HABILITATION À DIRIGER DES RECHERCHES

UNIVERSITÉ de BORDEAUX
ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE
ED n°39 - Spécialité Informatique

Scientific Visualization to Enhance Understanding of Simulation Data

Fabien VIVODTZEV

Organisme d'accueil : Commissariat à l'énergie atomique et aux énergies alternatives (CEA)
Centre d'études scientifiques et techniques d'Aquitaine (CESTA)

Soutenue le 6 février 2026 devant le jury composé de :

Ingrid Hotz Professeur, Linköping University	Rapporteur
Gerik Scheuermann Professeur, University of Leipzig	Rapporteur
Christoph Garth Professeur, Technische Universität Kaiserslautern	Rapporteur
Raymond Namyst Professeur, Université de Bordeaux	Examinateur
Franck Ledoux Directeur de Recherche CEA	Examinateur
François Mazen Ingénieur, Kitware SAS	Invité

Abstract

This thesis for the Habilitation à Diriger des Recherches (HDR) presents my twenty years of research in scientific visualization. The primary motivation of my work is to contribute to the continuous improvement of methods and tools for the analysis, interpretation and communication of results of numerical simulations for complex physical phenomena. Accelerated by the increasing capabilities of high-performance computing, one of the main challenge is to apprehend the growing mass and complexity of chaotic data. Thus, I have been investigating different subjects, such as feature extractions using Topological Data Analysis, data visualization software development, or immersive rendering to improve communication using 3D computer graphics and Virtual Reality.

Keywords : Scientific Visualization, Topological Data Analysis, Visualization Software, Virtual Reality

---

Résumé

Ce manuscrit d'Habilitation à Diriger des Recherches (HDR) présente mes vingt années de recherche dans le domaine de la visualisation scientifique. La motivation première de mes travaux est de contribuer à l'amélioration continue des méthodes et outils pour l'analyse, l'interprétation et la communication de résultats de simulations numériques décrivant des phénomènes physiques complexes. Accéléré par les capacités croissantes du calcul haute performance, l'un des principaux défis consiste à appréhender la masse et la complexité croissantes des données chaotiques. J'ai donc étudié différents sujets, tels que l'extraction de caractéristiques à l'aide de l'analyse topologique, le développement de logiciels de visualisation ou le rendu immersif afin d'améliorer la communication à l'aide de l'infographie 3D et de la réalité virtuelle.

Mots clés : Visualisation scientifique, Analyse Topologique, logiciels de visualisation, Réalité Virtuelle

« L'intelligence n'est pas la capacité de stocker des informations, mais de savoir où les trouver. »

— Albert Einstein

Contents

CHAPTER 1

Introduction

The primary motivation behind my research over the last twenty years in the field of scientific visualization has been to contribute to the continuous improvement of methods and tools for the analysis, interpretation, communication, and popularization of the results of numerical simulations of complex physical phenomena. Accelerated by the increasing capabilities of high-performance computing, one of the main challenges is to apprehend the growing mass and complexity of chaotic data.

Scientific visualization for simulation data is at the interface of many fields, including software engineering, applied mathematics, and computer graphics. The interdisciplinarity of this field is accentuated by the fact that visualization is applied to many areas of physics, as well as to many interlocutors ranging from expert physicists to external audiences interested in the scientific popularization of complex phenomena. My research in this field focuses on the different stages of data transformation for numerical simulation, integrating the scientific, technological, and human considerations inherent in the transmission of information by visual language. The domains addressed in these research projects are closely linked to the activities carried out at the French Alternative Energies and Atomic Energy Commission (CEA). The CEA uses a simulation-based approach to guaranteeing the performance of weapon systems, contributing to French nuclear deterrence. This simulation approach involves modeling extreme physical phenomena, simulating them numerically on supercomputers using calculation codes, validating them experimentally, and visualizing the results in many fields such as electromagnetism, fast dynamics, or aerodynamics. As described in chapter 2, I have been leading most of my research in visualization at the CEA.

The fundamental concepts used in my projects come from the domain of the Topological Data Analysis (TDA). The chapter 3 introduced these generic, robust, and efficient topological techniques for extracting hidden structural features from complex data. They offer an alternative

to traditional approaches based on image comparison, spectral analysis, or machine learning.

