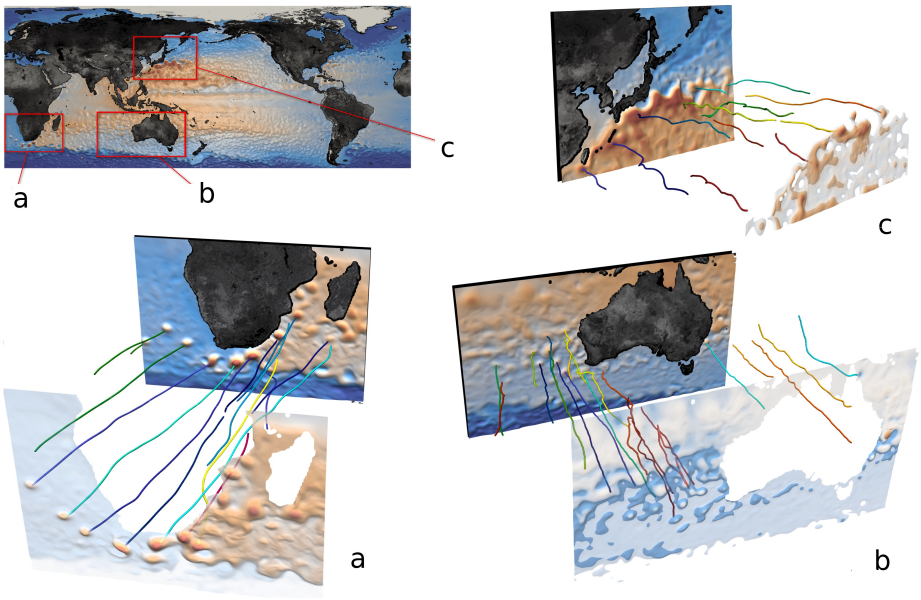

Figure 5.11 – *Sea surface height (SSH) captured over 365 days, 1 timestep every day. Local maxima are tracked in the region corresponding to the Agulhas Current, near South Africa (a); it is observed they are slowly drifting towards the west. SSH maxima are also drifting west in the less contrasted zone of the West Australian Current (b). Tracking in the region of the Kuroshio Current, near Japan (c), demonstrates a whirling behavior of local maxima.*

to an oceanographic dataset. The scalar field (sea surface height) is defined on 365 timesteps on a triangular mesh. We can see interesting trajectories corresponding to well-known oceanic currents. Drifting (a, b) and turbulent behaviors (c) of local extrema are highlighted. In Fig. 5.12, tracking is performed on the vorticity of highly unstable Boussinesq flow. Thanks to our analysis, trajectories can be filtered according to their temporal lifespan, revealing clearly different trajectory patterns among the turbulent features. This kind of analysis may be easily performed based on other trajectory attributes, depending on applicative contexts. In Fig. 5.13, we show our approach on a 3D hurricane dataset whose temporal resolution is such that a method based on overlaps of split-tree leaves (see Sec. 5.6.3) could not extract trajectories. In Fig. 5.14, our tracking framework correctly follows local extrema of the vorticity field in a simulated vortex street.

5.6.2 Tracking robustness

In the following two sections, we demonstrate the robustness and performance of our tracking framework.

We compare to the greedy approach based on the overlap of volumes [Bre+10; Bre+11; SB06; SW17] of split-tree leaves, which amounts to track-

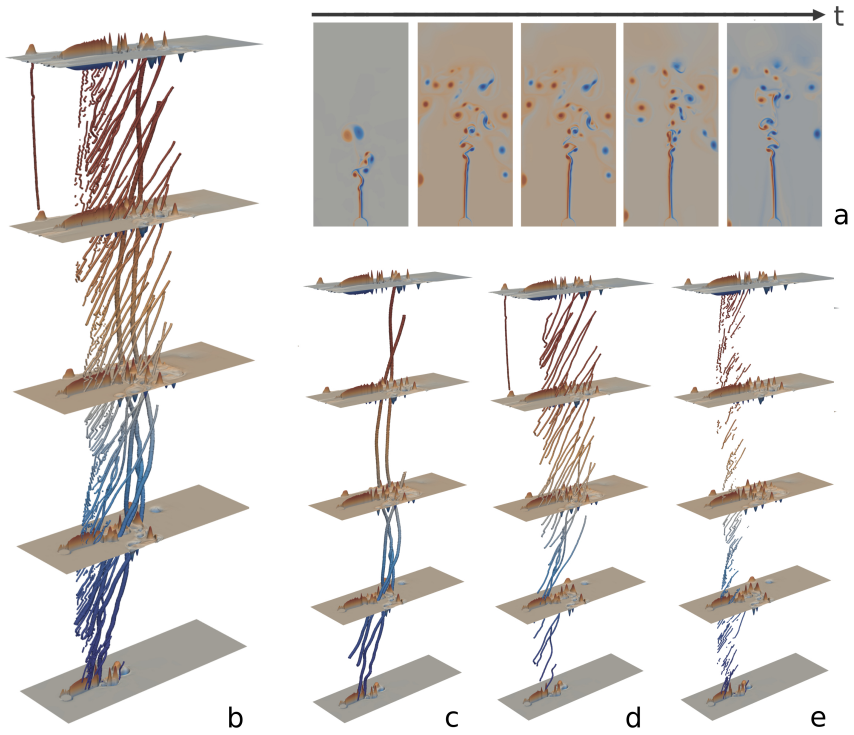

Figure 5.12 – *Boussinesq flow generated by a heated cylinder (a). Feature tracking is performed (b) on the fluid vorticity. Some vortices exist over a long period of time (c), as others vanish more rapidly (d), sometimes akin to noise (e). Feature trajectories can easily be filtered from their lifespan.*

ing local maxima. In this approach, for every pair of consecutive timesteps $(t, t+1)$, split-tree segmentations S_t and S_{t+1} are computed (these are a set of connected regions). Overlap scores are then computed for every pair of regions $(s_i, s_j) \in S_t \times S_{t+1}$, as the number of common vertices between s_i and s_j. Scores are sorted and s_i is considered matched to the highest positive scoring s_j such that s_j has not been matched before. Trajectories are extracted by repeating the process for all timesteps.

The robustness of our tracking framework is first assessed on a synthetic dataset consisting of whirling gaussians, on which we applied noise (Fig. 5.15). Identified trajectories are sensibly the same with a perturbation of 10% of the scalar range. The 75% most important features are still correctly tracked after a 25% random perturbation has been applied to the data.

In Fig. 5.14, our method is compared to the greedy approach, based on overlaps, while decreasing the temporal resolution. The overlap approach yields trajectories corresponding to noise (Fig. 5.14-e), which can be filtered by applying topological simplification beforehand (this would

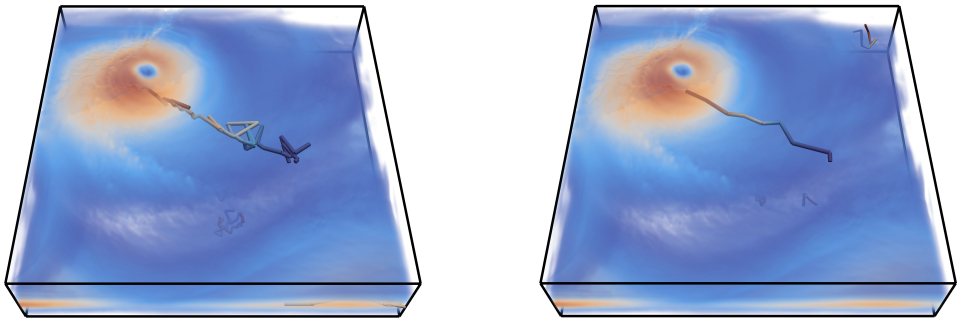

Figure 5.13 – *Tracking performed on the wind velocity on a 3D Isabel hurricane dataset before (left), and after (right) temporal downsampling (1 frame every 5 timesteps). The global maximum is tracked successfully despite the high instability displayed by the scalar field.*

have a significant computational cost as it requires to modify the original function), or by associating the scalar value of the function to every point in the trajectory and then filtering the trajectory in a post-process step. In our setup, is is much simpler to discard this noise, by using a threshold for discarding small persistence pairs before the matching (implying a faster matching computation). When downsampling the temporal resolution to only 20% of the timesteps, our approach still gives the correct results (Fig. 5.14, d vs. e). With 15% of the timesteps, our approach (Fig. 5.14-f) still agrees with the tracking on the full-resolution data (Fig. 5.14-b), until preceding features begin to catch up, resulting in a zig-zag pattern. By comparison, the overlap method fails to correctly track meaningful regions from the beginning of the simulation to its end; it is indeed dependent on the geometry of overlaps, which is unstable. It can be argued that the locality captured by overlaps is emulated in our framework by embedding and lifting the Wasserstein metric, when the overlap method does not take persistence into account when matching regions. Also note that if the saddle component of persistence pairs associated to maxima is ignored (i.e. if $\alpha = 0$ in Eq. 5.5 and Eq. 5.6) during the matching, then the geometrical distance can be insufficient for correctly tracking these persistence pairs (c). Therefore, the problem of matching persistence pairs for tracking topological features cannot be reduced to the (unbalanced) problem of assigning critical points in 4 dimensions (3 for the geometrical extent, one for the scalar value).

Fig. 5.13 further illustrates the robustness of our approach when downsampling the data temporal resolution. In hurricane datasets, local maxima can be displaced to geometrically distant zones between timesteps if those are taken at multiple-day intervals. This unstable behavior and

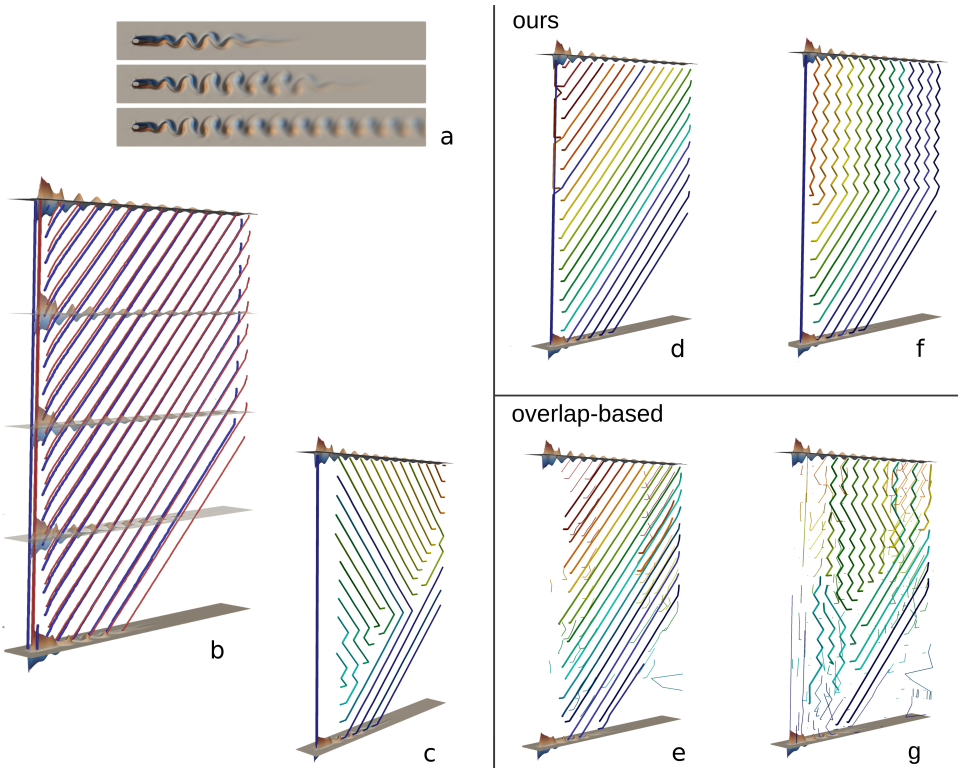

Figure 5.14 – *Simulated von Kármán vortex street (a), on which minima and maxima of the vorticity are tracked with our approach and 1% persistence filtering (b). Only taking the geometry and scalar value into account while doing the matchings (i.e. completely ignoring the* birth *in the lifted metric), is not sufficient to correctly track features (c). Maxima only are tracked considering 1 frame every 5 timesteps (d). With the same temporal resolution, the overlap-based approach (e) does capture small trajectories corresponding to noise, displayed with thinner lines, that have to be filtered for instance using topological simplification [TP12]. Considering 1 frame every 7 timesteps (f) still yields correct trajectories up to the point where, every other frame, optimal matchings for the metric are between a feature and the preceding one, due to features traveling fast. The overlap approach (g) is less stable in this case as it extracts erroneous trajectories from the very first stages of the simulation to the end.*

the absence of obvious overlaps makes it particularly difficult to track extrema; nevertheless, our framework managed to track them at a very low time-resolution.

5.6.3 Tracking performance

We then compare our framework with our implementation of the approach based on overlaps [Bre+11] on the ground of performance. Figures are given in Tab. 5.1. Note that our approach has the advantage of taking persistence diagrams as inputs, so it can be applied to unstructured or time-varying meshes, for which overlap computations are not trivial. Our

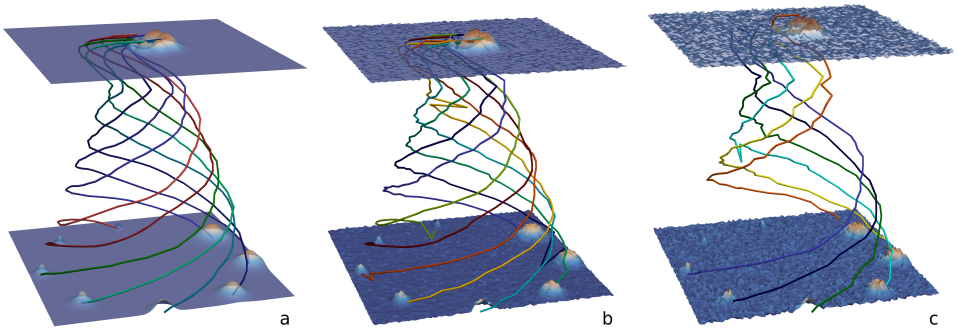

Figure 5.15 – *Lifted Wasserstein tracking performed on a set of whirling 2D gaussians (a). With noise accounting for 10% of the scalar range (b), feature trajectories are still correctly detected. For 25% noise (c), 75% of the features (namely, the 6 most prominent out of the initial 8) are still correctly tracked despite heavy perturbations.*

Table 5.1 – *Time performance comparison (CPU time in seconds) between the approach based on overlaps of volumes [Bre+11] and our lifted Wasserstein approach. Tracking is performed over 50 timesteps, on structured 2D (Boussinesq, Vortex street), structured 3D (Isabel), and unstructured 2D (Sea surface height) meshes. The pre-processing step (FTM) computes the topology of the dataset. The post-processing step extracts the tracking mesh, computes its attributes, and handles splitting and merging events. We observe a parallel speedup ranging from 4 to 6 for our approach on 8 threads (FTM and matching phases).*

Data-set	Pre-proc (s)	Matching (s)		Post-proc
	FTM	[Bre+11]	ours	(s)
Boussinesq	116	75	18	4.7
Vortex street	45	23	18	2.8
Isabel (3D)	863	>3k	17	162
Sea height	568	N.A.	277	113

approach is also relatively dimension-independent: though in 3D, computing overlaps is very time-consuming (Fig. 5.1-Isabel), the complexity of the Wasserstein matcher, which only takes embedded persistence diagrams as inputs, for a given number of persistence pairs is sensibly equivalent. For both Isabel and Sea surface height datasets, we applied a 4% persistence filtering on input persistence diagrams. As the experiments show, our approach is faster in practice than the overlap method with best-match search.

Table 5.2 – *Time performance comparison between the state-of-the-art Munkres-based approach [Wea13], and our modified sparse approach.*

Data-set	Sizes of diagrams	Time (s)	
		[Wea13]	ours
Starting vortex	473 – 489	68.6	1.26
Isabel	465 – 413	72.2	3.58
Boussinesq	1808 – 1812	11.1k	102
Sea height	1950 – 5884	26.5k	155

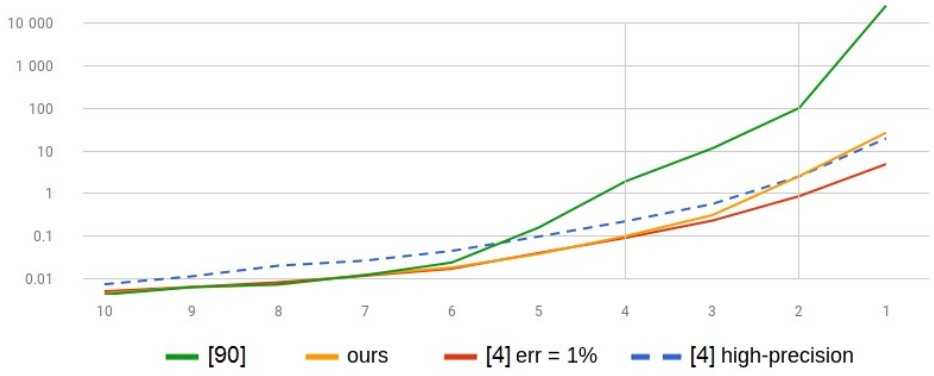

Figure 5.16 – *Running times in seconds of different matching approaches, for decreasing persistence thresholds expressed in percentage of the scalar range. The initial two diagrams containing 14,082 and 14,029 pairs are filtered to remove pairs whose persistence is less than the defined threshold, then a matching is computed with our new method, the reference exact method [Wea13], the approximate method [BC89], first with 1% accuracy, then with an accuracy of* 10^{-4}*% of the scalar range.*

5.6.4 Matching performance

Next, we compare the performance of the matching method we introduced in Sec. 5.3 to two other state-of-the art algorithms.

We compare it to the *reference approach* for the exact assignment problem [Wea13] based on the Kuhn-Munkres algorithm, and to our implementation of the *approximate approach* based on the auction algorithm [BC89; KMN17], on the ground of performance.

Tab. 5.2 shows that our new assignment algorithm is up to two orders of magnitude faster than the classical exact approach [Wea13]. In particular, the best speedups occur for the larger datasets which indicates that our approach also benefits from an improved scaling.

It is often useful in practice to discard low-persistence pairs prior to any topological data analysis as these correspond to noise. Fig. 5.16 compares the running times of our approach, [Wea13] and [BC89] as more and

more low-persistence pairs are taken into account. When removing pairs whose persistence is below 5% of the scalar range, which is commonly accepted as a conservative threshold, our approach is faster than all competing alternatives. When considering more low-persistence features, below 4%, our approach is competitive with the approximated auction approach with 1% error. Below 2%, only noise is typically added in the process. The performance of our algorithm becomes comparable to that of the high-precision auction approximation although our approach guarantees exact results.

5.6.5 Limitations

As we described, our framework enables the tracking of 0-1 and d-$(d-1)$ persistence pairs. It would be interesting to extend it to support the tracking of saddle-saddle pairs (in 3D) and see its application to meaningful use cases.