As part of the numerical simulation activities carried out at the CEA, my PhD work focused on a multi-resolution visualization strategy for very large unstructured meshes based on mesh simplification detailed in chapter 4. To guarantee the relevance of low-resolution 2D and 3D visualization of simulation results, particularly at the level of geometric features embedded in meshes, computational topological criteria have been proposed. The topology-preserving mesh simplification introduced is based on a domain extension by particular geometric elements. This innovative idea has been implemented in several fields. In electromagnetism, for example, it has made it easier to visualize the computation of an object's radar equivalent surface solved by Maxwell's equations. In neuroscience, it has been integrated into a pipeline for processing human cortexes as part of a surface comparison program used to diagnose cerebral diseases.

Computational topology is a more efficient approach than merely handling features of the geometric domain. Therefore, the scientific visualization community took an early view of this approach to numerical data such as scalar fields or vector fields used to describe physical entities (pressure, temperatures, velocity. . .). The chapter 5 focuses on TDA to help the mathematical modeling of instabilities. The development of a simulation code based on numerical concepts such as the Navier-Stokes equations requires the evaluation of numerous numerical ingredients, including Riemann solvers, integration schemes, and meshes. Therefore, based on known fluid mechanics assumptions, topological analysis protocols have been proposed to compare these large turbulent datasets generated due to the variability of the numerical components. Several topological abstractions have been proposed on domain-specific scalar fields to classify and compare the results to help the developers choose the best numerical ingredients in their simulation codes.

Many other chaotic data have been observed in a variety of real-world datasets, encompassing diverse fields such as aeronautics, climate modeling, material sciences, or astronomy. As described in several projects detailed in chapter 6, topological abstractions can be found for many application domain. For 2D turbulent flows, vortex tracking can be achieved with a topological signature and a measure of distance between vortices in a topological space. A similar method can be investigated for segmenting 3D vortices such as Taylor-Green vortices, which can be coupled to in situ visualizations. A key aspect of those topological segmentations is the study of their physical characterizations with traditional methods such as the kinetic energy spectrum of local turbulent vortices. The Rayleigh-Taylor hydrodynamic instabilities found during Inertial Confinement Fusion (ICF) need to be overcome to reach higher energy gain. Feature detection of these bubbles on x-rays acquired during laser shooting can be achieved with TDA.

Handling chaotic data and building simplified topological abstractions requires the evolution of high-performance visualization software as described in chapter 7. The key approach is to tightly couple the visualization software to the remote High-Performance Computing supercomputers. To compensate for the distance between data production sites and exploitation workstations, remote visualization techniques have been evaluated. Depending on the analysis software used by scientists, different solutions have been investigated, such as an approach using the client/server capabilities of the visualization software VisIt via a secure SSH tunnel adapted for large data volumes. The direct integration of rendering views into simulation platforms

is also to be considered with a visualization library such as VTK and its Java interface. To handle parallel visualization on large volumes of data, a work on data formats and a parallel file management strategy allows exploration of billions of cells on supercomputers. Immersive visualizations coupled to the HPC environment also need to be considered for certain applications. The virtual reality software environment needs to be adapted to the HPC supercomputers because the networks between them are often restricted or not connected to the Internet.

This thesis summarize twenty years of research in scientific visualization in order to defend the highest scientific qualification in France called *Habilitation à Diriger des Recherches* to direct research. This habilitation allows to supervise doctoral candidates and young researchers independently and to be eligible, in most disciplines, for full professor positions at universities. The guideline of my research is the improvement of the visualization of numerical simulation such as the ones led at CEA. Thus I have been investigating many subjects from the extraction of features of interest using topological data analysis (TDA), the exploitation of these data using software and data processing (HPDA) or immersive rendering to improve communication using 3D computer graphics and virtual reality (VR).

CHAPTER 2

Professional and Scientific Activities

This chapter provides a synopsis of my professional activities as a research engineer and team leader at CEA. It details my publications, my internal scientific animation, the supervision and the scientific responsibilities I had in the domain of scientific visualization over the past decades.

2.1 Activities at CEA

Deputy Head of the Muliphysics Simulation Department

2021 - present The activities of the deputy head of this department consist of (i) managing a team of approximately 50 research engineers, most of them Doctors including 8 researchers with an *Habilitation à Diriger des Recherches*, supervising an average of 45 students, consisting of about 5 post-docs, 20 PhD students, 15 interns, and 5 apprentices ; (ii) piloting the projects for the development, and validation of simulation codes and digital tools to design and guarantee the performance of Nuclear Warheads; (iii) managing a budget of several million euros per year for study contracts, simulation software and non-permanent staff salaries; (iv) setting up collaboration and study contracts with research organizations and industrial partners.