Besides, the *lifting* coefficients proposed in our metric (Eq. 5.5) might be seen as supplementary parameters that have to be tuned according to the dataset and applicative domain. Nonetheless, we observed in our experiments that these parameters do not require fine-tuning to produce meaningful tracking trajectories. The extent to which these can be enhanced by fine-tuning is left to future work.

The lifted distance can be generalized to take other parameters, such as the geometrical volume, mass, feature speed, into account, and be fine-tuned to answer the specificity of various scientific domains. Merging and splitting might also be enhanced, or given more flexibility, for instance with additional criteria. We also believe that the performance of the post-processing phase can be improved.

Additionally, we believe that the approximate auction algorithm can also take the lifted persistence metric into account by performing Wasserstein matchings between persistence pairs in 5 dimensions, and possibly benefit from geometry-based lookup accelerations, as suggested in lower dimension in [KMN17]. It remains to be clarified how the quality of the matchings is affected in practice by using an approximate matching method, and how sparsity can enhance the research phase for the auction algorithm.

We note that the theoretical complexity of our matching method is, as the Munkres method, cubic; however, the two orders of magnitude speedups demonstrated in our experiments allow to study more challeng-

ing datasets. For very large case studies, the use of persistence thresholds could prove quite helpful for controlling the computing time of matchings. Among other non-trivial tracking methods, some graph matching methods are based on *graph-edit distances* [Gao+10; Bek+14]. Their adaptation to the case of persistence diagrams or other topological structures (such as contour trees and Reeb graphs) may enable an additional structural regularization, this ought to be investigated in future work.

5.7 Summary

In this chapter, we presented an original framework for tracking topological features in a robust and efficient way. It is the first approach combining topological data analysis and transport for feature tracking. As the kernel of our approach, we proposed a sparse-compliant extension of the seminal assignment algorithm for the exact matching of persistence diagrams, leveraging in practice important speedups. We introduced a new metric for persistence diagrams that enhances geometrical stability and further improves computation time. Overall, in comparison with overlap-based techniques, our approach displays improved performance and robustness to temporal downsampling, as experiments have shown.

We released the implementation of our tracking framework open-source as a part of TTK [Tie+17]; we hope that it will be useful to the community with an interest for efficient tracking methods. We look forward to adapting it to tracking phenomena in *in-situ* contexts, where the large-scale time-varying data is accessed in a streaming fashion. As we are also interested in larger datasets, industrial partners are currently carrying out scaling tests on complex physical case studies for which one needs specifically adapted rendering techniques [Luk+17b] to apprehend the resulting graphical complexity of the topology evolution.

We also believe that the application potential of our matching framework can be studied for tasks other than time-tracking, for instance, self-pattern matching and symmetry detection [TN14]. As we will see in the next chapter, one particularly promising application for the newly developed metrics is feature comparison in ensemble data.

6 Application to parameter fitting in ensembles

Contents

In this chapter, we further increase the dimensionality of the targeted data, by considering a parametric study, that is, a survey involving a number of simulation runs obtained with varying model parameters, where a single run is an independent time-varying scalar data-set. In such studies, pertinent analyses leveraging the potential of all available data are difficult, notably because the I/O problematic becomes a major bottleneck, which reduces the range of possible analyses. A typical analysis

scientists would be interested in is assessing the quality of a simulation run, compared to the ground truth.

As in our applicative context, we are specifically interested in fluid flow in porous media, we present, in this chapter, a novel framework, based on topological data analysis, for the automatic evaluation and ranking of *viscous finger* simulation runs in an ensemble with respect to a reference aquisition. Individual fingers in a given time-step are associated with critical point pairs in the distance field to the injection point, forming persistence diagrams. Different metrics, based on optimal transport, for comparing time-varying persistence diagrams in this specific applicative case are introduced.

We evaluate the relevance of the rankings obtained with these metrics, both qualitatively thanks to a lightweight web visual interface, and quantitatively by studying the deviation from a reference ranking suggested by experts. Extensive experiments show the quantitative superiority of our approach compared to traditional alternatives. Our web interface allows experts to conveniently explore the produced rankings.

We show a complete viscous fingering case study demonstrating the utility of our approach in the context of porous media fluid flow, where our framework can be used to automatically discard physically-irrelevant simulation runs from the ensemble and rank the most plausible ones. We document an in-situ implementation to lighten I/O and performance constraints arising in the context of parametric studies. This contribution has been documented in the submitted manuscript [Sol+19].

6.1 Scientific issues

The chaotic nature of fluid flows makes it difficult to account for the propagation of initial uncertainties in numerical models, or uncertainties in model parameters. To predict uncertain phenomena, thanks to the increase in computing power in recent years, Monte Carlo methods have been broadly used, for instance in climate modeling, forecasts, statistical physics, chemistry and astrophysics. The idea is to compute a large number of simulations, called an *ensemble*, while densely sampling the space of input parameters. A *post-mortem* comparison (i.e. performed *after* simulations have been completed) to experimentally acquired data can then determine which simulations produced the most realistic outcomes and how input parameters affect their variability.

Specifically, in reservoir engineering, an area of petroleum engineering concerned with fluid flow through porous media, it is important to quantitatively predict well productions, i.e. the quantity of oil that can be extracted, in order to estimate the available reserves and to design surface facilities. Numerical models are subject to parameter uncertainties, and can be tuned by launching randomly sampled ensemble simulations. Usually, reference production rates and well pressures are history-matched with the simulated ensembles, which ideally would allow domain experts to restrict the space of input parameters. This history match procedure is usually applied at the field scale (oil and gas reservoirs), but also at the core scale (a few decimeters) when lab engineers want to match the behavior of experimental corefloods.

In practice, notably in the domain of Darcy-type simulations at the core scale, production and pressure data is not sufficient to infer model parameters. Further measuring tools have been recently integrated in lab experiments in order to constrain the parameter space, by monitoring the *saturation* scalar fields through X-rays, so as to obtain information on phase velocities and residual saturations. Here the saturation measures the volume fraction of a given phase in the geometrical domain. Observing these scalar fields seems relevant when the fluid behaves in a particularly chaotic way, so that simulations which are not physically adequate could be detected. The case of the viscous fingering phenomenon, an instability which occurs at the interface between two fluids of distinct viscosity in porous media, is of particular interest.

In this context, for all simulations quantitatively reproducing production and pressure data, experts have to visually inspect each member of

the ensemble to further discard non-physical simulations. This process is currently performed manually and can be time consuming. Moreover, the viscous fingering process involves a notoriously chaotic and unstable geometry. In particular, two different viscous fingering simulations can both be realistic from an expert's point of view (and yield valid physical properties for reservoir exploitation) even though saturation would admit fingers with a drastically different shape and distribution in space. This high geometrical variability makes it particularly challenging to derive a meaningful distance metric to compare saturation scalar fields between a simulation and a ground truth.

For studying scalar fields, topological data analysis (TDA) has been used in recent years as a robust and reliable setting, allowing to hierarchically define features of interest in the data [EH09]. Its applicability to time-varying data [SB06; Bre+10], ensembles [Fav+19] and comparisons [Sol+18a] makes it a reliable candidate for assessing the likeliness of simulations in an ensemble given a ground truth. Although several approaches have explored the promising potential of TDA for extracting and characterizing the features of interest in viscous fingering simulations [FGT16b; Luk+17a], no approach has been proposed to estimate the similarity between two time-varying viscous fingerings based on topological representations.

In this chapter, we address the aforementioned issues by proposing a novel framework, based on topological data analysis, for quantitatively ranking simulations from an ensemble with respect to a ground truth in a viscous fingering case study. This framework allows experts to easily separate the most realistic simulations from the most unrealistic ones. It is based on a new approach for comparing temporal sequences of *persistence diagrams*, specifically adapted to the problem of viscous fingering. Extensive experiments quantitatively show the superiority of our approach compared to traditional alternatives. The framework also includes an interactive visual system for exploring the output rankings. Finally, we report a complete case study for which the presented approach has been applied *in-situ* (i.e. during the simulation).

6.1.1 Related work

Viscous fingering

Viscous fingering, sometimes known as the Saffman-Taylor instability, is

a well-known instability encountered in soils and porous media [ST58], arising from the unfavorable mobility ratio between an injected fluid and the fluid in place, for instance when injecting water in a highly viscous oil. These phenomena have been studied in the context of petroleum engineering at multiple scales [Ska+14; Gao11]. Other factors than the viscosity ratio are at play, such as properties inherent to the medium in which the fingering takes place [Hom87; TSJ15; Ska+11; TJC16]. In practice, performing waterflood in highly viscous oil can lead to physical instabilities resulting in fingering patterns, with water flowing in preferential paths and bypassing large quantities of oils. To prevent this phenomenon, polymer can be injected in order to increase the water viscosity, therefore making the injection front more stable, and leading to increased macroscopic oil recovery [Lou+18]. There are multiple numerical models that can describe the evolution of fluids in the context of water floods; some have been qualitatively compared to acquisitions [Ria+07; SIK12; Lou+18], but the literature lacks robust ways to quantitatively compute their difference.

Topological data analysis techniques have been used to study the viscous fingering phenomenon, for instance in ensembles of particle simulations, to determine how the resolution affects fingers [FGT16b; Luk+17a] or to provide frameworks for their visual exploration and interpretation [Luc+19]; but never, to our knowledge, for the purpose of comparing simulations, in particular with a reference. TDA techniques have also been applied *in-situ* [Lan+14], which demonstrates their interest and relevance in the context of large-scale simulations. However, to the best of our knowledge, no data analysis method has yet been proposed for the in-situ analysis of viscous fingers.

Feature-oriented distances

For comparing simple discrete scalar fields such as images, intuitive approaches are point-wise geometric distances such as the Euclidean and chord distances, or distances with a statistic awareness such as the Mahalanobis distance or correlation coefficients [CC05] (note that this is distinct from distances assessing a compression loss, as seen in chapter 3). In specific applicative domains, however, the experts' knowledge should be accounted for to gain a more precise insight of what is of actual interest in the data and which patterns or subsets are interesting to compare. Consequently, feature-oriented distance definitions are exposed in the remainder of this section. Associating geometrical loci in scalar data based on a high-level definition of features of interest often relies on com-

puting the overlap of geometrical sub-domains [SW17; Bre+10; Bre+11; SB06; Sil95; SW96; SW97; SW98]. Such methods are used for feature tracking in time-varying data [SW99]. On another note, Transportation theory offers an important continuous formulation of this problematic, with the notion of a Wasserstein and *Earth mover's* distance [LB01], which has gained interest in recent years [Cut13; Sol+16; Sol+15; Lav+18]. In the discrete setup, when applied to topological structures such as persistence diagrams, transport-based matching methods suffer from instabilities in the geometrical domain [CEM06], for which the underlying metric can be specifically corrected [Sol+18a] depending on the context. Though this family of approaches for computing distances between features based on transport seems promising for the problem of comparing viscous fingers, there is, to our knowledge, no work studying such an application.

6.1.2 Contributions

This chapter makes the following new contributions:

1. **Approach:** we present a novel analysis framework allowing to select relevant members in a simulated ensemble given a ground truth. The system yields a ranking that allows to visually explore the most likely simulations and discard the most unrealistic ones.

2. **Metrics:** new topological metrics for comparing time-varying viscous fingers are introduced, based on the Wasserstein matching of persistence diagrams, specifically tuned for the viscous fingering phenomenon and integrated over time.

3. **Case study:** a complete case study of a viscous fingering simulation ensemble is documented, along with a proof-of-concept in-situ implementation of our approach.

4. **Evaluation:** the metrics and ranking framework are qualitatively evaluated with feedback from domain experts. The quantitative performance of our approach is also analyzed and its superiority over traditional alternatives is demonstrated.

6.2 Darcy-type porous media simulation

This section describes the context of reservoir simulation. As highlighted in chapter 2, there are multiple models for simulating flow in porous media. Though our viscous finger analysis framework is not limited to a

specific simulation model, we introduce here Darcy-type simulations, for which the physics is governed by quantities averaged over control volumes. We consider diphasic flow with oil and water.

Eq. 6.1 describes mass conservation, where $i \in \{o, w\}$ denotes the oil and water phase; ϕ is the porosity of the medium; ρ_i is the mass density of phase i; S_i is the *saturation* of phase i (it stands for the volume fraction of phase i); q_i is the well source term (injection/production) of phase i; and $\mathbf{v}_i$ is the velocity of phase i. If V_{tot} denotes the total volume, then the mass of component i is given by $m_i = V_{\text{tot}}\phi\rho_i S_i$.

$$\frac{\partial}{\partial t}(\phi\rho_i S_i) = -\nabla \cdot \rho_i \mathbf{v}_i + q_i \tag{6.1}$$

Darcy's law is an equation that describes fluid flow in porous media, determined experimentally by H. Darcy in 1856 for one phase [Dar56], and which can be derived from the Stokes equations [Whi86]. Its extension to multiphasic flow is given in Eq. 6.2, where $\mathbf{v}_i$ is the velocity of phase i; $\mathbf{K}$ is the absolute permeability tensor of the porous medium; μ_i is the viscosity of i; $\mathbf{g}$ is the acceleration of gravity; P_i is the pressure of phase i; kr_i is the relative permeability of phase i. In our model, kr_i is a function of water saturation.

$$\mathbf{v}_i = -\mathbf{K}\frac{kr_i}{\mu_i}(\overrightarrow{\nabla} P_i - \rho_i \mathbf{g}) \tag{6.2}$$

Furthermore, as shown in Eq. 6.3, oil saturation can be simply expressed in terms of water saturation, and water pressure can be expressed in terms of oil pressure, with P_c being the *capillary pressure*, a function of water saturation.

$$\begin{cases} S_w = 1 - S_o \\ P_w = P_o - P_c \end{cases} \tag{6.3}$$

In this model, the unknowns are the saturations S_i and pressures P_i. The system formed by Eq. 6.1, 6.2, and 6.3 can then be solved numerically to yield the evolution of fluid in porous media under Darcy's approximation. Moreover, models exist [Lev41; Tho60; BC64] for expressing P_c as a function of S_w, which can be obtained experimentally through centrifugal fan experiments. Relative permeabilities kr_o and kr_w, also functions of S_w, are more elusive. Numerous models have been proposed in the literature in various contexts [Cor54; CR56; Chi84; Kil76; Car81; ALE99; FB98], and

there is a number of methods for building them from interpretation of lab experiments [OBT90; Mac+93; DAB00; Ric+52; Hag80]. Their correct definition, however, is key to a realistic description of flow in porous media, and can be quite difficult to obtain depending on the recovery mechanism, especially in processes involving severe viscous fingering patterns (in which case Darcy's law can become approximate) or when dealing with an extra fluid phase, like an injected gas phase [Bak88], notably because of the limited availability of experimental measurements. In the remainder of this work, relative permeabilities are considered as an input parameter of simulations.

Most of reservoir simulators are based on finite volumes discretizations of Eq. 6.1, 6.2, 6.3 on a gridded 2D or 3D model, in which independent variables are constant in each grid block. These quantities must be determined at each time-step by solving the sets of non-linear conservation equations. The results shown in the experiments section were obtained in the 2D case with an in-house research reservoir simulator [Pat+14; JML14] using an IMPES scheme (IMplicit Pressure, Explicit Saturation) [CHL04], which separately computes saturation with an explicit time approximation, and pressure with an implicit one. At every time-step, scalar data defined on control volumes is updated. As there are multiple variables, the simulator outputs multiple fields, like phase pressures and saturations. The pressure field is very diffusive, and in the diphasic case the saturation is constrained by Eq. 6.3. Thus, a good indicator of the simulation state is the scalar field of water saturation S_w, which we will use as input data in the following.

6.3 Analysis framework

This section describes the problem of representing viscous fingers appearing in time-varying saturation fields, comparing them across simulations, and our approach for addressing this problem. In the following, we will note each time step of the reference ground-truth acquisition A_t and each time step of a simulation run S_t. Then, the goal of our framework is to efficiently compute relevant similarity measures, to rank simulation runs in order of increasing distance to the acquisition, so as to present to the experts the most plausible simulations for further inspection (Fig. 6.1). Note that the number of *available* acquired time steps A_t is in practice significantly lower than the number of simulated time steps S_t. The simulator is thus set up to output additional time steps corresponding to a specific

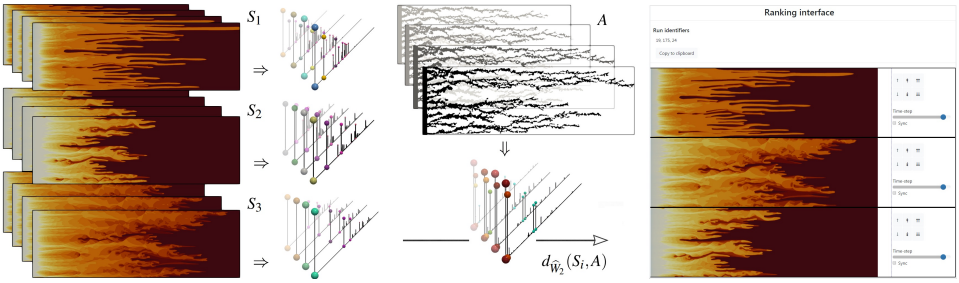

Figure 6.1 – *Overview of the ranking framework. An ensemble of viscous fingering simulation runs S_1, S_2, S_3 is launched, and the persistence diagram of newly available time-steps can be computed* in-situ *(left). Only persistence diagrams for which there is a matching ground truth image (center, top) are computed. Diagrams of every simulation are compared with the diagrams of the ground truth (center, bottom) at matching time-steps. This comparison, based on a metric $\widehat{W}_2$ combining the notions of persistence and geometry, outputs a distance measurement, which can be integrated over time to form the metric $d_{\widehat{W}_2}$. This produces a final ranking (right) which characterizes the quality of simulations, allowing experts to select and explore best performing runs automatically.*

set of volumes of injected water, which were recorded for each time step A_t. This physical criterion allows to reliably match in time acquired and simulated time steps.

6.3.1 Feature representation

As discussed in the introduction, trying to reproduce the viscous fingering phenomenon with Darcy-type simulation software is very challenging because the fingering geometry greatly varies when one modifies input parameters, even slightly. In particular, the input parametersconsidered here are the relative permeabilities kr_i. When comparing a simulation to an acquired ground-truth, this great geometrical variability challenges traditional image based distances, either point-wise based (L_2 norm) or morphing based [Cut13]. Moreover, the raw geometry of the viscous fingers can be insufficient in practice to identify all plausible simulations. Indeed, two geometrically different simulations can be deemed equally plausible by the experts if they share more abstract similarities, involving the number of fingers, their prominence and their progress in the porous medium. Thus, a proper feature representation, capable of abstracting these informations, is required to correctly represent the viscous fingering. Fig. 6.2 illustrates the extent to which the geometry of fingers may vary across simulations and how clearly distinct simulations can be judged as equally plausible by the experts.