Senior Expert at CEA in Scientific Visualization

2016 - present The senior expert is called upon to guide the scientific activities related to his field. The CEA expertise track has 4 levels. My expertise (level 3/4) concerns data analysis in the simulation and performance guarantee chains for Nuclear Warheads in a high-performance computing environment. In particular, the axes of expertise on which I am consulted concern

the scientific visualization of large volumes of data, topological data analysis, software strategy for remote and parallel visualization, interoperability of software and exploitation formats, monitoring of academic collaborations, and the production of 3D computer graphics films.

Head of the Software Development for Simulation Laboratory

2014 - 2021 The activities of the head of the laboratory consist of managing and organizing the implementation of the scientific computing environment for simulation software. It consists of approximately 10 research engineers in scientific computing and software engineering. It hosts PhD students, research engineers, and final-year interns. Based on software engineering methods, the laboratory designs, develops, verifies, deploys, and maintains simulation software and associated tools such as meshers, visualization tools, computing platforms, and databases.

Research Engineer in Scientific Visualization

2009 - 2014 Research engineer within the Software Development laboratory, in charge of scientific visualization software and the exploitation of simulation software chains for different physics such as fluid mechanics, fast dynamics, electromagnetism, and thermomechanics.

Physical Models Engineer for Technical-Operational Simulations

2006 - 2009 Research engineer within the Software Development laboratory in the field of software engineering for the reverse engineering of physical models within a technical-operational analysis simulator.

2.2 Publications

Thesis (1)

[1] Hiérarchisation et visualisation multirésolution de résultats issus de codes de simulation, Ph.D. thesis, Université Joseph Fourier, Grenoble, 2005, in collaboration with CEA and EVASION Project (UJF, INPG, CNRS, INRIA), Director: Georges-Pierre Bonneau (INRIA, Université Joseph Fourier, Grenoble), Supervisor: Paul Le Texier (CEA), Chairman: Pascal Guitton (LaBRI - INRIA - Université Bordeaux), Reviewer: Jean-Michel Dischler (LSIIT - Université Louis Pasteur, Strasbourg), Reviewer Wilfrid Lefer (LIUPPA – Université de Pau et des Pays de l'Adour), Examiner: Jean-Philippe Nominé (CEA), 2005.

Book Edition (1)

[2] Topological and Statistical Methods for Complex Data, Tackling Large-Scale, High-Dimensional and Multivariate Data Sets, Editors: J. Bennett, F. Vivodtzev and V. Pascucci, Springer,

"Mathematics and Visualization" series, ISBN: 978-3-662-44900-4, 297 pages, Springer, 2015.

Book Chapters (3)

[3] Piecewise Polynomial Monotonic Interpolation of 2D Gridded Data – L. Allemand-Giorgis, G.-P. Bonneau, S. Hahmann and F.Vivodtzev, Topological and Statistical Methods for Complex Data - Tackling Large-Scale, High-Dimensional and Multivariate Data Sets, "Mathematics and Visualization" series, Editors J. Bennett, F. Vivodtzev, V.Pascucci, Springer, 2014.

[4] Substructure Topology Preserving Simplification of Tetrahedral Meshes - F. Vivodtzev, G.-P Bonneau, S. Hahmann, H. Hagen, Topological Methods in Data Analysis and Visualization, Mathematics and Visualization, Springer Edition, 2011.

[5] Brain Mapping Using Topology Graphs Obtained by Surface Segmentation - F. Vivodtzev, L. Linsen, B. Hamann, K.I. Joy, B. Olshausen, Chapter in the book Scientific Visualization: The Visual Extraction of Knowledge from Data by G.-P. Bonneau, T. Ertl, G. M. Nielson, ISBN 3-540-26066-8, Springer-Verlag, Heidelberg, Germany, 2003.

Publications in Peer-Reviewed Journals (1)

[6] Topology Preserving Simplification of 2D Non-Manifold Meshes with Embedded Structures- F. Vivodtzev, G.-P. Bonneau, P. Le Texier - The Visual Computer, 21(8), Springer-Verlag, Heidelberg, Germany, 2005.