The water saturation scalar field allows to visually identify fingers,

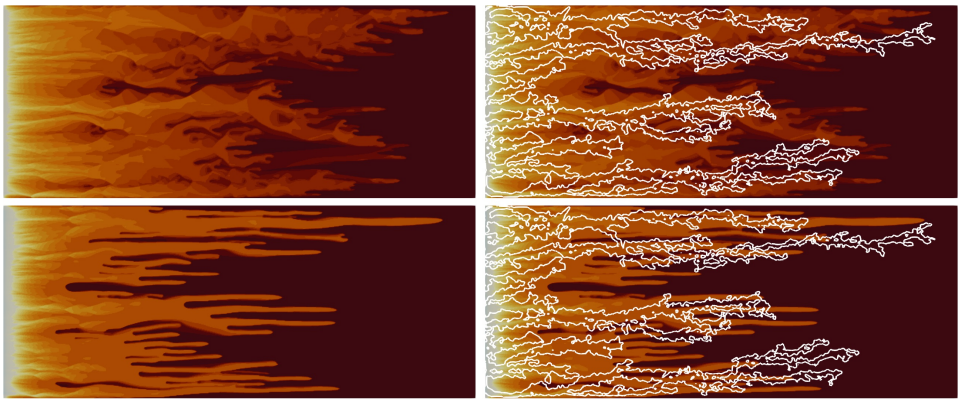

Figure 6.2 – *Late time-steps of two Darcy-type simulation runs launched with different model parameters (left column). The ground truth obtained with X-rays is contoured in white (right column, superimposed). Runs exhibit a very chaotic finger geometry.*

because they form a clear, sharp frontier with the background (as the geometric domain was initially filled with oil). The first step for identifying fingers then consists in extracting a sub-level set $f_{-\infty}^{-1}(w)$ of water saturation, for an isovalue w chosen properly (typically 0.12 in our experiments), to extract the geometric domain $\mathcal{M}$ where fingers are effectively present. Let $\mathcal{F} = f_{-\infty}^{-1}(w)$ be that sub-part of $\mathcal{M}$. The same workflow can be applied on acquired X-ray images.

To compare a simulation to an acquired ground-truth, a naive strategy consists in estimating overlaps between the sub-level sets of saturation of the simulation and the acquisition, for a given time-step, and use the area of such an overlap as a measure of likeliness. However, this purely geometric approach appears to be inadequate in practice due to the important variability in the number and shape of fingers, which then would not be accounted for, as illustrated in Fig. 6.2.

A natural way of characterizing fingers while taking their shape into consideration is to provide $\mathcal{F}$ with a *descriptive* scalar field, for instance a geodesic distance from the injection point. Here, the injection point is the left boundary of the domain, so the scalar field can simply be the x geometrical coordinate. Local maxima of this new scalar field would then correspond to the tips of viscous fingers, and saddles to valleys between fingers. Since they correspond to finger tips, maxima of the x geometrical coordinate provide a useful information to represent the progress of each finger in the porous medium. Moreover, in this setting, the persistence of the pair involving each maximum directly represents the length of the corresponding finger, which can be used as a reliable measure of importance given this application, to distinguish the main fingers from noise.

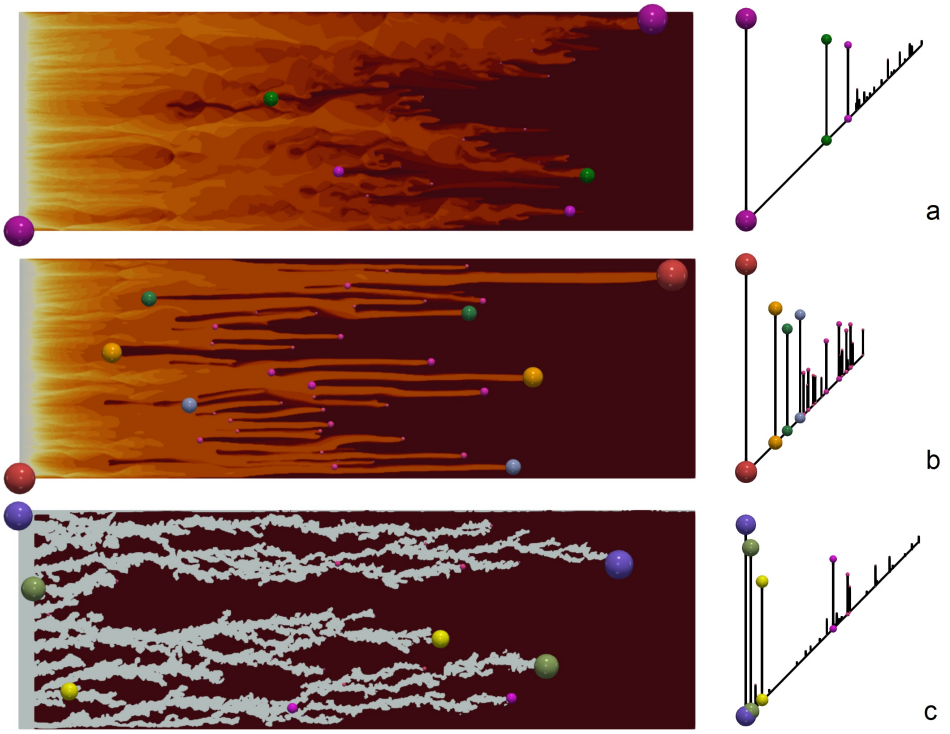

Figure 6.3 – *Simulated time-steps (left column, a and b) and the matching ground truth image (left column, c). Critical points are represented with spheres, and the corresponding persistence diagram is shown on the right. As every critical point belongs to only one persistence pair, the color of spheres encodes their persistence pair; and their diameter encodes its height (the larger, the higher the persistence). The most important fingers can clearly be identified by looking at the most persistent pairs in diagrams (right column). For instance, we can see that the three most important fingers in the acquisition are the purple, green and yellow ones.*

The persistence diagram directly captures this information, in a robust and hierarchical setting. Fig. 6.3 illustrates the correspondence between fingers in the domain and pairs of critical points in persistence diagrams. In this context, persistence diagrams seem to be a promising feature representation for viscous fingers, since they efficiently describe their number, progress through the porous medium as well as their prominence.

6.3.2 Metrics between time-varying persistence diagrams

Considering that viscous fingers are captured by persistence diagrams, computing the similarity of a simulation with respect to a ground truth would require, in a first step, to compute distances between persistence diagrams. As outlined in the introduction, metrics have been introduced for this purpose, notably the (2-)Wasserstein distance in the *birth-death* space, noted W_2 (Eq. 3.3).

A drawback of W_2 is that it does not take into consideration geometrical information, other than coming from the birth-death space. Fig. 6.4 illustrates this limitation. To avoid this problem, a *lifted* adaptation of W_2, noted $\widehat{W}_2$, including geometrical components can be considered, as we introduced in chapter 5 (see Eq. 5.5). It is subject to input parameters indicating the importance given to each geometrical component. Note that, the *Earth mover's distance* [LB01], noted EMD, which is an alternative of interest too for our application, is a special case of $\widehat{W}_2$, for $\alpha_x = \alpha_y = 0$. It is similar to W_2, but it only operates on the geometrical space instead of the birth-death space. Thus, the lifted Wasserstein distance $\widehat{W}_2$ can be interpreted as a compromise between the W_2 distance in the diagram birth-death space and the EMD in the geometrical domain.

In practice, an important characteristic of a viscous fingering simulation run is the moment when the longest finger arrives at the right boundary, called *breakthrough time*. Correctly predicting this event is essential because once it is reached, it means a preferential path has been formed, allowing water to easily flow through, therefore impacting production. Consequently, the position of local maxima (i.e. fingertips) is more important than the position of saddles (i.e. finger branchings).

Then, given a time-step t, to compare the persistence diagrams coming from a simulation S_t and the acquisition A_t, metrics should be more sensitive to the advancement of fingertips, then to the global extent of fingers, and lastly to their y location in the domain. Thus, at this point, we propose to select the following metrics:

- The Earth mover's distance for local maxima: $EMD(S_t, A_t)$
- The 2-Wasserstein distance: $W_2(S_t, A_t)$
- The 2-Wasserstein distance, lifted to include geometrical information (the position of critical points): $\widehat{W}_2(S_t, A_t)$. As in this application, the advancement of fingertips is much more important than their vertical position in the domain, we only consider the x-coordinate of critical points. Thus, lifting coefficients (cf. Eq. 5.5) are $\beta_x = 10/\gamma$ (γ being the extent of the geometrical domain), $\beta_y = 0$, and $\alpha_x = \alpha_y = 1/\rho$ (ρ being the range of the scalar function).

Characterizing the evolution of fingers through time raises the necessity to integrate these metrics, as they are intended to evaluate the proximity between persistence diagrams for a single time-step t. Thus, to measure the distance from a time-varying simulation S to the time-varying acquired

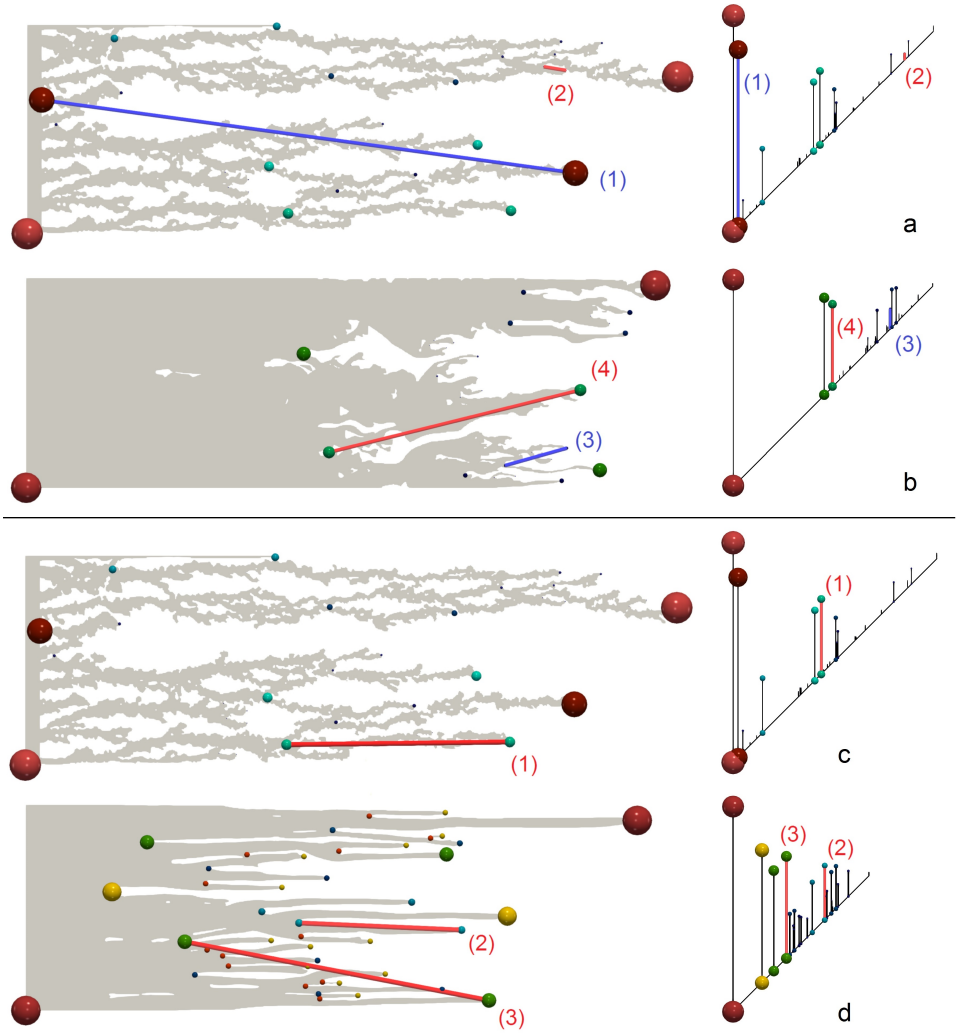

Figure 6.4 – *Limitations of matching methods based on geometry only (a, b) and matching methods based on persistence only (c, d). As the Earth mover's distance (top) only considers the geometrical location of extrema, it can incorrectly associate critical points belonging to unrelated fingers. For instance, the large pair in the acquisition, represented with a blue segment (a, (1)), is matched to a pair with low persistence (b, (3)) because their maxima are geometrically close; and the large red finger in the middle of a simulation (b, (4)) is matched to a small protrusion (a, (2)) attached to the largest finger in the acquired image. On the bottom, the reference metric for matching persistence diagrams, the 2-Wasserstein metric, is shown to associate the bottom finger in the acquisition (c, (1)) to a finger in the middle of a simulation (d, (2)), because their persistence is comparable. Taking both geometry and persistence into account, a* lifted *version of the Wasserstein metric associates (c, (1)) to (d, (3)), which is farther away in terms of persistence, but has the nearest maximum, and is qualitatively the best match.*

ground truth A, we introduce time-integrated versions, based on the L_2 norm, of the metrics mentioned above:

- $d_{EMD}(S,A) = \left(\sum_t \left(EMD(S_t, A_t)\right)^2\right)^{1/2}$
- $d_{W_2}(S,A) = \left(\sum_t \left(W_2(S_t, A_t)\right)^2\right)^{1/2}$
- $d_{\widehat{W}_2}(S,A) = \left(\sum_t \left(\widehat{W}_2(S_t, A_t)\right)^2\right)^{1/2}$

As suggested by the experts, the displacement speed of the saturation front is key to predicting breakthrough time. They suggested to match in priority simulations which display compatible fronts in terms of velocity during the experiment. Given fingers are captured by persistence diagrams, a possibility for appreciating their evolution with respect to that suggestion would be to compute the sequence of distances between diagrams in successive time steps. In other words, for each couple of consecutive time steps t and $t+1$, compute a distance between S_t and S_{t+1} (subsection 3.3.4), and integrate for all time steps. Here the chaotic behavior displayed by fingers when input simulation parameters change need not be taken into account: we are considering a unique simulation run, which has temporal coherence, therefore it is easier to choose a fitting metric.

As highlighted in Fig. 6.5, a working solution is the 2-Wasserstein distance, lifted to give more importance to the y coordinate of maxima: $\widetilde{W}_2(S_t, S_{t+1})$, with lifting coefficients $\beta_x = 0$, $\beta_y = 10/\gamma$, $\alpha_x = \alpha_y = 1/\rho$ (γ is the geometrical extent; ρ is the scalar range). Because of the variability in the number of fingers, however, considering the difference of traveled distances alone could be problematic, for many little fingers going slow could compare close to few fast fingers. We then consider the mean traveled distance per finger. Thus, if n_{A_t} (resp. n_{S_t}) denotes the number of fingers in the acquisition (resp. simulation) at time-step t, we propose to evaluate the velocity-oriented difference given by:

- $d_{\widetilde{W}_2}(S,A) = \left(\sum_t \left(\frac{1}{n_{S_t}}\widetilde{W}_2(S_t, S_{t+1}) - \frac{1}{n_{A_t}}\widetilde{W}_2(A_t, A_{t+1})\right)^2\right)^{1/2}$

Fig. 6.6 summarizes the different metrics discussed in this subsection. Then, given an ensemble of time-varying viscous fingering simulations, each run S can be compared to the reference acquired ground-truth A and runs can be ranked in increasing order of distance to A and presented to the experts for further visual inspection.

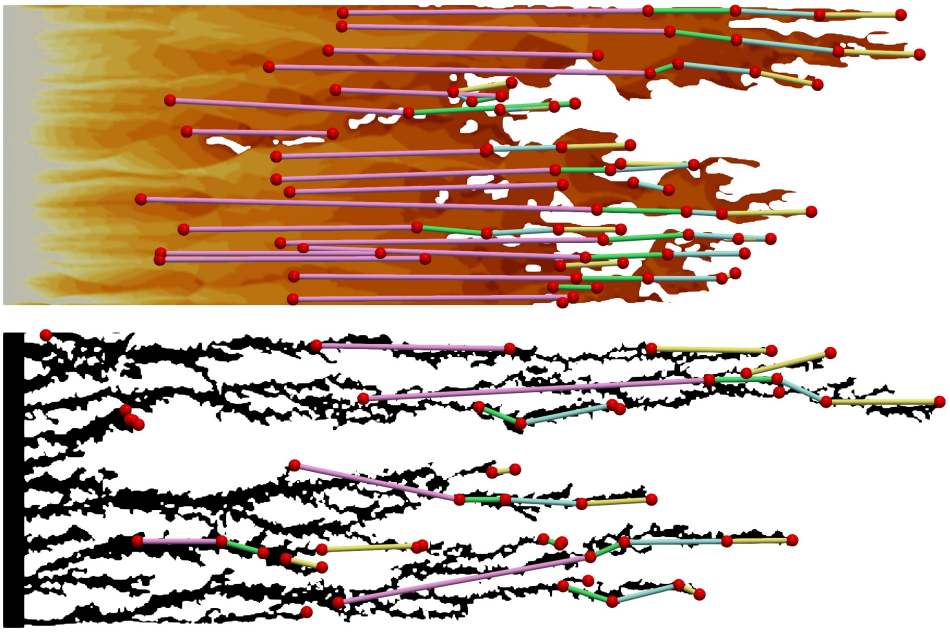

Figure 6.5 – *Critical point trajectories based on optimal matchings. Within a given simulation (or the acquisition, bottom), the geometrical coherence of fingers allows to use a lifted version (that gives importance to the y-coordinate of fingers) of the Wasserstein metric to correctly track the evolution of persistence pairs. Comparing the mean distance traveled by fingers between simulations and the acquisition, for each pair of time-steps, is proposed as a* velocity*-aware metric.*

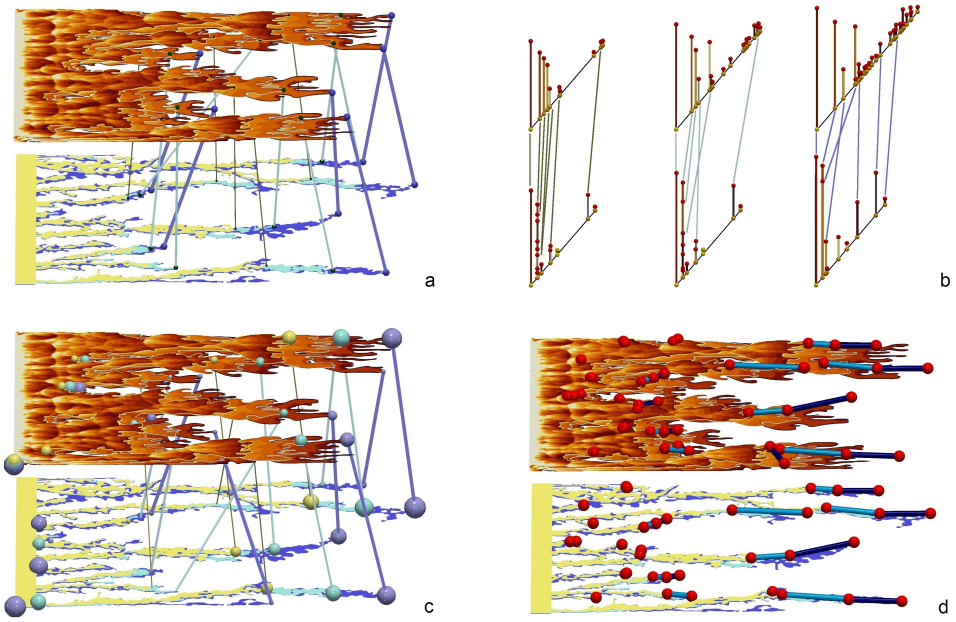

Figure 6.6 – *Proposed metrics. Three consecutive time-steps are represented in yellow, cyan and purple (a, c, d). The Earth mover's distance (a) only considers the geometrical position of critical points for identifying fingers between the simulation and acquisition. The Wasserstein distance (b) only considers persistence pairs in the* birth-death *space. The lifted Wasserstein distance (c) considers both the geometry and persistence during the matchings. The velocity-oriented distance (d) compares the mean transport of persistence pairs between the acquisition and the simulation.*

6.3.3 In-situ deployment

Doing feature extraction and comparisons can be problematic for very large numbers of simulations, in terms of data movement. Fortunately, computing the metrics we just presented does not require to have all time-steps available at once, and hence may be done in a progressive fashion. We propose, within our framework, to implement the computation of metrics comparing the acquired reference to the simulation *in-situ*, that is, without storing time-steps to the disk first. Precomputed persistence diagrams for the acquisition are first loaded in memory. Whenever the simulation attains a time for which there is a corresponding acquisition time step, the saturation scalar field is passed to our analysis pipeline, which applies a threshold, extracts the persistence diagram, and computes the per time step distance to the acquisition diagram (for instance $\widehat{W}_2(S_t, A_t)$). The distance can then be accumulated as the simulation unfolds. The *in-situ* application of our pipeline is optional: time-steps can still be saved to the disk and the pipeline applied *post-mortem* if desired.

6.3.4 Visual interface

Each metric previously mentioned naturally produces a ranking of simulations, from the most to the less plausible ones. We propose a way to visually inspect those rankings with a lightweight HTML+Javascript application, as illustrated in Fig. 6.7. We use the same interface, as we will see in Sec. 6.4.1, to allow experts to rank and label simulations in order to form a *reference ranking*. This visual interface offers linked views of the saturation scalar fields, to visually compare simulations runs, for a given time step t which can be interactively selected. If needed (in particular to generate a reference ranking, cf. Sec. 6.4.1), the experts can interactively modify the suggested ranking by displacing a selected run up or down the ranking, either by unit or long jumps (typically skipping 10 or 50 positions).

6.4 Case study

This section exposes our experimental setting, details a complete viscous fingering use case and summarizes the results of our approach in terms of performance and quality, compared to other classical methods.

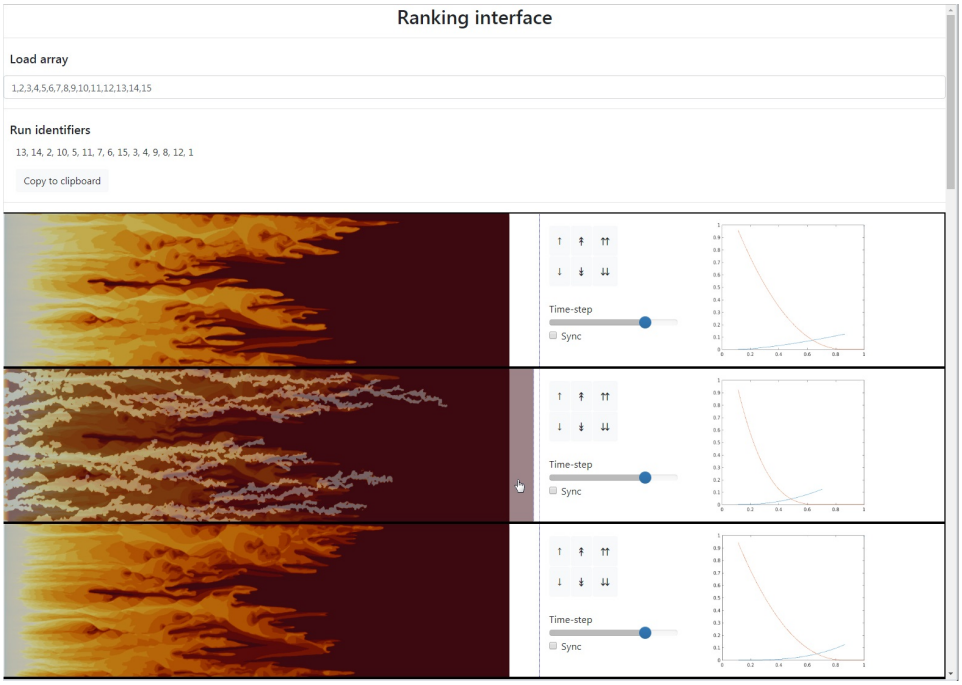

Figure 6.7 – *Lightweight web interface for exploring and modifying simulation rankings. An ordered list of runs can be loaded as an input. Time-steps of runs are then displayed on the left pane; they can be hovered with the mouse to be compared with the matching acquired image. A slider allows to select the time-step to compare. Users can edit the ranking with swapping buttons. For each run, kr curves (input parameters of the simulation model) are displayed on the right.*

6.4.1 Experimental protocol

The behavior of a slab, initially filled with oil and water at connate water saturation, then subject to a water injection in reservoir conditions is captured through X-rays: X-ray images are processed in order to be converted to maps of the fluid saturations within the slab. 2D simulations are then launched with varying input parameters in order to match the simulation results to the experimental measurements and to the fluid saturation maps derived from X-ray images. The resemblance of fingers can be taken into account manually by experts, involving an interpretation of X-rays and an assessment of likeliness according to their expertise. A reference ranking of simulations is then produced by the experts with the help of our visual interface (Sec. 6.3.4), and is compared to the rankings generated by the metrics proposed in our framework (Sec. 6.3.2). The performance and quality of our approach are then evaluated.

Acquisition

The acquisition process is long (several months) and expensive. The de-

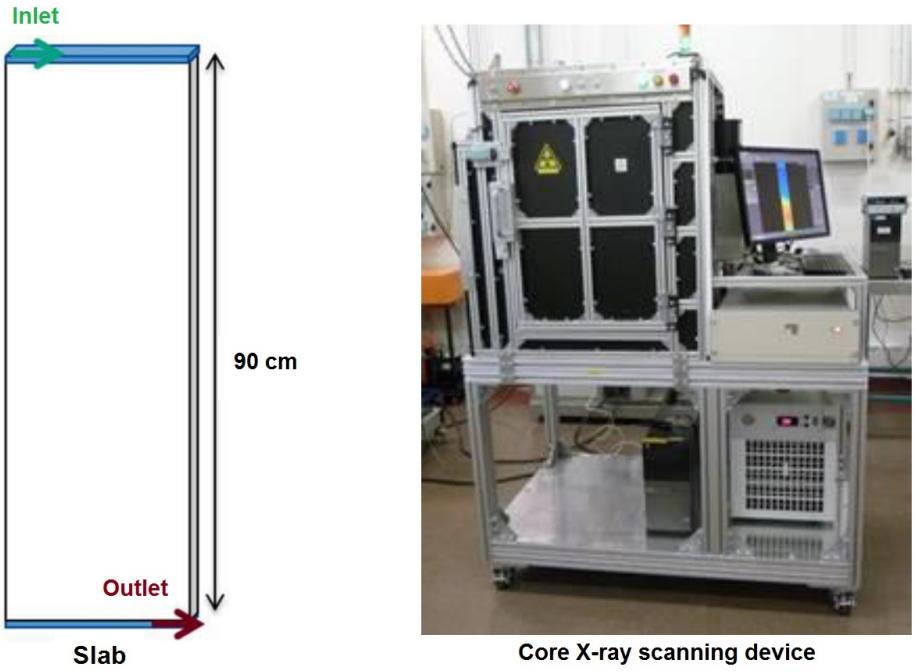

Figure 6.8 – *Schematic view of the slab used for the acquisition (left, experimental protocol described in [Lou+18]). It is disposed vertically during the capture. On the right, a typical X-ray scanning device for imaging flow in porous media is shown. Though it was not the case for the captures shown in this chapter, acquisitions can be operated on cores subject to subsurface reservoir conditions, at high pressures (more than* 600 *bars).*

tailed experimental setup is that of [Lou+18; Fab+15], further described in [Ska+14; Ska+09; Ska+12]. We consider slabs of Bentheimer sandstone, of dimension $30\times90\times2.45$ cm, with porosity of approximately 23% and absolute permeability of 2.5 Darcy (when $S_w = 1$). The slab is coated with two epoxy layers. Three grooves are cut into the first epoxy layer on the extreme faces, and connected to injection and production rails. It is mounted vertically in a 2D X-ray scanning rig (Fig. 6.8).

Slabs first undergo cleaning and calibration processes. A tracer test validates the homogeneous behavior of the slab, then oil is injected to reach initial conditions. The system is then aged at 50°C and ambient pressure for a month, in order to get closer to field conditions.

The experiment takes place at near-atmospheric pressure (2 to 3 bars). The water injection rate is kept constant at 3 cm^3/h, which corresponds to the velocity in fields far from wells. One of the fluids is doped with an X-ray absorbing chemical for increasing the contrast. The X-ray scanner is equipped with an X-ray source (40 to 60 kV at maximum 0.4 mA) and a camera capturing a slice of 0.5×11.5 cm. The camera moves in horizontal rows along the slab. An image scan for a 30×30 cm sample takes

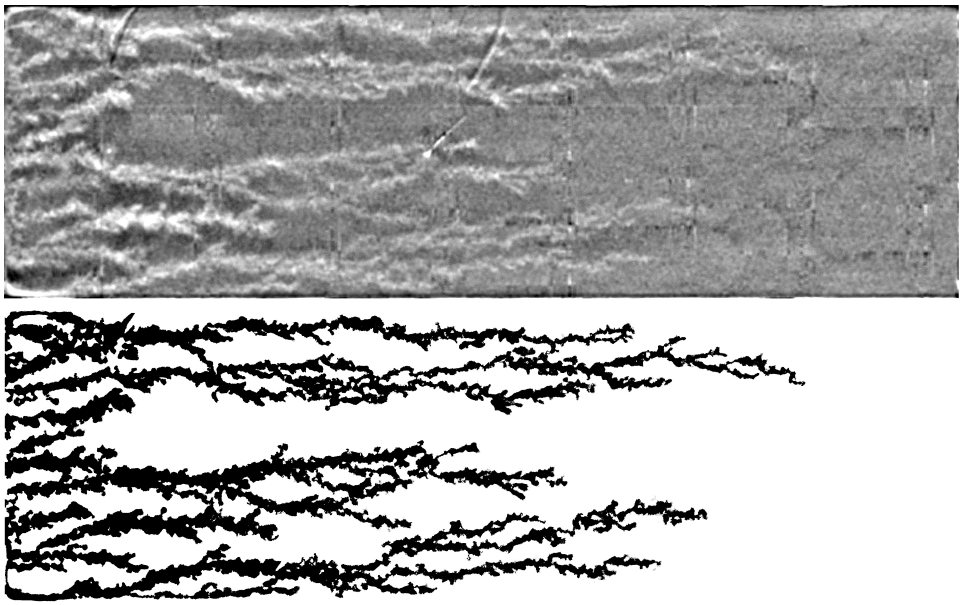

Figure 6.9 – *De-noised X-ray capture (top) and segmented fingers (bottom). Fingers were manually detoured by experts.*

4-5 min, during which the fluid has moved by about 0.1 to 0.2 mm. The captured images, which are noisy and exhibit severe vertical and horizontal artifacts, are filtered [Ska+12], and manually segmented by an expert (Fig. 6.9) to differentiate fingers from the background, hence forming a reference finger geometry $\mathcal{F}_A$.

Simulations

The input parameters of the 2D simulations are relative permeabilities (kr_w and kr_o). In our model, they are a function of water saturation S_w. We consider relative permeabilities in the form of simple Corey curves (Fig. 6.10, see Eq. 6.4, [BC64]), subject to parameters kr_o^0 (oil relative permeability endpoint), S_{or} (residual oil saturation), and power law exponents n_c and n_w. Other quantities like S_{wc} (connate water saturation) and kr_w^0 (water relative permeability endpoint) are determined by measurement.

$$\begin{cases} kr_o(S_w) = kr_o^0 \times \left(\frac{1-S_w-S_{or}}{1-S_{or}-S_{wc}}\right)^{n_c} \\ \\ kr_w(S_w) = kr_w^0 \times \left(\frac{S_w-S_{wc}}{1-S_{or}-S_{wc}}\right)^{n_w} \end{cases} \tag{6.4}$$

The varying parameters of these curves, kr_o^0, S_{or}, n_c and n_w, were randomly sampled and selected using the algorithm by Wootton, Sergent, Phan-Tan-Luu [SCS12] to ensure a good initial covering of the space. The geometrical domain is discretized on a regular grid of 290×890 blocks. 200 runs were launched on 400 simulation nodes (2 MPI ranks per run), then

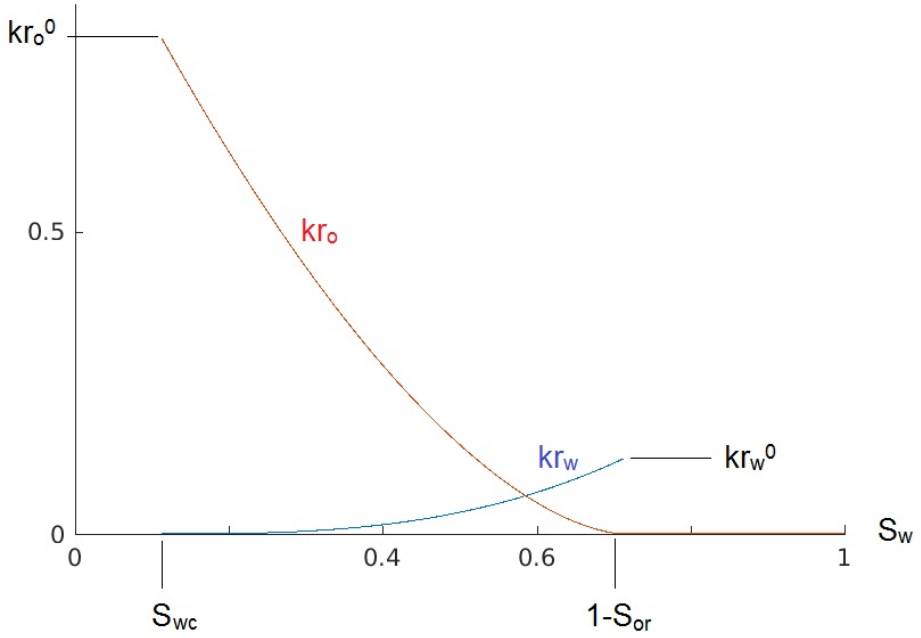

Figure 6.10 – *Typical relative permeability curves: for oil (red) and water (blue). In this graph, the abscissa is water saturation S_w. Intuitively, it represents the extent to which the flow of a phase (say the flow of oil) is inhibited by the presence of another (say the presence of water).*

time-steps for which there was a corresponding X-ray image were saved (8 available segmented images).

Expert ground truth

Images of the water saturation scalar field were captured for each simulation at available X-ray time-steps, for experts to manually form a reference ranking. During this process, experts would quickly discard runs that seem too far from the acquired X-ray image (either because fingers are advancing too slow, too fast, or in a too diffusive fashion). Then, they would closely look at the shape and advancement of fingers when comparing two close runs. We propose to use our lightweight web based visual interface (Sec. 6.3.4), presented in Fig. 6.7, to alleviate this tedious process. Note that images from all simulations were necessary for the experts to form the reference ranking, so the corresponding simulated time-steps were saved to the disk. Once this reference is formed, later analyses can be done *in-situ*.

6.4.2 Framework performance

In this section, we evaluate the quantitative gains of using our approach *in-situ* (Sec. 6.3.3), in terms of time and storage.

Table 6.1 – *Time performance comparison (CPU time in seconds), for a single time-step, between the* in-situ *implementation (everything is computed during the simulation using the local CPU's memory) and the* post-mortem *implementation (the scalar fields are analyzed and compared in a post-processing stage).*

Step	CPU time (s)		Detailed CPU time (s)	
	in-situ	*post-mortem*		
Simulation iteration	3.096	3.096		
Time step storage		0.076		
Catalyst analysis	1.111		0.063	Persistence diagram
			0.002	Distance
			0.001-	Distance storage
			1.046	Catalyst overhead
Data transfer		0.021		Lustre to workstation
Data conversion		0.246		.unrst to .vtk
Paraview analysis		2.189	0.084	Persistence diagram
			0.002	Distance
			2.085	Paraview overhead
Analysis time	**1.111**	**2.532**		
Total processing	4.207	5.628		

Tab. 6.1 provides a CPU time comparison of our analysis pipeline based on the lifted 2-Wasserstein metric (Sec. 6.3.2), for the two different strategies: *(i) in-situ*, where the analysis is run on the fly during the simulation and without data storage and *(ii) post-mortem*, where selected time steps are stored to disk to be analyzed after the simulation has finished. Persistence diagrams are computed using the algorithm by Gueunet et al. [Gue+17], and Wasserstein distances are computed using the exact approach by Soler et al. [Sol+18a], both available in the Topology ToolKit (TTK, [Tie+17]). The *in-situ* implementation is based on Paraview Catalyst [Aya+15], which is called by the simulation code at selected time steps to run a python script instantiating our analysis pipeline.

The numbers are given for a single simulation time-step, therefore at the finest possible time resolution (about ten thousand time-steps are required to complete a run). Figures are averages on the time-steps of a typical run. *In-situ* computations are done on a supercomputer (among the 51^{st} of TOP500 Nov. 2018) with Xeon(R) E5-2680v3 processors; the post-process is done on a local workstation with a Xeon(R) E5-2640v3 processor, so the performance is not the same for computing persistence diagrams.

Ideally, overheads due to different data layouts and conversions (lines *"Catalyst overhead"* and *"Paraview overhead"*, Tab. 6.1) in the simulator and VTK/ParaView would not enter into account (if the simulator were to directly output a VTK data array). We are left with two unneeded stages in the *post-mortem* approach: time step writes and data transfer. Selecting 8 time-steps from 200 simulations, this amounts to 155.2 s. of IO time versus approximately 0.6 ms. necessary to write the $1,600$ doubles in the *in-situ* case (the $1,600$ distance estimations), which is $260,000$ times faster.

In terms of data storage, the *post-mortem* strategy requires to store and potentially transfer 3.28 GiB (2.1 MiB per time-step) of data, versus 12.5 KiB for $1,600$ doubles (representing the $1,600$ distance estimations), which is $275,000$ times lighter.