Publications in Peer-Reviewed Conference Proceedings (12)

[7] Topological Representations for Data Analysis: Use Cases on Instabilities and Turbulences, F. Vivodtzev, F. Nauleau, A. Casner, J. Tierny, SNA + MC 2024, proceedings of Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo, Paris, France, October 20-24, 2024, published in EPJ Web of Conferences, 2024.

[8] Identifying Locally Turbulent Vortices within Instabilities, F. Vivodtzev, F. Nauleau, J.-P. Braeunig, J. Tierny, LDAV 2024, 2024 IEEE 14th Symposium on Large Data Analysis and Visualization, (short paper) in conjunction with IEEE VIS 2024, St. Pete Beach, Florida, USA, arXiv:2408.12662, October 13-18, 2024.

[9] Topological Data Analysis of 3D Ablative Rayleigh-Taylor Instability Dataset for Automatic Segmentation, F. Vivodtzev, A. Casner, L. Masse, L. Ceurvorst, S. Khan, V. Smalyuk, LDAV 2023, 2023 IEEE 13th Symposium on Large Data Analysis and Visualization (short paper), in conjunction with IEEE VIS 2023, Melbourne, Australia, 22-27 October, 2023.

[10] Topological Analysis of Ensembles of Hydrodynamic Turbulent Flows - An Experimental Study, Florent Nauleau, F. Vivodtzev, T. Bridel-Bertomeu, H. Beaugendre, J. Tierny, LDAV 2022, proceedings of 2022 IEEE 12th Symposium on Large Data Analysis and Visualization in conjunction with IEEE VIS 2022, Oklahoma City, USA, arXiv:2207.14080, 16-21 October, 2022.

[11] In Situ Segmentation of Turbulent Flow with Topology Data Analysis, F. Nauleau, B. Fovet, F. Vivodtzev, SIGGRAPH'22, ACM SIGGRAPH 2022, (short paper) ACM SIGGRAPH 2022, Vancouver, Canada, article No.:61, ISBN:9781450393614, 8-11 August, 2022.

[12] Immersed Boundaries In Hypersonic Flows With Considerations About High-Fidelity And Massive Parallelism, F. Nauleau, T. Bridel-Bertomeu, H. Beaugendre, F. Vivodtzev, ECCOMAS Congress 2022, Proceedings of the 8th European Congress on Computational Methods in Applied Sciences and Engineering, DOI: 10.23967/eccomas.2022.142, Oslo, Norway, 5-9 June, 2022.

[13] Topological Analysis of High Velocity Turbulent Flow – Th. Bridel-Berthomeu, B. Fovet, J. Tierny, F. Vivodtzev, LDAV 2019, 2019 IEEE 9th Symposium on Large Data Analysis and Visualization, (short paper) Vancouver, Canada, 2019.

[14] Remote Visualization of Large Scale Fast Dynamic Simulations in a HPC Context, F. Vivodtzev and I. Bertron, Proceedings of 2014 IEEE 4th Symposium on Large Data Analysis and Visualization (short paper), Paris, November 2014.

[15] Open source Software to Visualize Complex Data on Remote CEA's Supercomputing Facilities, F. Vivodtzev and T. Carrard, Proceedings of the Workshop on Open Source in Visualization at EUROVIS 2014 (short paper), Swansea, Wales, 2014.

[16] VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data - H.Childs, E.Brugger, B.Whitlock, J.Meredith, S. Ahern, K.Bonnell, M.Miller, G.Weber, C.Harrison, D.Pugmire, T.Fogal, C. Garth, A.Sanderson, E.Wes Bethel, M.Durant, D. Camp, J.Favre, O.Rubel, P. Navratil, M. Wheelera, P. Selbya and F.Vivodtzev, Proceedings of Scientific Discovery through Advanced Computing Review, SciDac, 2011.

[17] Automatic Feature-based Surface Mapping for Brain Cortices - F. Vivodtzev, D. Wiley, L. Linsen, J. Jones, N. Amenta, B. Hamann, K.I. Joy - In R.F Erbacher, J.C Roberts, M.T Groehn, K. Boerner, M.C Hao and P.C Wong editors, Visualization and Data Analysis 2006, Bellingham, Washington. SPIE - The International Society for Optical Engineering, 2006.

[18] Hierachical Isosurface Segmentation Based on Discrete Curvature - F.Vivodtzev, L.Linsen, G.-P.Bonneau, B.Hamann, K.I. Joy, B. Olshausen - In G.-P. Bonneau, S. Hahmann, C. D. Hansen, editors, Proceedings of VisSym03, Eurographics, IEEE TVCG Symposium on Visualization. New York, New York, 2003.

Invited Presentations (2)

[19] Visualisation at CEA: HyperTreeGrid and Other Physics, M. Stauffert, F. Vivodtzev, ParaView User Day Europe 2024, presentation at the Focus session, ENS Lyon, France, September 26th, 2024.

[20] Meeting on Topological Analysis and Visualization of Vector Fields, Fabien Vivodtzev, CIRM, Luminy, January 27-29, 2013.

Thematic Conference Presentations without Proceedings (5)

[21] On the road to build an open-source VR workflow in a Linux restricted network environment, F. Vivodtzev, A. Gomez, Sandia 5th Annual XR Conference, Sandia National Laboratories, August 1-4 2023.

[22] Topological Data Analysis for numerical method comparisons of 2D turbulent flows, F. Nauleau, Th. Bridel-Bertomeu, F. Vivodtzev, H. Beaugendre, J. Tierny conference Journée Visu, France, 2021.

[23] A k-exact ENO-like least-square reconstruction for compressible flows applied to sharp immersed boundaries, T. Bridel-Bertomeu, F. Nauleau, H. Beaugendre, F. Vivodtzev, 10th French biennial in Applied Mathematics and Industry, SMAI, 2021.

[24] Framework for Automatic Meshing Applied to 2D and 3D Electromagnetic Simulations, F. Vivodtzev, T. Hocquellet, M. Lecouvez, ECCOMAS 2020, presentation at the 14th WCCM and European Community on Computational Methods in Applied Sciences Congress 2020.

[25] User experience feedback: how to make remote visualization issues compatible in a highly secure environment, D.Nassiet, F.Vivodtzev, Forum TeraTec 2012.

[26] Brain Atlas Mapping, A. R. Fuller, B. Hamann, K. I. Joy, Edward G. Jones, L. Linsen, B. A. Olshausen, T. W. Slankard, J. Stone, F. Vivodtzev, G. H. Weber, D. F. Wiley, P. C. Yau, presented at the Society for Neuroscience, 34th annual meeting, San Diego, California, 2003.

Technical Reports (3)

[27] Technical report on open-source VR prototype in a Linux restricted network environment, A.Gomez, F.Vivodtzev, https://fabien-vivodtzev.github.io/pdf/article-2023-xr.pdf, 2023.

[28] Analyzing complex features in ensemble data with the Topology ToolKit in ParaView, J. Tierny, F. Vivodtzev, C. Gueunet, Kitware blog, https://www.kitware.com/analyzing-complex-features-in-ensemble-data-with-the-topology-toolkit-in-paraview/, 2023.

[29] Multi-resolution visualization of simulation results - F.Vivodtzev - CHOCS avancées 2009, ISSN 1961-7399, CEA, 2009.

Software Contribution (1)

[30] Open-source contribution to VTK in collaboration with Kitware for the Java API allowing the encapsulation of a generic rendering window regardless of its origin (awt, swt, swing). https://github.com/Kitware/VTK/blob/master/Wrapping/Java/vtk/rendering/awt/vtkPanel Component.java, 2012.

2.3 Internal Scientific Animation at CEA

Contribution to CEA Scientific Evaluations (2)

Some CEA research topics are evaluated every 6 years in the same way as HCERES evaluations. The evaluation committee, made up of researchers from the scientific community, particularly highlights the international recognition and great attractiveness of CEA regarding its scientific potential and research activities. It highlights strengths, areas for improvement, and makes recommendations for the future. During these evaluations, I presented scientific advances related to scientific visualization and meshes.

[31] Scientific visualization and data exploitation from numerical simulation, High Performance Computing Scientific Evaluation, CEA/DIF, Bruyères Le Châtel, F.Vivodtzev, 2022.

[32] Towards a multi-scale design tool: geometry, meshing, studies, and visualization, Electromagnetism Scientific Evaluation, CEA/CESTA, Le Barp, F. Vivodtzev, 2021.