Thus, overall, the in-situ instantiation of our framework reduces data movement by 5 orders of magnitude, while dividing by 2.3 the time required to analyze a time-step (line *"Analysis time"*, Tab. 6.1).

6.4.3 Ranking quality

In this section, we evaluate quantitatively the relevance of the rankings obtained with each of the metrics discussed in Sec. 6.3.2, and compare them to rankings obtained with *overlap methods*, traditionally used for associating geometrical sub-domains [SW17; Bre+10; Bre+11; SB06; Sil95].

Let $\mathcal{F}_{A_t}$ be the acquired finger geometry (Sec. 6.4.1) and $\mathcal{F}_{S_t}$ be the sub-level set of the simulated water saturation at time step t. The overlap $O(A_t, S_t)$ between A_t and S_t is the volume of $\mathcal{F}_{A_t} \cap \mathcal{F}_{S_t}$ divided by the volume of $\mathcal{F}_{A_t} \cup \mathcal{F}_{S_t}$. From this we can define a distance:

- $d_O(A, S) = \left(\sum_t \left(1 - O(A_t, S_t)\right)^2 \right)^{1/2}$

Integrating the overlap $\breve{O}_t(S) = 1 - O(S_t, S_{t+1})$ between S_t and S_{t+1} for a single simulation and comparing it to the integrated overlap for the acquisition yields a *velocity*-oriented version:

- $d_{\breve{O}}(A, S) = \left(\sum_t \left(\breve{O}_t(S) - \breve{O}_t(A)\right)^2 \right)^{1/2}$

At this point, we need to compare different rankings to the reference ranking constituted by experts. Let R_1 and R_2 be two rankings of n simulations. One of the most commonly used methods for computing a degree of similarity between R_1 and R_2 is **Kendall's** τ [CD10; GI11]: for all couples $(r_i, r_j) \in R_1^2$ and $(s_i, s_j) \in R_2^2$:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \text{sign}(r_i - r_j)\text{sign}(s_i - s_j) \tag{6.5}$$

Table 6.2 – *Quality of rankings. Kendall coefficients between each ranking and the reference ground truth formed by experts are computed (closest to* 1 *is best). Since the order in which the poorest runs are ordered in the expert's ranking is arbitrary, coefficients are also computed for the (50 and 25) best simulations according to each method. The best coefficient for each case is shown in bold.*

Method	O	W_2	$\widehat{W}_2$	EMD	$\breve{O}$	$\breve{W}_2$
All	0.37	0.25	0.26	0.15	0.12	**0.41**
Top 50	0.22	0.46	**0.66**	0.47	-0.29	0.46
Top 25	0.13	0.29	**0.84**	0.70	-0.13	0.42

It corresponds to the number of pairs (i, j) for which r_i and r_j in R_1 have the same ordering as s_i and s_j in R_2 minus the number of pairs for which the orderings in R_1 and R_2 are different. In other words, it is the difference between the number of concordant pairs and the number of discordant pairs. The closer this number is to 1 in absolute value, the more compatible the rankings, τ being close to -1 indicates that the two rankings are in reverse order.

Over the 200 examined runs, many were quickly discarded by experts during the manual ranking, because they were too far from the acquisition. Thus, as the order of the poorest runs is not important to the experts, we also compute the similarity with the reference ranking for the best (top 50 and top 25) identified runs according to each method. Resulting Kendall coefficients are exposed in Tab. 6.2.

Observing lines 2 and 3 in Tab. 6.2, we can first note that the overlap method (column "O") does not perform well. This behavior was expected because of the very chaotic geometry of fingers. The Wasserstein method, which is the traditional reference metric for comparing persistence diagrams, is shown in column "W_2". The Earth mover's distance method (column "EMD") only takes the geometrical information of extrema into account, regardless of their persistence. It seems to perform better than W_2, which is unexpected, because EMD can wrongly associate small-scale details to large-scale ones. The lifted Wasserstein method, which includes persistence information and favors a geometrical direction, is shown in column "$\widehat{W}_2$". As it achieves the best overall Kendall coefficients, it seems that $\widehat{W}_2$ manages to combine the advantages of both EMD and W_2, not just being a simple compromise between the two. Lastly, metrics based on the difference of distances traveled by fingers (columns $\breve{O}$ and $\breve{W}_2$) do not appear to be able to produce relevant rankings.

Table 6.3 – *Appreciation of rankings for the 25 best runs (the 25 first of each method's ranking). Diffuse runs, slow runs and runs in common with the expert's ranking are counted for each method. Each ranking is shown to an expert using our web interface and their appreciation is noted.*

Method	O	W_2	$\widehat{W}_2$	EMD	$\breve{O}$	$\widetilde{W}_2$
too diffuse	0	4	0	5	0	7
too slow	0	0	0	0	17	0
common	0	10	21	18	0	11
appreciation	poor	good	best	poor	wrong	wrong

6.4.4 Expert feedback

In this section we expose a qualitative appreciation, collected from experts, and a discussion of ranking results.

We show in Tab. 6.3 the qualitative appreciation of rankings. The poor performance of velocity-based metrics ($\breve{O}$ and $\widetilde{W}_2$) was unexpected. Looking at the rankings, we see that aberrant runs are considered close to the ground truth by these two metrics. There are three types of aberrant runs: too slow, too fast, and too diffusive (i.e. whose finger tips grow large and do not form a very sharp frontier with the background, as illustrated in Fig. 6.11). $\breve{O}$ gives a good score to runs that are too slow, and $\widetilde{W}_2$, on the contrary, scores highly runs that are too diffusive.

The number of slow runs is counted in Tab. 6.3. The $\breve{O}$ approach, which computes overlaps in consecutive time-steps, does not discard them. The reason for this is that in simulations, the water saturation front is very smooth though in the acquisition fingers display a quite dendritic structure. Therefore, the overlap between successive time-steps of smooth fingers going slow compares close to the overlap between successive time-steps of thin fingers going fast.

The number of runs which exhibit a very diffusive behavior is also counted. These diffuse fingers seem to give trouble to the $\widetilde{W}_2$ metric (and also to EMD and W_2). This is because in the set of available simulations, among all which are diffuse some inevitably end up at the exact same advancement as the acquisition when the threshold stage (introduced in Sec. 6.3.1) is applied. Taking into account the number of fingers ($\widetilde{W}_2$) or considering their branching events (W_2) is apparently insufficient to discard them. Note that the $\widehat{W}_2$ metric (and even W_2) were well appreciated by the experts because the top simulations in their rankings display

Figure 6.11 – *Diffuse run example. The tips of fingers grow wider than their base, forming a sort of inverted funnels. The saturation field does not exhibit a very sharp frontier with the background.*

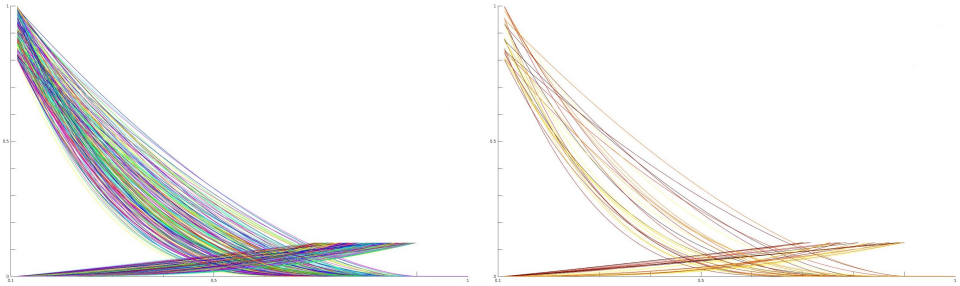

Figure 6.12 – *Input relative permeability curves of all 200 simulations (left, with random colors) and for the 25 best selected runs (right, darker is closest to ground truth). No clear pattern was seen that could discriminate relative permeability curves yielding the best fingers.*

fingers whose tips are quite close to the acquisition near breakthrough time, though for W_2, there seems to be a higher distance variability. As for the basic overlap method O, it fails to identify the real best simulations, though it does not incorrectly bring out aberrant runs either (be it too diffuse or too slow). Its ranking, though, feels random to the experts. Overall, the best performing metric seems to be $\widehat{W}_2$, as confirmed quantitatively and qualitatively.

Taking a step back, the approach we proposed is appealing to experts because it allows them to include geometrical information into their parametric studies, in an autonomous and systematic way (instead of manually inspecting and checking runs). Using the $\widehat{W}_2$-based ranking results, we present in Fig. 6.12 all permeability curves and those yielding the best simulation runs. We see no clear pattern arising, either visually or numerically with respect to L_2 or Haussdorff [Mun14] distances between curves. This confirms the well-known difficulty of calibrating *kr* curves.

6.5 Summary

In this chapter, we presented a framework for enabling the automatic comparison and ranking of simulation runs to an acquired ground truth. We presented a set of metrics specifically adapted to this task in the case of viscous fingering in porous media. After evaluation, we identified the best fitting approach (the $\widehat{W}_2$ metric, which computes a geometrically tuned Wasserstein distance between simulation and acquisition persistence diagrams, on a per-time-step basis). This quantitative measurement method supplements the expert's, and allows them to automatically form a subjective ranking close to one they would have manually produced. We demonstrated the possibility and showed the advantage of implementing the computation of this metric in-situ, speeding up the analysis pipeline by a factor of 2.3 and reducing data movement by 5 orders of magnitude. We proposed a lightweight web interface to explore automatically generated rankings and manually edit them. As with the best metric $\widehat{W}_2$, there are still some diffuse runs in the ranked best fifty, we believe it could be further enhanced, for instance by considering the sharpness of the water saturation front, or by augmented the persistence diagram with the individual volume of fingers. Future experiments should assess how sensitive the ranking is to such a modification of the lifting, as well as modifications of the lifting coefficients. Besides, though in our experimental setting, simulations took place in a 2D domain, nothing in our approach is restrictive to this case: in future work, this methodology can be applied to 3D cases.

On another note, on the set of 200 simulations, we were not able to identify a regime of best-matching input parameters. The combination of our metric with production and pressure data (at injectors/producers) in a follow-up study would be interesting in this regard. Trying to understand the influence of the space of input parameters, here *kr* curves, proves quite challenging. In our study, only four parameters (power law exponents and endpoints) were sampled, yielding a four-dimensional space, but the number of sampled parameter may be significantly higher. In particular, this study may be extended to permeability curves other than based on simple Corey power laws.

We think it would be insightful to develop a visual interface for exploring such spaces of model parameters. We hope to see, in future studies, how accounting for the geometrical and topological quality of a modeled phenomenon can be used to infer or restrict model parameters.

Conclusion

7

In this thesis, we proposed different methods, based on topological data analysis, and in particular *persistence*-oriented constructs, in order to address modern problematics concerning the increasing difficulty in the analysis of scientific data.

We first introduced in chapter 2 the scientific domains in which this work is inscribed: reservoir simulation for oil and gas exploration, scientific visualization and topological data analysis. We highlighted some modern problematics inherent (but not exclusive) to these fields, that are the focus of this thesis. In particular, we emphasized on the difficulty to analyze modern data-sets, on the one hand due to the increase in data volumes, and on the other hand due to their growing complexity, from static scalar data, to time-varying data, to multi-parameter studies.

Next, in chapter 3, we exposed the theoretical foundations and background of topology and topological data analysis. We presented the central concept of topological *persistence*, which offers a hierarchical characterization of structures of interest in the data; we then reviewed state-of-the-art methods allowing to compute and compare persistence-based characterizations from scalar data. Specifically, we discussed the *bottleneck* metric (yielding "coarse" topological comparisons), and the *Wasserstein* metric (yielding "finer" topological comparisons).

From then on, we presented the following new contributions.

7.1 Summary of contributions

With the later purpose of applying our methods to industrial data (*e.g.* linked to reservoir simulation), under a unified framework, we devised the approach of this thesis with the idea of defining structures of interest in scalar data as topological constructs based on the notion of persistence. More accurately, we expressed them as *topological features*, consisting of persistence pairs in persistence diagrams. Henceforth, we focused on the

problematics raised in the context of data reduction and analysis as follows:

Lossy compression with topological guarantees (chapter 4)

In a first effort to address the issue of data volume growth, we proposed a new lossy compression scheme. We designed our method with strong topological guarantees concerning the bottleneck metric, so that the topological features we defined could be preserved through the compression process. We empirically showed that the approach could compress the data with favorable topological accuracy with respect to the Wasserstein metric as well, as compared to other state-of-the-art compression algorithms. We showcased the method on concrete examples, yielding in practice high compression factors (56 and 360, for CDF and medical scan data, respectively). We proposed two extensions to our approach, one offering additional control over geometrical error (*e.g.* the point-wise error), another allowing to use our method conjointly with existing lossy compressors in order to enforce topological control.

These contributions were integrated within the Topology Tool-Kit (TTK [Tie+17]) and released open-source on github.

Feature tracking in time-varying data (chapter 5)

Drawing on the advantages of topological persistence, we then proposed to address the problem of analyzing more involved scientific data, in the present case time-varying data. For that we designed a new method for following, or *tracking*, topological features over time. This raised robustness and performance challenges. To overcome the limitations of the standard Wasserstein metric in this regard, we proposed an extension of that metric, as well as a new efficient way to compute it, gaining orders of magnitude speedups over the state-of-the-art exact approach and proving faster in practice than approximate methods for small diagrams. We demonstrated the applicability and robustness of our tracking solution on 2D and 3D time-varying data-sets.

Notice that using a unified framework, based on persistence diagrams, is advantageous for it allows to use conjointly the tracking method we proposed with our compression approach, as the latter offers guarantees on the preservation of topological features. This novel feature tracking approach, as well as the enhanced method for computing Wasserstein distances and extensions, were also integrated within TTK and released open-source on github.

Ranking simulation runs to a ground truth in parametric studies (chapter 6)

Finally, we targeted even more challenging scientific data, with a parametric study related to reservoir simulation: the modeling of the *viscous fingering* phenomenon in porous media. We showed on this practical case how to capture precise structures of interest in scalar data (sets of viscous fingers), with the help of persistence-based topological abstractions. We then further adapted the topological metrics that we defined in the context of generic time-varying data, to the case of viscous fingers, for the purpose of capturing discrepancies between simulation runs and X-ray images acquired in lab experiments.

We evaluated the proposed metrics with feedback from experts. Finally, based on the best evaluated topological metric, we constituted a ranking framework for rating the fidelity of simulation runs with respect to the ground truth. The framework is implemented in an *in-situ* environment, with the purpose of addressing the data movement problematic that arises in such parametric studies due to infrastructure limitations.

7.2 Perspectives

The technical contributions brought in this thesis are integrated in TTK, which is an open-source BSD-licensed framework (published on github), under active development, and used by both industrial and academics. This is a sound basis for many evolution perspectives thanks to an active open-source community.

This thesis was also partially inscribed in the AVIDO project, aimed at exploring "in situ analysis and visualization for large scale numerical simulations", involving industrial (EDF SA, Kitware, Total SA) and academic (CNRS, INRIA) partners. Its success lays the ground for more future collaborations between these partners, and indicates a growing interest of the industry in topological data analysis (in particular persistence-based methods) for scientific data analysis and visualization.

We suggest hereafter some open problems and directions that we identify as a pertinent continuation of the work presented in this thesis.

Industrial in-situ integration and deployment

As the persistence-driven feature definition, and consequently the ranking and comparison methods seem adapted to the study of viscous

fingers, we believe it would benefit from a tighter integration between the reservoir simulation software and TTK.

For example, a follow-up viscous fingering study combining the new proposed topological metrics with pre-existing metrics (which consider one-dimensional history data but not scalar fields) would be interesting.

Another interesting example for domain experts would be the coupling of our compression algorithm with the simulator in an *in-situ* environment, allowing to conduct persistence-driven analysis on reduced data, *post-mortem*. This raises multiple challenges as, first, the topological simplification step is sequential, and, second, our compression algorithm should be adapted to handle multi-block datasets, which is mandatory in the case of large field simulations launched on many MPI ranks.

A further possible development of topology-aware compression would be its direct application to time-varying data, as does ISABELA [Lak+11], benefiting from the topological proximity between successive time-steps, and its adaptation to the case of unstructured meshes.

Generalization to other topological abstractions

Throughout this thesis, we showed how metrics relating to persistence diagrams could be adapted for the needs of *(i)* topological loss evaluation, *(ii)* time-varying feature tracking, *(iii)* comparing simulation runs.

As a proper definition of these metrics is done by fine-tuning lifting coefficients, which should be done depending on the geometry of the studied phenomenon, the precisely quantified effects of changing these coefficients should be further studied in a first step, especially regarding our ranking framework.

Then, we believe that these essential metrics could be adapted and applied to other, richer TDA constructs, such as the Reeb graph [Ree46], which may capture richer discrepancies in the topological changes of contours (with applications, for instance, in molecular dynamics); or the 1-skeleton of the Morse-Smale complex [Gyu08], which operates based on the scalar field gradient (with applications, for instance, in the extraction of the topological skeleton of porous networks).

These structures may require more involved comparison metrics, such as Levenshtein distances [Lev66] or graph-edit distances [SF83; Sri+18]. The computational aspect is important in this regard, and we believe that these metrics could benefit greatly from the persistence formalism, which allows to consider topological features at coarse (hence computationally reachable) levels.

Figure 7.1 – *3D CT-scan of a rock containing fossils (left); segmentation induced by* 20% *of the most persistent features (center); segmentation induced by the four most persistent features (right). Image from [PT16].*

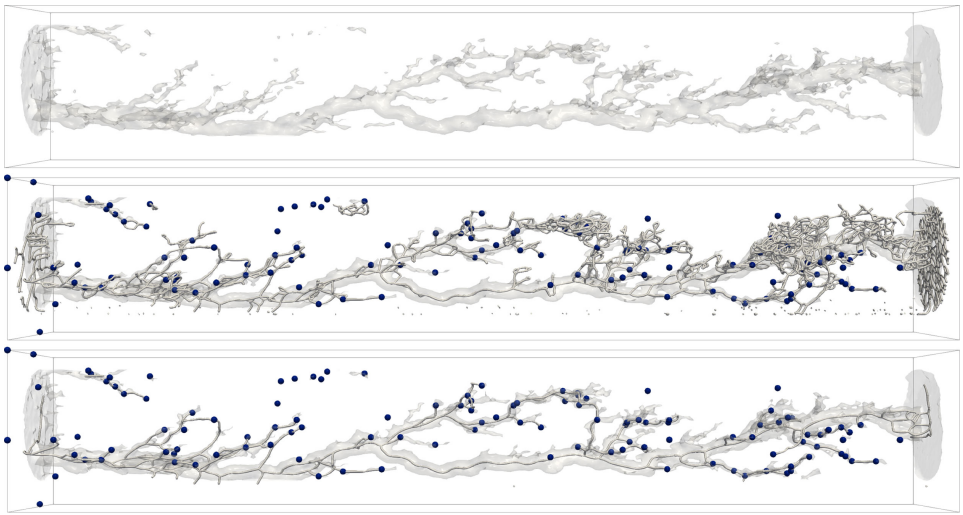

Figure 7.2 – *Wormhole structure in porous media (top), whose topological skeleton was extracted by computing the 1-separatrices of the Morse-Smale complex (middle); the most* persistent *separatrices capture the wormhole structure (bottom). From [PT16].*

Industrial perspectives

Considering the issues raised by data growth in size and intricacy, we believe that efforts should be made to reach towards both data reduction and an *in-situ* applicability of new analysis methods. In the longer run, we would be interested to see the methods presented here and their possible extension to other topological structures applied to different types of scientific studies.