Contribution to the scientific strategy for the CEA's Simulation Department (1)

The scientific strategy of CEA is regularly updated to follow the evolution of program needs in development and the actions to be taken under the scientific activity. Within the CEA Simulation and Warhead Performance Guarantee department, the Scientific Orientation Plan is a document that capitalizes on this entire scientific strategy, for which I co-drafted the chapter on data processing and scientific visualization.

[33] Numerical methods and software development for simulation, Scientific Orientation Plan of the Simulation and Warhead Performance Guarantee Department 2019 – P-H. Maire, F.Vivodtzev, 2019.

Production of 3D Computer Graphics Movies for CEA Scientific Outreach

This scientific outreach activity has led to the production of about thirty scientific and educational movies to improve the understanding of physical phenomena and the presentation of innovative concepts. Techniques for adapting 3D models (CAD and meshes) and 3D computer graphics techniques and tools (Blender) are used to enhance scientific visualizations. This scientific outreach relies on a scripting process specific to each physical domain. The majority of these films are classified.

2024 Presentation of flight tests: 10 films presenting the phases of a flight test of a strategic M51 ballistic missile without a nuclear warhead. They have been presented by the CEA during the briefing before the launch to the various industrial and state partners since 2009.

2018 Operation of a safety device - Classified animated film on the operating mode of a safety device of a weapon system, 2018.

2018 Design and performance guarantee of an Oceanic Nuclear Warhead in 3D – a classified 3D film illustrating many physics necessary for the design and performance guarantee of an Oceanic Nuclear Warhead.

2012 2012 Re-entry STD Certification - Classified film presenting the results of the calculation codes for the 2012 re-entry standard presented for certification, 2012.

2011 Science and Defense Award - Film for the DGA Science and Defense award given to ASTRIUM, EADS, and the CEA showing the data used in the context of HPC simulations in electromagnetism, 2011.

2.4 Scientific Responsibilities

Organization of International Conferences (2)

2024 Main organizer of the annual meeting of the collaboration between the NNSA (National Nuclear Security Administration) and the CEA in computing and High-Performance Computing, June 24-27, 2024, Bordeaux.

2013 Main organizer of the seminar Workshop on the analysis of large-scale, high-dimensional and multivariate data using topology and statistics, CEA, Sandia National Laboratories, Kitware SAS, Le Barp.

Organization of National Conferences (2)

2022 Co-organizer of the French Visualization Day (Journée Visu 2022) of the CNRS GDR IG-RV, in collaboration with LaBRI, INRIA, Talence.

2011 Member of the organizing committee of the French Visualization Day (Journée Visu2011) of the CNRS GDR IG-RV, Télécom Paris.

2011 Member of the organizing committee of the CEA/EDF/INRIA Summer School: Building the next generation of visualization software, June 2011, CEA-Cadarache.

Member of Scientific Committees (5)

2020-today Member of the NNSA/CEA collaboration in computing and High-Performance Computing on the topic of visualization (theme leader in 2021).

2020 Member of the Young Expert group of the CEA Digital Mission.

2017 Member of the strategic scientific committee of the INRIA MMG consortium on remeshing tools.

2014 Member of the Industry committee for Eurographics '14, Strasbourg.

2012 Member of the scientific committee of the CNRS GDR IGRV Visualization working group.

Editorial Activities and Reviews (2)

2021 Evaluator of ANR generic calls for projects (CE46 - Numerical models, simulation, applications).

2015 Reviewer for the Journal Computers and Graphics, Elsevier.

Thesis Jury (1)

2011 Thesis examiner for the University of Pau and Pays de l'Adour - Volumetric Reconstruction of Simulation Results Based on Chimera, Minh Duc Huynh, 2011.

Scientific Advisor Supervision (2)

To support the development of skills on certain cutting-edge topics, CEA recruits scientific advisors in different fields. On the topics of meshing and scientific visualization, I set up and monitored the activities of the following advisors.

2024 2025 **Julien Tierny**, Sorbonne Université, Scientific visualization and topological data analysis.

2015 2019 **Pascal Frey**, Sorbonne Université, Methods for adapting geometric data and advanced unstructured meshing techniques.