In the context of the academic-industrial partnership of this thesis, TDA techniques were presented to experts of other scientific domains linked to oil and gas applications. For example, in the field of Digital Rock Physics, specialists were able to perform segmentations of microfossils from 3D CT-scans (Fig. 7.1), and to characterize wormhole structures in porous media (Fig. 7.2).

This demonstrates the growing interest of the industry for topology-based solutions for the analysis and reduction of large data, which seems particularly promising for future potential applications.

Bibliography

[Aki+19] Kazunori Akiyama, Antxon Alberdi, Walter Alef, Keiichi Asada, Rebecca Azulay, Anne-Kathrin Baczko, David Ball, Mislav Baloković, John Barrett, Dan Bintley, et al. "First M87 Event Horizon Telescope Results." In: *The Astrophysical Journal Letters* 875.1 (2019) (cit. on p. 1).

[ALE99] Faruk O Alpak, Larry W Lake, and Sonia M Embid. "Validation of a modified Carman-Kozeny equation to model two-phase relative permeabilities". In: *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers. 1999 (cit. on p. 115).

[Att+09] D. Attali, M. Glisse, S. Hornus, F. Lazarus, and D. Morozov. "Persistence-sensitive simplification of functions on surfaces in linear time". In: *TopoInVis Workshop*. 2009 (cit. on p. 49).

[Att+13] D Attali, U Bauer, O Devillers, M Glisse, and A Lieutier. "Homological reconstruction and simplification in R3". In: *Proc. of ACM Symposium on Computational Geometry*. 2013 (cit. on pp. 46, 78).

[Aya+15] Utkarsh Ayachit, Andrew Bauer, Berk Geveci, Patrick O'Leary, Kenneth Moreland, Nathan Fabian, and Jeffrey Mauldin. "Paraview catalyst: Enabling in situ data analysis and visualization". In: *Proc. of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Vis.* ACM. 2015 (cit. on pp. 19, 129).

[Aya+16] Utkarsh Ayachit, Andrew Bauer, Earl PN Duque, Greg Eisenhauer, Nicola Ferrier, Junmin Gu, Kenneth E Jansen, Burlen Loring, Zarija Lukić, and Suresh Menon. "Performance analysis, design considerations, and applications of extreme-scale in situ infrastructures". In: *SuperComputing*. 2016 (cit. on p. 18).

[Bak88] LE Baker. "Three-phase relative permeability correlations". In: *SPE Enhanced Oil Recovery Symposium*. Society of Petroleum Engineers. 1988 (cit. on p. 116).

[Ban70] T. F. Banchoff. "Critical Points and Curvature for Embedded Polyhedral Surfaces". In: *The American Mathematical Monthly* (1970) (cit. on p. 38).

[Bau+16] Andrew C Bauer, Hasan Abbasi, James Ahrens, Hank Childs, Berk Geveci, Scott Klasky, Kenneth Moreland, Patrick O'Leary, Venkatram Vishwanath, and Brad Whitlock. "In situ methods, infrastructures, and applications on high performance computing platforms". In: *Computer Graphics Forum*. Vol. 35. 3. Wiley Online Library. 2016 (cit. on p. 19).

[BB67] Cx K Batchelor and GK Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 1967 (cit. on p. 11).

[BC64] Royal Harvard Brooks and Arthur Thomas Corey. "Hydraulic properties of porous media and their relation to drainage design". In: *Transactions of the ASAE* (1964) (cit. on pp. 115, 127).

[BC89] D. P. Bertsekas and D. A. Castanon. "The Auction Algorithm for the Transportation Problem". In: *Ann. Oper. Res.* 20.1-4 (1989) (cit. on pp. 49, 85, 105).

[BC99] Rainer E Burkard and Eranda Cela. "Linear assignment problems and extensions". In: *Handbook of combinatorial optimization*. Springer, 1999 (cit. on p. 47).

[BDM09] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment Problems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009 (cit. on p. 85).

[Bek+14] Kenes Beketayev, Damir Yeliussizov, Dmitriy Morozov, Gunther H Weber, and Bernd Hamann. "Measuring the distance between merge trees". In: *Topological Methods in Data Analysis and Visualization III*. Springer, 2014 (cit. on p. 107).

[Ben+12] J. C. Bennett, H. Abbasi, P. T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pebay, D. Thompson, H. Yu, F. Zhang, and J. Chen. "Combining in-situ and in-transit processing to enable extreme-scale scientific analysis". In: *High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for*. Nov. 2012 (cit. on p. 18).

[Ber98] Dimitri P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. 1998 (cit. on p. 85).

[Bha+18] Harsh Bhatia, Attila G. Gyulassy, Vincenzo Lordi, John E. Pask, Valerio Pascucci, and Peer-Timo Bremer. "TopoMS: Comprehensive Topological Exploration for Molecular and Condensed-Matter Systems". In: *Journal of Comp. Chemistry* (2018) (cit. on p. 16).

[BHB84] John Baillieul, John Hollerbach, and Roger Brockett. "Programming and control of kinematically redundant manipulators". In: *The 23rd IEEE Conference on Decision and Control*. IEEE. 1984 (cit. on p. 25).

[Bia+08] S. Biasotti, D. Giorgio, M. Spagnuolo, and B. Falcidieno. "Reeb graphs for shape analysis and applications". In: *TCS* (2008) (cit. on p. 17).

[BL71] François Bourgeois and Jean-Claude Lassalle. "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices". In: *Commun. ACM* 14.12 (Dec. 1971) (cit. on p. 88).

[Bli78] James F Blinn. "Simulation of wrinkled surfaces". In: *ACM SIGGRAPH*. Vol. 12. 3. ACM. 1978 (cit. on p. 15).

[BLW12] U Bauer, C Lange, and M Wardetzky. "Optimal topological simplification of discrete functions on surfaces". In: *Discrete and Computational Geometry* (2012) (cit. on pp. 49, 77).

[Boc+18] A. Bock, H. Doraiswamy, A. Summers, and C. Silva. "TopoAngler: Interactive Topology-Based Extraction of Fishes". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018) (cit. on pp. 16, 34).

[Bot88] Raoul Bott. "Morse theory indomitable". In: *Publications Mathématiques de l'IHÉS* 68 (1988) (cit. on p. 24).

[Bou06] Nicolas Bourbaki. *Théorie des ensembles*. Springer, 2006 (cit. on p. 26).

[Bou07] Nicolas Bourbaki. *Topologie générale: Chapitres 1 à 4*. Springer Science & Business Media, 2007 (cit. on p. 26).

[BPZ99] Chandrajit L. Bajaj, Valerio Pascucci, and Guozhong Zhuang. "Single Resolution Compression of Arbitrary Triangular Meshes with Properties". In: *IEEE DC*. 1999 (cit. on p. 59).

[BR07] Martin Burtscher and Paruj Ratanaworabhan. "High Throughput Compression of Double-Precision Floating-Point Data". In: *2007 Data Compression Conference (DCC'07)*. 2007 (cit. on p. 57).

[Bre+10] P. T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. "Analyzing and Tracking Burning Structures in Lean Premixed Hydrogen Flames". In: *IEEE Transactions on Visualization and Computer Graphics* 16.2 (Mar. 2010) (cit. on pp. 83, 84, 100, 112, 114, 130).

[Bre+11] P.T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell. "Interactive Exploration and Analysis of Large Scale Simulations Using Topology-based Data Segmentation". In: *IEEE Transactions on Visualization and Computer Graphics* (2011) (cit. on pp. 16, 34, 74, 83–85, 100, 103, 104, 114, 130).

[BS98] Chandrajit Bajaj and Daniel Schikore. "Topology preserving data simplification with error bounds". In: *Computers and Graphics* 22.1 (1998) (cit. on p. 59).

[BW94] Michael Burrows and David Wheeler. *A block sorting lossless data compression algorithm*. Tech. rep. Digital Equipment Corporation, 1994 (cit. on p. 57).

[Car81] Francis M Carlson. "Simulation of relative permeability hysteresis to the nonwetting phase". In: *SPE annual technical conference and exhibition*. Society of Petroleum Engineers. 1981 (cit. on p. 115).

[Cas19] Davide Castelvecchi. "Black hole pictured for first time — in spectacular detail". In: *Nature* (Apr. 2019) (cit. on p. 1).

[Cat74] Edwin Catmull. *A subdivision algorithm for computer display of curved surfaces*. Tech. rep. Utah Univ. Salt Lake City School of Computing, 1974 (cit. on p. 15).

[CC05] Chaur-Chin Chen and Hsueh-Ting Chu. "Similarity measurement between images". In: *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*. Vol. 2. IEEE. 2005 (cit. on p. 113).

[CCO17] Mathieu Carrière, Marco Cuturi, and Steve Oudot. "Sliced Wasserstein Kernel for Persistence Diagrams". In: *ICML*. 2017 (cit. on p. 86).

[CD10] Christophe Croux and Catherine Dehon. "Influence functions of the Spearman and Kendall correlation measures". In: *Statistical methods & applications* 19.4 (2010) (cit. on p. 130).

[CEH05] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. "Stability of Persistence Diagrams". In: *Proc. of ACM Symposium on Computational Geometry*. 2005 (cit. on pp. 16, 25, 42, 45, 46, 56, 61, 65, 67, 71, 86).

[CEM06] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. "Vines and Vineyards by Updating Persistence in Linear Time". In: *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*. SCG '06. Sedona, Arizona, USA: ACM, 2006 (cit. on pp. 86, 114).

[Cha+09] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Oudot. "Proximity of Persistence Modules and their Diagrams". In: *SoCG*. 2009 (cit. on pp. 45, 86).

[Chi84] Gian Luigi Chierici. "Novel relations for drainage and imbibition relative permeabilities". In: *Society of Petroleum Engineers Journal* 24.03 (1984) (cit. on p. 115).

[CHL04] Zhangxin Chen, Guanren Huan, and Baoyan Li. "An improved IMPES method for two-phase flow in porous media". In: *Transport in porous media* 54.3 (2004) (cit. on p. 116).

[Com13] DoE Advanced Scientific Computing Advisory Committee. *Synergistic Challenges in Data-Intensive Science and Exascale Computing*. Tech. rep. DoE Advanced Scientific Computing Advisory Committee, Data Sub-committee, 2013 (cit. on p. 17).

[Cor54] Arthur T Corey. "The interrelation between gas and oil relative permeabilities". In: *Producers monthly* (1954) (cit. on p. 115).

[CR56] Arthur Thomas Corey and CH Rathjens. "Effect of stratification on relative permeability". In: *Journal of Petroleum Technology* 8.12 (1956) (cit. on p. 115).

[CSA03] Hamish Carr, Jack Snoeyink, and Ulrike Axen. "Computing contour trees in all dimensions". In: *Computational Geometry* 24.2 (2003) (cit. on pp. 17, 47, 56, 74).

[CSP04] H. Carr, J. Snoeyink, and M. van de Panne. "Simplifying Flexible Isosurfaces Using Local Geometric Measures". In: *IEEE VIS*. 2004 (cit. on pp. 16, 34, 44, 74, 77).

[Cui+16] Hong Cui, Jingjing Zhang, Chunfeng Cui, and Qinyu Chen. "Solving large-scale assignment problems by Kuhn-Munkres algorithm". In: *International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII)*. Jan. 2016 (cit. on p. 93).

[Cut13] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013 (cit. on pp. 85, 114, 117).

[CW84] John Cleary and Ian Witten. "Data Compression Using Adaptive Coding and Partial String Matching". In: *IEEE Trans. Comm.* 32.4 (1984) (cit. on p. 57).

[DAB00] David A DiCarlo, Sahni Akshay, and MJ Blunt. "Three-phase relative permeability of water-wet, oil-wet, and mixed-wet sandpacks". In: *SPE Journal* 5.01 (2000) (cit. on p. 116).

[Dar05] Toby Darling. *Well logging and formation evaluation*. Elsevier, 2005 (cit. on p. 10).

[Dar56] H. Darcy. *Les fontaines publiques de la ville de Dijon*. Dalmont, 1856. URL: https://books.google.co.uk/books?id=42EUAAAAQAAJ (cit. on p. 115).

[DC16] Sheng Di and Franck Cappello. "Fast Error-Bounded Lossy HPC Data Compression with SZ". In: *IEEE Symp. on PDP*. 2016 (cit. on pp. 55, 58).

[De +15] Leila De Floriani, Ulderico Fugacci, Federico Iuricich, and Paola Magillo. "Morse complexes for shape segmentation and homological analysis: discrete models and algorithms". In: *Computer Graphics Forum* (2015) (cit. on p. 16).

[DK91] Akio Doi and Akio Koide. "An efficient method of triangulating equi-valued surfaces by using tetrahedral cells". In: *IEICE Transactions on Information and Systems* 74.1 (1991) (cit. on p. 34).

[DL14] Guillaume Damiand and Pascal Lienhardt. *Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing*. A K Peters/CRC Press, Sept. 2014 (cit. on p. 25).

[Ede+04] Herbert Edelsbrunner, John Harer, Ajith Mascarenhas, and Valerio Pascucci. "Time-varying Reeb Graphs for Continuous Space-time Data". In: *Proceedings of the Twentieth Annual Symposium on Computational Geometry*. SCG '04. ACM, 2004 (cit. on p. 85).

[EH04] H. Edelsbrunner and J. Harer. "Jacobi Sets of Multiple Morse Functions". In: *Foundations of Computational Mathematics*. 2004 (cit. on p. 84).

[EH08] H. Edelsbrunner and J. Harer. "Persistent Homology – a survey". In: *American Mathematical Society* (2008) (cit. on p. 17).

[EH09] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2009 (cit. on pp. 1, 16, 21, 24, 25, 32, 37, 83, 112).

[ELZ02] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. "Topological Persistence and Simplification". In: *Disc. Compu. Geom.* 28.4 (2002) (cit. on pp. 16, 17, 25, 42, 47, 56).

[EM90] Herbert Edelsbrunner and Ernst P Mucke. "Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms". In: *ToG* 9.1 (1990) (cit. on pp. 33, 37).

[EMP06] H. Edelsbrunner, D. Morozov, and V. Pascucci. "Persistence-sensitive simplification of functions on 2-manifolds". In: *Proc. of ACM Symposium on Computational Geometry*. 2006 (cit. on pp. 46, 49).

[Fab+15] C Fabbri, R De Loubens, A Skauge, PA Ormehaug, B Vik, M Bourgeois, D Morel, and G Hamon. "Comparison of History-Matched Water Flood, Tertiary Polymer Flood Relative Permeabilities and Evidence of Hysteresis During Tertiary Polymer Flood in Very Viscous Oils". In: *SPE Asia Pacific Enhanced Oil Recovery Conference*. Society of Petroleum Engineers. 2015 (cit. on p. 126).

[Far08] Michael Farber. *Invitation to topological robotics*. Vol. 8. European Mathematical Society, 2008 (cit. on p. 24).

[Fav+19] Guillaume Favelier, Noura Faraj, Brian Summa, and Julien Tierny. "Persistence atlas for critical point variability in ensembles". In: *IEEE TVCG* 25.1 (2019) (cit. on pp. 17, 112).

[FB98] Darryl H Fenwick and Martin J Blunt. "Network modeling of three-phase flow in porous media". In: *SPE Journal* 3.01 (1998) (cit. on p. 115).

[FGT16a] G. Favelier, C. Gueunet, and J. Tierny. "Visualizing ensembles of viscous fingers". In: *IEEE SciVis Contest*. 2016 (cit. on p. 76).

[FGT16b] Guillaume Favelier, Charles Gueunet, and Julien Tierny. "Visualizing ensembles of viscous fingers". In: *IEEE Scientific Visualization Contest*. 2016 (cit. on pp. 16, 112, 113).

[Fou15] Blender Foundation. *Blender Reference Manual*. `https://docs.blender.org/manual/en/latest/render/cycles/introduction.html`. 2015 (cit. on p. 16).

[Gao+10] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. "A survey of graph edit distance". In: *Pattern Analysis and applications* 13.1 (2010) (cit. on p. 107).

[Gao11] Chang Hong Gao. "Advances of polymer flood in heavy oil recovery". In: *SPE heavy oil conference and exhibition*. Society of Petroleum Engineers. 2011 (cit. on p. 113).

[GI11] Atila Göktas and Öznur Isçi. "A comparison of the most commonly used measures of association for doubly ordered square contingency tables via simulation". In: *Metodoloski zvezki* 8.1 (2011) (cit. on p. 130).

[Gib73] Josiah Willard Gibbs. "A Method of Geometrical Representation of the Thermodynamic Properties by Means of Surfaces". In: *Transactions of Connecticut Academy of Arts and Sciences* (1873) (cit. on p. 15).

[GK12] Satoru Goto and Kazushi Komatsu. "The configuration space of a model for ringed hydrocarbon molecules". In: *Hiroshima Mathematical Journal* 42.1 (2012) (cit. on p. 25).

[Gol66] Solomon W. Golomb. "Run-length encodings". In: *IEEE Trans. on IT* 12.3 (1966) (cit. on p. 57).

[Got88] Daniel H Gottlieb. "Topology and the robot arm". In: *Acta Applicandae Mathematica* 11.2 (1988) (cit. on p. 24).

[Gou71] Henri Gouraud. "Continuous shading of curved surfaces". In: *IEEE transactions on computers* 100.6 (1971) (cit. on p. 15).

[Gri81] Hubert Brian Griffiths. *Surfaces*. CUP Archive, 1981 (cit. on p. 30).

[Gro+07] S. Grottel, G. Reina, J. Vrabec, and T. Ertl. "Visual Verification and Analysis of Cluster Detection for Molecular Dynamics". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007) (cit. on p. 84).

[GT88] Harold N Gabow and Robert E Tarjan. "Algorithms for two bottleneck optimization problems". In: *Journal of Algorithms* 9.3 (1988) (cit. on p. 48).

[Gue+14] D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. "Characterizing Molecular Interactions in Chemical Systems". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014) (cit. on p. 16).

[Gue+16] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. "Contour Forests: Fast Multi-threaded Augmented Contour Trees". In: *IEEE LDAV*. 2016 (cit. on p. 44).

[Gue+17] Charles Gueunet, Pierre Fortin, Julien Jomier, and Julien Tierny. "Task-based Augmented Merge Trees with Fibonacci Heaps". In: *IEEE LDAV*. 2017 (cit. on pp. 47, 68, 97, 99, 129).

[GW11] Y. Gu and C. Wang. "TransGraph: Hierarchical Exploration of Transition Relationships in Time-Varying Volumetric Data". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011) (cit. on p. 84).

[Gyu+08] A. Gyulassy, P. T. Bremer, B. Hamann, and V. Pascucci. "A practical approach to Morse-Smale complex computation: Scalability and Generality". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008) (cit. on pp. 17, 44).

[Gyu+14] A. Gyulassy, D. Guenther, J. A. Levine, J. Tierny, and V. Pascucci. "Conforming Morse-Smale Complexes". In: *IEEE Transactions on Visualization and Computer Graphics* (2014) (cit. on p. 44).

[Gyu+15] A. Gyulassy, A. Knoll, K.C. Lau, B. Wang, P.T. Bremer, M.E. Papka, L. A. Curtiss, and V. Pascucci. "Interstitial and Interlayer Ion Diffusion Geometry Extraction in Graphitic Nanosphere Battery Materials". In: *IEEE Transactions on Visualization and Computer Graphics* (2015) (cit. on p. 16).

[Gyu08] A. Gyulassy. "Combinatorial Construction of Morse-Smale Complexes for Data Analysis and Visualization". PhD thesis. University of California at Davis, 2008 (cit. on p. 138).

[Hag80] Jo Hagoort. "Oil recovery by gravity drainage". In: *Society of Petroleum Engineers Journal* 20.03 (1980) (cit. on p. 116).

[Har98] John C Hart. "Morse theory for implicit surface modeling". In: *Mathematical Visualization*. Springer, 1998 (cit. on p. 25).

[Hat02] A Hatcher. *Algebraic Topology*. Cambridge University Press, 2002 (cit. on p. 30).

[Hau91] Jean-Claude Hausmann. "Sur la topologie des bras articulés". In: *Algebraic topology Poznań 1989*. Springer, 1991 (cit. on p. 24).

[Hei+16] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth. "A Survey of Topology-based Methods in Visualization". In: *Computer Graphics Forum* 35.3 (2016) (cit. on p. 16).

[His18] Hiroshima Prefectural Museum of History. *Nihon fuso koku no zu*. `https://mainichi.jp/english/articles/20180616/p2a/00m/0na/021000c`. 2018 (cit. on p. 14).

[HK73] John E Hopcroft and Richard M Karp. "An n^5/2 algorithm for maximum matchings in bipartite graphs". In: *SIAM Journal on computing* 2.4 (1973) (cit. on p. 47).

[Hom87] George M Homsy. "Viscous fingering in porous media". In: *Annual review of fluid mechanics* 19.1 (1987) (cit. on p. 113).

[HT94] Stephen M. Hannon and J. Alex Thomson. "Aircraft Wake Vortex Detection and Measurement with Pulsed Solid-state Coherent Laser Radar". In: *Journal of Modern Optics* 41.11 (1994) (cit. on p. 83).

[Huf52] David Huffman. "A Method for the Construction of Minimum-Redundancy Codes". In: *Proceedings of the IRE* 40.9 (1952) (cit. on pp. 55, 57).

[HV91] Paul G. Howard and Jeffrey Scott Vitter. "Analysis of Arithmetic Coding for Data Compression". In: *IEEE DCC*. 1991 (cit. on p. 57).

[IKK12] Jeremy Iverson, Chandrika Kamath, and George Karypis. "Fast and Effective Lossy Compression Algorithms for Scientific Datasets". In: *Euro-Par*. 2012 (cit. on pp. 55, 58, 70, 72).

[ILS05] Martin Isenburg, Peter Lindstrom, and Jack Snoeyink. "Lossless Compression of Predicted Floating-Point Geometry". In: *Computer-Aided Design* 37.8 (2005) (cit. on p. 57).

[JML14] S Jaure, A Moncorge, and R de Loubens. "Reservoir simulation prototyping platform for high performance computing". In: *SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference and Exhibition*. Society of Petroleum Engineers. 2014 (cit. on p. 116).

[JS04] Guangfeng Ji and Han-Wei Shen. "Efficient Isosurface Tracking Using Precomputed Correspondence Table". In: *Eurographics / IEEE VGTC Symposium on Visualization*. Ed. by Oliver Deussen, Charles Hansen, Daniel Keim, and Dietmar Saupe. The Eurographics Association, 2004 (cit. on p. 84).

[JS06] Guangfeng Ji and Han-Wei Shen. "Feature Tracking Using Earth Mover's Distance and Global Optimization". In: *Proc. of Pacific Graphics*. 2006 (cit. on p. 85).

[JSW03] Guangfeng Ji, Han-Wei Shen, and Rephael Wenger. "Volume Tracking Using Higher Dimensional Isosurfacing". In: *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. VIS '03. 2003 (cit. on p. 84).

[K S+06] K. Shi, H. Theisel, T. Weinkauf, H. Hauser, H.-C. Hege, and H.-P. Seidel. "Path Line Oriented Topology for Periodic 2D Time-Dependent Vector Fields". In: *Proc. Eurographics / IEEE VGTC Symposium on Visualization (EuroVis '06)*. Lisbon, Portugal, May 2006 (cit. on p. 84).

[Kaj86] James T Kajiya. "The rendering equation". In: *ACM SIGGRAPH*. Vol. 20. 4. ACM. 1986 (cit. on p. 15).

[Kan42] L Kantorovich. "On the translocation of masses". In: *AS USSR* (1942) (cit. on pp. 45, 85).

[Kas+11] J. Kasten, J. Reininghaus, I. Hotz, and H.C. Hege. "Two-dimensional time-dependent vortex regions based on the acceleration magnitude". In: *IEEE Transactions on Visualization and Computer Graphics* (2011) (cit. on p. 16).

[KE07] Thomas Klein and Thomas Ertl. "Scale-Space Tracking of Critical Points in 3D Vector Fields". In: *Topology-based Methods in Visualization*. Ed. by Helwig Hauser, Hans Hagen, and Holger Theisel. 2007 (cit. on p. 84).

[Kil76] JE Killough. "Reservoir simulation with history-dependent saturation functions". In: *Society of Petroleum Engineers Journal* 16.01 (1976) (cit. on p. 115).

[KMN17] Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. "Geometry Helps to Compare Persistence Diagrams". In: *J. Exp. Algorithmics* 22 (Sept. 2017) (cit. on pp. 49, 85, 86, 105, 106).

[Kum16] Kalyan Kumaran. "Introduction to Mira". In: *Code for Q Workshop*. 2016 (cit. on p. 19).

[KY55] H. W. Kuhn and Bryn Yaw. "The Hungarian method for the assignment problem". In: *Naval Res. Logist. Quart* (1955) (cit. on pp. 83, 87).

[Lak+11] Sriram Lakshminarasimhan, Neil Shah, Stéphane Ethier, Scott Klasky, Robert Latham, Robert B. Ross, and Nagiza F. Samatova. "Compressing the Incompressible with ISABELA: In-situ Reduction of Spatio-temporal Data". In: *Euro-Par*. 2011 (cit. on pp. 55, 57, 138).

[Lan+06] D. E. Laney, P.T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. "Understanding the Structure of the Turbulent Mixing Layer in Hydrodynamic Instabilities". In: *IEEE Transactions on Visualization and Computer Graphics* (2006) (cit. on p. 85).

[Lan+14] A. Landge, V. Pascucci, A. Gyulassy, J. Bennett, H. Kolla, J. Chen, and T. Bremer. "In-Situ Feature Extraction of Large Scale Combustion Simulations Using Segmented Merge Trees". In: *SuperComputing*. 2014 (cit. on pp. 83, 113).

[Lav+18] Hugo Lavenant, Sebastian Claici, Edward Chien, and Justin Solomon. "Dynamical optimal transport on discrete surfaces". In: *SIGGRAPH Asia 2018 Technical Papers*. ACM. 2018 (cit. on p. 114).

[LB01] Elizaveta Levina and Peter Bickel. "The EarthMover's distance is the Mallows distance: some insights from statistics". In: *IEEE ICCV*. Vol. 2. 2001 (cit. on pp. 85, 114, 120).

[LCL16] Peter Lindstrom, Po Chen, and En-Jui Lee. "Reducing Disk Storage of Full-3D Seismic Waveform Tomography (F3DT) through Lossy Online Compression". In: *Computers & Geosciences* 93 (2016) (cit. on p. 58).

[Lev41] MoC Leverett. "Capillary behavior in porous solids". In: *Transactions of the AIME* 142.01 (1941) (cit. on p. 115).

[Lev66] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady* 10 (8 1966) (cit. on p. 138).

[LFR13] Benjamin Lorendeau, Yvan Fournier, and Alejandro Ribes. "In-situ visualization in fluid mechanics using catalyst: A case study for code saturne". In: *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*. IEEE. 2013 (cit. on p. 19).

[LI06] Peter Lindstrom and Martin Isenburg. "Fast and Efficient Compression of Floating-Point Data". In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006) (cit. on pp. 55, 57).

[Lin14] Peter Lindstrom. "Fixed-Rate Compressed Floating-Point Arrays". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014) (cit. on pp. 55, 57, 61, 62, 65, 67–69, 72, 73, 78).

[LLT04] Thomas Lewiner, Helio Lopes, and Geovan Tavares. "Applications of Forman's discrete Morse theory to topology visualization and mesh compression". In: *IEEE Transactions on Visualization and Computer Graphics* 10.5 (2004) (cit. on p. 25).

[LM97] Neil K. Laurance and Donald M. Monro. "Embedded DCT Coding with Significance Masking". In: *IEEE ICASPP*. 1997 (cit. on p. 57).

[Lou+18] Romain de Loubens, Guillaume Vaillant, Mohamed Regaieg, Jianhui Yang, Arthur Moncorgé, Clement Fabbri, and Gilles Darche. "Numerical Modeling of Unstable Waterfloods and Tertiary Polymer Floods Into Highly Viscous Oils". In: *SPE Journal* (2018) (cit. on pp. 11, 113, 126).

[Luc+19] Timothy Luciani, Andrew Burks, Cassiano Sugiyama, Jonathan Komperda, and G Elisabeta Marai. "Details-First, Show Context, Overview Last: Supporting Exploration of Viscous Fingers in Large-Scale Ensemble Simulations". In: *IEEE TVCG* 25.1 (2019) (cit. on p. 113).

[Luk+17a] Jonas Lukasczyk, Garrett Aldrich, Michael Steptoe, Guillaume Favelier, Charles Gueunet, Julien Tierny, Ross Maciejewski, Bernd Hamann, and Heike Leitte. "Viscous fingering: A topological visual analytic approach". In: *Applied Mechanics and Materials*. Vol. 869. 2017 (cit. on pp. 112, 113).

[Luk+17b] Jonas Lukasczyk, Gunther Weber, Ross Maciejewski, Christoph Garth, and Heike Leitte. "Nested Tracking Graphs". In: *Comp. Graph. For.* 36.3 (2017) (cit. on pp. 85, 107).

[Mac+93] Donald J MacAllister, Kenneth C Miller, Stephan K Graham, and Chung-Tien Yang. "Application of X-ray CT scanning to determine gas/water relative permeabilities". In: *SPE formation evaluation* 8.03 (1993) (cit. on p. 116).

[Max90] James Clerk Maxwell. *The Scientific Letters and Papers of James Clerk Maxwell: 1846-1862*. Vol. 1. CUP Archive, 1990 (cit. on p. 15).

[Mil63] John Milnor. *Morse Theory*. Princeton U. Press, 1963 (cit. on pp. 24, 35, 37).

[Mon81] Gaspard Monge. "Mémoire sur la théorie des déblais et des remblais". In: *Académie Royale des Sciences de Paris* (1781) (cit. on pp. 45, 85).

[Mor+11] Kenneth Moreland, Ron Oldfield, Pat Marion, Sebastien Jourdain, Norbert Podhorszki, Venkatram Vishwanath, Nathan Fabian, Ciprian Docan, Manish Parashar, Mark Hereld, Michael E. Papka, and Scott Klasky. "Examples of in Transit Visualization". In: *Proceedings of the 2Nd International Workshop on Petascal Data Analytics: Challenges and Opportunities*. PDAC '11. Seattle, Washington, USA: ACM, 2011 (cit. on p. 18).

[Mor10] Dmitriy Morozov. *Dionysus*. `http://www.mrzv.org/software/dionysus/`. 2010 (cit. on pp. 48, 86, 89, 91).

[Mun14] James Munkres. *Topology*. Pearson Education, 2014 (cit. on p. 133).

[Mun57] James Munkres. *Algorithms for the assignment and transportation problems*. 1957 (cit. on pp. 48, 83, 85, 87, 88).

[Nok07] Scott B Nokleby. "Singularity analysis of the Canadarm2". In: *Mechanism and Machine Theory* 42.4 (2007) (cit. on p. 25).

[Nuc+17] Girijanandan Nucha, Georges-Pierre Bonneau, Stefanie Hahmann, and Vijay Natarajan. "Computing contour trees for 2d piecewise polynomial functions". In: *Computer Graphics Forum*. Vol. 36. 3. Wiley Online Library. 2017, pp. 23–33 (cit. on p. 47).

[OBT90] MJ Oak, LE Baker, and DC Thomas. "Three-phase relative permeability of Berea sandstone". In: *Journal of Petroleum Technology* 42.08 (1990) (cit. on p. 116).

[OLe+16] Patrick O'Leary, James Ahrens, Sébastien Jourdain, Scott Wittenburg, David H Rogers, and Mark Petersen. "Cinema image-based in situ analysis and visualization of MPAS-ocean simulations". In: *Parallel Computing* 55 (2016) (cit. on p. 18).

[Ols10] Eric T. Olson. *Personal identity*. 2010 (cit. on p. iii).

[Oze+14] S. Ozer, D. Silver, K. Bemis, and P. Martin. "Activity Detection in Scientific Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (Mar. 2014) (cit. on p. 84).

[Pas+07] V Pascucci, G Scorzelli, P T Bremer, and A Mascarenhas. "Robust on-line computation of Reeb graphs: simplicity and speed". In: *ToG* 26.3 (2007) (cit. on pp. 17, 44).

[Pas+10] V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny. *Topological Data Analysis and Visualization: Theory, Algorithms and Applications*. Springer, 2010 (cit. on p. 16).

[Pat+14] Leonardo Patacchini, Romain De Loubens, Arthur Moncorge, and Adrien Trouillaud. "Four-fluid-phase, fully implicit simulation of surfactant flooding". In: *SPE Reservoir Evaluation & Engineering* 17.02 (2014) (cit. on p. 116).

[Pet+10] Lies Peters, Rob Arts, Geert Brouwer, Cees Geel, Stan Cullick, Rolf J Lorentzen, Yan Chen, Neil Dunlop, Femke C Vossepoel, and Rong Xu. "Results of the Brugge benchmark study for

flooding optimization and history matching". In: *SPE Reservoir Evaluation & Engineering* 13.03 (2010) (cit. on p. 12).

[Pho75] Bui Tuong Phong. "Illumination for computer generated pictures". In: *Communications of the ACM* 18.6 (1975) (cit. on p. 15).

[Pin84] Steven Pinker. "Visual cognition: An introduction". In: *Cognition* 18.1 (1984) (cit. on p. 14).

[Pop03] Stéphane Popinet. "Gerris: A Tree-based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries". In: *J. Comput. Phys.* 190.2 (2003) (cit. on p. 99).

[Pos+03] Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. "The State of the Art in Flow Visualisation: Feature Extraction and Tracking". In: *Computer Graphics Forum* 22.4 (2003) (cit. on p. 84).

[PT16] M Plainchault and J Tierny. *Topological Data Analysis for Scientific Visualization - From Theory to Applications*. Mathias 2016, TOTAL S.A. 2016 (cit. on p. 139).

[Ras+11] Michel Rasquin, Patrick Marion, Venkatram Vishwanath, Benjamin Matthews, Mark Hereld, Kenneth Jansen, Raymond Loy, Andrew Bauer, Min Zhou, Onkar Sahni, Jing Fu, Ning Liu, Christopher Carothers, Mark Shephard, Michael Papka, Kalyan Kumaran, and Berk Geveci. "Electronic Poster: Co-visualization of Full Data and in Situ Data Extracts from Unstructured Grid Cfd at 160K Cores". In: *Proceedings of the 2011 Companion on High Performance Computing Networking, Storage and Analysis Companion*. SC '11 Companion. Seattle, Washington, USA: ACM, 2011 (cit. on p. 18).

[Ras+14] Michel Rasquin, Cameron Smith, Kedar Chitale, E Seegyoung Seol, Benjamin A Matthews, Jeffrey L Martin, Onkar Sahni, Raymond M Loy, Mark S Shephard, and Kenneth E Jansen. "Scalable implicit flow solver for realistic wing simulations with flow control". In: *Computing in Science & Engineering* 16.6 (2014) (cit. on p. 19).

[Ree46] Georges Reeb. "Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique". In: *Académie des Sciences* (1946) (cit. on pp. 44, 138).

[Rei+12] Jan Reininghaus, Jens Kasten, Tino Weinkauf, and Ingrid Hotz. "Efficient computation of combinatorial feature flow fields". In: *IEEE Transactions on Visualization and Computer Graphics* 18.9 (2012) (cit. on p. 84).

[Rei+15] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. "A stable multi-scale kernel for topological machine learning". In: *IEEE CVPR*. 2015 (cit. on p. 86).

[Ria+07] Amir Riaz, Guo-Qing Tang, Hamdi A Tchelepi, and Anthony R Kovscek. "Forced imbibition in natural porous media: Comparison between experiments and continuum models". In: *Physical Review E* 75.3 (2007) (cit. on p. 113).

[Ric+52] JG Richardson, JK Kerver, JA Hafford, and JS Osoba. "Laboratory determination of relative permeability". In: *Journal of Petroleum Technology* 4.08 (1952) (cit. on p. 116).

[Rin+13] Todd Ringler, Mark Petersen, Robert L. Higdon, Doug Jacobsen, Philip W. Jones, and Mathew Maltrud. "A multi-resolution approach to global ocean modeling". In: *Ocean Modelling* 69 (2013) (cit. on p. 83).

[Riv+12] Marzia Rivi, Luigi Calori, Giuseppa Muscianisi, and Vladimir Slavnic. "In-situ visualization: State-of-the-art and some use cases". In: *PRACE White Paper* (2012) (cit. on p. 18).

[RJR00] K.A. Robbins, C. Jeffrey, and S. Robbins. "Visualization of Splitting and Merging Processes". In: *Journal of Visual Languages and Computing* 11.6 (2000) (cit. on p. 85).

[RKB06] Paruj Ratanaworabhan, Jian Ke, and Martin Burtscher. "Fast Lossless Compression of Scientific Floating-Point Data". In: *IEEE Data Compression*. 2006 (cit. on p. 57).