2.5 Supervision of Research in Scientific Visualization

PhD Student (1)

2020 2023 **Florent NAULEAU** - PhD thesis, Université de Bordeaux – Immersed boundary methods for large-scale simulation of hypersonic re-entry vehicles. Research on numerical schemes for fluid mechanics and topological data analysis of turbulent flows - co-supervision with Héloïse Beaugendre (Université de Bordeaux) - recruited as a research engineer (permanent contract) in meshing and fluid mechanics at CEA.

Research Engineers on Temporary Contracts (6)

To explore certain cutting-edge topics, CEA employs research engineers on temporary contracts. The following has been supported by myself.

2022 **Adrien Gomez** – Research engineer in computer graphics, 3D modeling and virtual reality (18 months). Production of scientific films and setup of virtual reality scenes in a secure environment, recruited as a research engineer (permanent contract) in computer graphics and virtual reality at the CEA.

2017 **Ludovic Père** – Engineer for meshing methodology and multiphysics projection (22 months), recruited as a meshing and electromagnetic stealth studies engineer (permanent contract) at the CEA.

2016 **Benjamin FOVET** - Software engineer for the exploitation of the fast dynamics simulation chain (12 months), recruited as an HPC and software engineer (permanent contract) at the CEA.

2014 **Pauline FOURTILLAN** - Software engineer in data exploitation and visualization (20 months), recruited (permanent contract) as a signal processing engineer at the CEA.

2009 **Sébastien BARBIER** - Research engineer in scientific visualization in a High-Performance Computing environment (12 months), recruited (permanent contract) as a research engineer at the CEA and then at Thermo Fisher.

Master Level Internships (8)

2019 **Clément BROUARD** - PFE Enseirb-Matmeca: Scientific visualization for simulation in a high-performance computing context, recruited as a data processing development engineer at Epsyl of the Alcen group.

2019 **Rémi GARDE** - PFE Ensimag Grenoble INP: Exploration of new scientific visualization techniques in simulation, recruited as a gameplay designer engineer at Asobo Studio.

2013 **Mickaël LESLOURDY** - Master 2 at the Université de Bordeaux: Visualization tools for the exploitation of simulation chains, recruited as a software research and development engineer at Thermo Fisher Scientific.

2012 **Vivien BOUCHONNEAU** - Master 2 at the Université de Bordeaux: Stereoscopy for scientific visualization, recruited as a software engineer in virtual reality at Immersion.

2010 **Romain LEGAY** - Master 2 at the Université de Bordeaux: 3D visualization solutions, recruited as an Inria engineer and then founder of Nurea for medical imaging.

2010 **Bruno BERENGUEL** - Master 2 at the Université de Pau et des Pays de l'Adour: Development of Eclipse scientific visualization plugin integrated into an Eclipse RCP platform.

2009 **Aurélien MORA** - PFE EPSI Bordeaux: The impact of new technologies in scientific visualization, recruited as a lead developer at Thales Systèmes Aéroportés, and as a self-employed blockchain solutions developer.

2008 **Julien BRUNEAU** - PFE Enseirb-Matmeca: Definition and creation of a software workshop for video editing for simulation, continued with a PhD at Inria in software engineering, recruited as a development engineer at Walt Disney Imagineering and then as a research engineer at Inocess.

Summer Schools Teaching (2)

2019, 2016 Sorbonne Université - 30h TD equivalent - Summer school of the Institute for Science, Computing and Data: Explore your data and tell meaningful scientific stories, Roscoff: introduction to scientific visualization, training on VisIt and Blender software, storytelling, development of a C++ reading module in VisIt, development of a VTK Python visualizer.

Initial Training - average of 18 hours of TD equivalent per year over 10 years (8)

2019-2024 Enseirb-Matmeca – 10h TD equivalent - 3rd year Computer Science major in High-Performance Computing and Data Science: Introduction to scientific visualization, training on VisIt, Paraview software, development in VTK Python, and topological data analysis.

2010, 2013 Université de Pau et des Pays de l'Adour – 16h TD equivalent – 3rd year Master of Science and Technology, Computer Science major: Introduction to scientific visualization, the link between computer graphics and visualization, project management in computer science.