[RPS01] Freek Reinders, Frits H. Post, and Hans J.W. Spoelder. "Visualization of time-dependent data with feature tracking and event detection". In: *The Visual Computer* 17.1 (Feb. 2001) (cit. on p. 84).

[RWS11] V. Robins, P. Wood, and A. Sheppard. "Theory and algorithms for constructing discrete Morse complexes from grayscale digital images". In: *IEEE Trans. on Pat. Ana. and Mach. Int.* (2011) (cit. on p. 44).

[SA14] TOTAL S.A. Sismage CIG internal document. 2014 (cit. on p. 10).

[SA16] TOTAL S.A. internal document. 2016 (cit. on p. 18).

[Sam+94] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. "Visualizing features and tracking their evolution". In: *Computer* 27.7 (1994) (cit. on p. 84).

[SB06] B. S. Sohn and Chandrajit Bajaj. "Time-varying contour topology". In: *IEEE Transactions on Visualization and Computer Graphics* 12.1 (2006) (cit. on pp. 83, 84, 100, 112, 114, 130).

[SCS12] J Santiago, M Claeys-Bruno, and M Sergent. "Construction of space-filling designs using WSP algorithm for high dimensional spaces". In: *Chemometrics and Intelligent Laboratory Systems* 113 (2012) (cit. on p. 127).

[Sew17] Julian Seward. *Bzip2 data compressor*. `http://www.bzip.org`. 2017 (cit. on pp. 64, 65).

[SF83] Alberto Sanfeliu and King-Sun Fu. "A distance measure between attributed relational graphs for pattern recognition". In: *IEEE transactions on systems, man, and cybernetics* 3 (1983) (cit. on p. 138).

[SH05] Barton T Stander and John C Hart. "Guaranteeing the topology of an implicit surface polygonization for interactive modeling". In: *ACM SIGGRAPH 2005 Courses*. ACM. 2005 (cit. on p. 25).

[Shi+09] Kuangyu Shi, Holger Theisel, Helwig Hauser, Tino Weinkauf, Kresimir Matkovic, Hans-Christian Hege, and Hans-Peter Seidel. "Path Line Attributes - an Information Visualization Approach to Analyzing the Dynamic Behavior of 3D Time-Dependent Flow Fields". In: *Topology-Based Methods in Visualization II*. Springer, 2009 (cit. on p. 84).

[Shi+16] Nithin Shivashankar, Pratyush Pranav, Vijay Natarajan, Rien van de Weygaert, EG Patrick Bos, and Steven Rieder. "Felix: A Topology Based Framework for Visual Exploration of Cosmic Filaments". In: *IEEE TVCG* (2016) (cit. on p. 17).

[SIK12] Jyotsna Sharma, Sarah B Inwood, and Anthony Kovscek. "Experiments and analysis of multiscale viscous fingering during forced imbibition". In: *SPE Journal* 17.04 (2012) (cit. on p. 113).

[Sil95] Deborah Silver. "Object-Oriented Visualization". In: *IEEE Comput. Graph. Appl.* 15.3 (May 1995) (cit. on pp. 84, 114, 130).

[Ska+09] Arne Skauge, K Sorbie, Per Arne Ormehaug, and T Skauge. "Experimental and numerical modeling studies of viscous unstable displacement". In: *IOR 2009-15th European Symposium on Improved Oil Recovery*. 2009 (cit. on p. 126).

[Ska+11] A Skauge, T Horgen, B Noremark, and B Vik. "Experimental Studies of Unstable Displacement in Carbonate and Sandstone Material". In: *IOR 2011-16th European Symposium on Improved Oil Recovery*. 2011 (cit. on p. 113).

[Ska+12] Arne Skauge, Per Arne Ormehaug, Tiril Gurholt, Bartek Vik, Igor Bondino, and Gerald Hamon. "2-D Visualisation of unstable waterflood and polymer flood for displacement of heavy oil". In: *SPE Improved Oil Recovery Symposium*. Society of Petroleum Engineers. 2012 (cit. on pp. 126, 127).

[Ska+14] Tormod Skauge, Bartek Florczyk Vik, Per Arne Ormehaug, Berit K Jatten, Vegard Kippe, Ingun Skjevrak, Dag Chun Standnes, Knut Uleberg, and Arne Skauge. "Polymer flood at adverse mobility ratio in 2D flow by X-ray visualization". In: *SPE EOR Conference at Oil and Gas West Asia*. Society of Petroleum Engineers. 2014 (cit. on pp. 113, 126).

[SKK91] Yoshihisa Shinagawa, Tosiyasu L Kunii, and Yannick L Kergosien. "Surface coding based on Morse theory". In: *IEEE Computer Graphics and Applications* 5 (1991) (cit. on p. 25).

[Sol+15] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas J. Guibas. "Convolutional wasserstein distances: efficient optimal transportation on geometric domains". In: *ACM Trans. Graph.* 34.4 (2015) (cit. on pp. 86, 114).

[Sol+16] Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. "Entropic Metric Alignment for Correspondence Problems". In: *ACM Trans. Graph.* 35.4 (July 2016) (cit. on pp. 86, 114).

[Sol+18a] Maxime Soler, Mélanie Plainchault, Bruno Conche, and Juilen Tierny. "Lifted Wasserstein Matcher for Fast and Robust Topology Tracking". In: *IEEE Symposium on Large Data Analysis and Visualization*. 2018 (cit. on pp. 4, 17, 82, 112, 114, 129).

[Sol+18b] Maxime Soler, Mélanie Plainchault, Bruno Conche, and Julien Tierny. "Topologically controlled lossy compression". In: *2018 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE. 2018 (cit. on pp. 4, 17, 54).

[Sol+19] Maxime Soler, Mélanie Plainchault, Martin Petitfrere, Gilles Darche, Bruno Conche, and Julien Tierny. "Ranking Viscous Finger Simulations to an Acquired Ground Truth with Topology-aware Matchings". In: (2019) (cit. on pp. 5, 110).

[Son+14] S. W. Son, Z. Chen, W. Hendrix, A. Agrawal, W-k. Liao, and A. Choudhary. "Data Compression for the Exascale Computing Era - Survey". In: *Supercomputing Frontiers and Innovations* 1.2 (2014) (cit. on pp. 17, 55).

[Sou] Cyprien Soulaine. *On the origin of Darcy's law* (cit. on p. 11).

[Sou11] T. Sousbie. "The Persistent Cosmic Web and its Filamentary Structure: Theory and Implementations". In: *Royal Astronomical Society* 414.1 (2011) (cit. on p. 17).

[Sri+18] Raghavendra Sridharamurthy, Talha Bin Masood, Adhitya Kamakshidasan, and Vijay Natarajan. "Edit Distance between Merge Trees". In: *IEEE transactions on visualization and computer graphics* (2018) (cit. on p. 138).

[SSS74] Ivan E Sutherland, Robert F Sproull, and Robert A Schumacker. "A characterization of ten hidden-surface algorithms". In: *ACM Computing Surveys (CSUR)* 6.1 (1974) (cit. on p. 15).

[ST58] Philip Geoffrey Saffman and Geoffrey Ingram Taylor. "The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 245.1242 (1958) (cit. on p. 113).

[SW03] Jens Schneider and Rüdiger Westermann. "Compression Domain Volume Rendering". In: *IEEE VIS*. 2003 (cit. on p. 57).

[SW17] H. Saikia and T. Weinkauf. "Global Feature Tracking and Similarity Estimation in Time-Dependent Scalar Fields". In: *Comput. Graph. Forum* 36.3 (June 2017) (cit. on pp. 84, 100, 114, 130).

[SW96] D. Silver and X. Wang. "Volume tracking". In: *Visualization '96. Proceedings.* Oct. 1996 (cit. on pp. 84, 114).

[SW97] D. Silver and Xin Wang. "Tracking and visualizing turbulent 3D features". In: *IEEE Transactions on Visualization and Computer Graphics* 3.2 (Apr. 1997) (cit. on pp. 84, 114).

[SW98] D. Silver and X. Wang. "Tracking scalar features in unstructured data sets". In: *Visualization '98. Proceedings*. Oct. 1998 (cit. on pp. 84, 114).

[SW99] Deborah Silver and Xin Wang. "Visualizing evolving scalar phenomena". In: *Future Generation Computer Systems* 15.1 (1999) (cit. on pp. 84, 114).

[TC16] J. Tierny and H. Carr. "Jacobi Fiber surfaces for Bivariate Reeb space computation". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016) (cit. on p. 44).

[Tea11] NASA Goddard/MODIS Rapid Response Team. *NASA's Aqua Satellite View of Cloud-Free Japan on April 5, 2011*. 2011 (cit. on p. 14).

[Ter+17] Théophile Terraz, Alejandro Ribes, Yvan Fournier, Bertrand Iooss, and Bruno Raffin. "Melissa: large scale in transit sensitivity analysis avoiding intermediate files". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM. 2017 (cit. on p. 19).

[TGP14] J. Tierny, D. Guenther, and V. Pascucci. "Optimal General Simplification of Scalar Fields on Surfaces". In: *Topological and Statistical Methods for Complex Data*. Springer, 2014 (cit. on p. 77).

[The+04] H. Theisel, T. Weinkauf, H. C. Hege, and H. P. Seidel. "Stream line and path line oriented topology for 2D time-dependent vector fields". In: *IEEE Visualization 2004*. Oct. 2004 (cit. on p. 84).

[The+05] H. Theisel, T. Weinkauf, H. C. Hege, and H. P. Seidel. "Topological methods for 2D time-dependent vector fields based on stream lines and path lines". In: *IEEE Transactions on Visualization and Computer Graphics* 11.4 (July 2005) (cit. on p. 84).

[Thi+10] Marco Thiele, Roderick Batycky, Stefan Pöllitzer, and Torsten Clemens. "Polymer-flood modeling using streamlines". In: *SPE Reservoir Evaluation & Engineering* 13.02 (2010) (cit. on p. 12).

[Tho60] JHM Thomeer. "Introduction of a pore geometrical factor defined by the capillary pressure curve". In: *Journal of Petroleum Technology* (1960) (cit. on p. 115).

[Tie+09] Julien Tierny, Attila Gyulassy, Eddie Simon, and Valerio Pascucci. "Loop Surgery for Volumetric Meshes: Reeb Graphs Reduced to Contour Trees". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009) (cit. on pp. 17, 44).

[Tie+17] Julien Tierny, Guillaume Favelier, Joshua A. Levine, Charles Gueunet, and Michael Michaux. "The Topology ToolKit". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017) (cit. on pp. 30, 68, 99, 107, 129, 136).

[Tie16] Julien Tierny. "Contributions to Topological Data Analysis for Scientific Visualization". PhD thesis. UPMC-Paris 6 Sorbonne Universités, 2016 (cit. on pp. 21, 51).

[TJC16] Takeshi Tsuji, Fei Jiang, and Kenneth T Christensen. "Characterization of immiscible fluid displacement processes with various capillary numbers and viscosity ratios in 3D natural sandstone". In: *Advances in Water Resources* (2016) (cit. on p. 113).

[TN14] D. M. Thomas and V. Natarajan. "Multiscale Symmetry Detection in Scalar Fields by Clustering Contours". In: *IEEE Transactions on Visualization and Computer Graphics* (2014) (cit. on p. 107).

[TP12] Julien Tierny and Valerio Pascucci. "Generalized Topological Simplification of Scalar Fields on Surfaces". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012) (cit. on pp. 36, 49, 61–66, 68, 71, 73, 77, 97, 103).

[TR98] Gabriel Taubin and Jarek Rossignac. "Geometric Compression Through Topological Surgery". In: *ACM Transactions on Graphics* 17.2 (1998) (cit. on p. 59).

[Tri+02] X. Tricoche, T. Wischgoll, R. G Scheuermann, and H. Hagen. "Topology tracking for the visualization of time-dependent two-dimensional flows". In: *Computers and Graphics* 26.2 (2002) (cit. on p. 84).

[TS03] H. Theisel and H.-P. Seidel. "Feature Flow Fields". In: *Procs. of the Symp. on Data Visualisation 2003*. VISSYM '03. Grenoble, France: Eurographics Association, 2003 (cit. on p. 84).

[TSJ15] Mathias Trojer, Michael L Szulczewski, and Ruben Juanes. "Stabilizing fluid-fluid displacements in porous media through wettability alteration". In: *Physical Review Applied* 3.5 (2015) (cit. on p. 113).

[TV98] S. Tarasov and M. Vyali. "Construction of contour trees in 3D in O(n log n) steps". In: *Proc. of ACM Symposium on Computational Geometry*. 1998 (cit. on p. 47).

[TW84] William P Thurston and Jeffrey R Weeks. "The mathematics of three-dimensional manifolds". In: *Scientific American* 251.1 (1984) (cit. on p. 24).

[Unn87a] Kazutaka Unno. *Cartography in Japan - Nagakubo Sekisui's Kaisei Nihon Yochi Rotei Zenzu*. Ed. by John Brian Harley, David Woodward, and G Malcolm Lewis. The history of cartography. 1987 (cit. on p. 14).

[Unn87b] Kazutaka Unno. *Cartography in Japan - The map of Japan at Jōtoku Temple*. Ed. by John Brian Harley, David Woodward, and G Malcolm Lewis. The history of cartography. 1987 (cit. on p. 14).

[Vil08] Cédric Villani. *Optimal Transport: Old and New*. 2009th ed. Springer, Sept. 2008 (cit. on p. 85).

[Wea13] John Weaver. *An implementation of the Kuhn–Munkres algorithm*. `https://github.com/saebyn/munkres-cpp`. 2013 (cit. on pp. 86, 105).

[Web+11] Gunther Weber, Peer-Timo Bremer, Marcus Day, John Bell, and Valerio Pascucci. "Feature Tracking Using Reeb Graphs". In: *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011 (cit. on p. 85).

[Whi05] Turner Whitted. "An improved illumination model for shaded display". In: *ACM Siggraph 2005 Courses*. ACM. 2005 (cit. on p. 16).

[Whi86] Stephen Whitaker. "Flow in porous media I: A theoretical derivation of Darcy's law". In: *Transport in porous media* 1.1 (1986) (cit. on p. 115).

[Wid+15] W. Widanagamaachchi, J. Chen, P. Klacansky, V. Pascucci, H. Kolla, A. Bhagatwala, and P. T. Bremer. "Tracking features in embedded surfaces: Understanding extinction in turbulent combustion". In: *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)*. Oct. 2015 (cit. on p. 85).

[Wil17] Qi Wu Will Usher. *Topology Guided Volume Exploration*. `https://github.com/Twinklebear/topo-vol`. 2017 (cit. on p. 17).

[WT17] Chaoli Wang and Jun Tao. "Graphs in Scientific Visualization: A Survey". In: *Computer Graphics Forum* 36.1 (2017) (cit. on p. 85).

[XBJ16] Qingrong Xiong, Todor G Baychev, and Andrey P Jivkov. "Review of pore network modelling of porous media: experimental characterisations, network constructions and applications to reactive transport". In: *Journal of contaminant hydrology* 192 (2016) (cit. on p. 11).

[Yu+10] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K. L. Ma. "In Situ Visualization for Large-Scale Combustion Simulations". In: *IEEE Computer Graphics and Applications* 30.3 (May 2010) (cit. on p. 18).

[ZFL07] Fumin Zhang, E. Fiorelli, and N. E. Leonard. "Exploring scalar fields using multiple sensor platforms: Tracking level curves". In: *2007 46th IEEE Conference on Decision and Control*. Dec. 2007 (cit. on p. 83).

[Zha+12] F. Zhang, S. Lasluisa, T. Jin, I. Rodero, H. Bui, and M. Parashar. "In-situ Feature-Based Objects Tracking for Large-Scale Scientific Simulations". In: *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*. Nov. 2012 (cit. on p. 83).

[ZL77] Jacob Ziv and Abraham Lempel. "A Universal Algorithm for Sequential Data Compression". In: *IEEE Trans. on IT* 23.3 (1977) (cit. on pp. 55, 57).

[ZL78] Jacob Ziv and Abraham Lempel. "Compression of Individual Sequences via Variable-Rate Coding". In: *IEEE Trans. on IT* 24.5 (1978) (cit. on pp. 55, 57).

Réduction et Comparaison de Structures d'Intérêt dans des Jeux de Données Massifs par Analyse Topologique

Dans cette thèse, nous proposons différentes méthodes, basées sur l'analyse topologique de données, afin de répondre aux problématiques modernes concernant l'analyse de données scientifiques. Dans le cas de données scalaires, extraire un savoir pertinent à partir de données statiques, de données qui varient dans le temps, ou données d'ensembles s'avère de plus en plus difficile. Nos approches pour la réduction et l'analyse de telles données reposent sur l'idée de définir des structures d'intérêt dans les champs scalaires à l'aide d'abstractions topologiques. Dans un premier temps, nous proposons un nouvel algorithme de compression avec pertes offrant de fortes garanties topologiques, afin de préserver les structures topologiques tout au long de la compression. Des extensions sont proposées pour offrir un contrôle supplémentaire sur l'erreur géométrique. Nous ciblons ensuite les données variables dans le temps en proposant une nouvelle méthode de suivi des structures topologiques, basée sur des métriques topologiques. Ces métriques sont étendues pour être plus robustes. Nous proposons un nouvel algorithme efficace pour les calculer, obtenant des accélérations de plusieurs ordres de grandeur par rapport aux approches de pointe. Enfin, nous appliquons et adaptons nos méthodes aux données d'ensembles relatives à la simulation de réservoir, dans un cas de digitation visqueuse en milieu poreux. Nous adaptons les métriques topologiques pour quantifier l'écart entre les simulations et la vérité terrain, évaluons les métriques proposées avec le retour d'experts, puis implémentons une méthode de classement in-situ pour évaluer la fidélité des simulations.

Large Data Reduction and Structure Comparison with Topological Data Analysis

In this thesis, we propose different methods, based on Topological Data Analysis, in order to address modern problematics concerning the increasing difficulty in the analysis of scientific data. In the case of scalar data defined on geometrical domains, extracting meaningful knowledge from static data, then time-varying data, then ensembles of time-varying data proves increasingly challenging. Our approaches for the reduction and analysis of such data are based on the idea of defining structures of interest in scalar fields as topological features. In a first effort to address data volume growth, we propose a new lossy compression scheme which offers strong topological guarantees, allowing topological features to be preserved throughout compression. The approach is shown to yield high compression factors in practice. Extensions are proposed to offer additional control over the geometrical error. We then target time-varying data by designing a new method for tracking topological features over time, based on topological metrics. We extend the metrics in order to overcome robustness and performance limitations. We propose a new efficient way to compute them, gaining orders of magnitude speedups over state-of-the-art approaches. Finally, we apply and adapt our methods to ensemble data related to reservoir simulation, for modeling viscous fingering in porous media. We show how to capture viscous fingers with topological features, adapt topological metrics for capturing discrepancies between simulation runs and a ground truth, evaluate the proposed metrics with feedback from experts, then implement an in-situ ranking framework for rating the fidelity of simulation runs.