CHAPTER 3

Concepts of Computational Topology

Topology is a branch of mathematics that provides methods for describing the connectivity of elements within a space. Euclid's geometry focuses on properties that remain unchanged by rigid displacements in Euclidean space, while topology examines properties that are invariant to continuous deformations and are independent of metric and projective representations. Topology can be used to describe the connectivity of objects by ignoring their spatial embedding and considering only neighborhood and continuity properties. Computational topology is the study of topology based on computational algorithms using computers. This field is central to our work in topological data analysis on large simulation data, improving domain-specific understanding with efficient scientific visualizations.

In 1736, the renowned mathematician Leonhard Euler tackled the "Seven Bridges of Königsberg" problem. The city of Königsberg, then in Prussia, was divided by a river with two large islands, connected to each other and to the mainland by seven bridges.

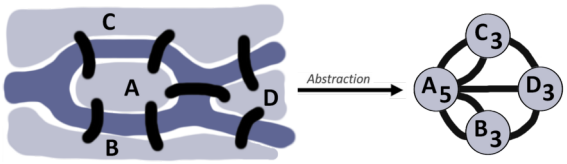


Figure 3.1: Abstraction of the map with a graph for the seven Bridges of Königsberg : How to cross each bridge exactly once and return to the starting point ?

The challenge posed to Euler was to find a path that crossed each bridge exactly once and returned to the starting point. Euler's groundbreaking insight was that the actual distances and shapes were irrelevant, and only the connections mattered. This abstraction to how islands are connected is the core of graph theory and a precursor to topological thinking. By counting

the number of bridges (edges) connected to each landmass (node), Euler could determine the "degree" of each vertex. He famously proved that for such a walk to exist, there could be at most two nodes with an odd degree (number of bridges connected to it). In the Königsberg problem, all four landmasses had an odd number of bridges, making the desired path impossible. This abstraction laid the conceptual groundwork for what would much later become the formal study of topology.

This chapter introduces the formalism underlying data representation and the notion of homology, which allows us to characterize data from a topological point of view. Various topological data representations are presented, such as critical points or persistence diagrams, but for a deeper introduction to topology, the reader should refer to [Edelsbrunner and Harer, 2009] and [Zomorodian, 2010].

3.1 Geometric Input Domain

The input data used in scientific visualization can be expressed as a discretization of the underlying space, called a mesh, which is based on the mathematical principles presented in this section.

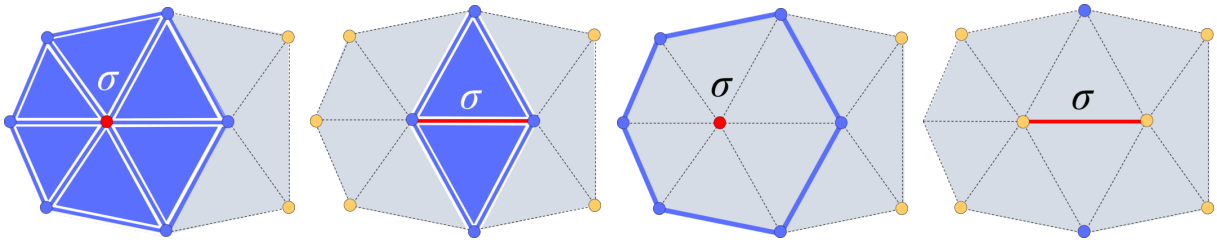
(Topological space) A topology of a set X is a collection of subsets of $\mathcal{T}$ such as: (i) the empty set $\emptyset$ and $\mathcal{T}$ are in $\mathcal{T}$, (ii) the union of any collection of $\mathcal{T}$ is in $\mathcal{T}$, (iii) the intersection of any finite collection of $\mathcal{T}$ is in $\mathcal{T}$. A set X for which a topology $\mathcal{T}$ is defined is called a topological space denoted $\mathbb{X}$. Thus a topology provides information about the connectivity of a space without using specific metrics such as areas, distances, or angles.

(Homeomorphism) An homeomorphism $f : \mathbb{X} \to \mathbb{Y}$ is a bijective application such that f and f^{-1} are continuous. Two topological spaces $\mathbb{X}$ and $\mathbb{Y}$ are said to be homeomorphic if there is a homeomorphism from $\mathbb{X}$ into $\mathbb{Y}$. Thus $\mathbb{X}$ is homeomorphic to $\mathbb{Y}$, $\mathbb{X}$ and $\mathbb{Y}$ have the same topological type, and denoted it as $\mathbb{X} \approx \mathbb{Y}$. The topological spaces connected in the same way can be grouped in the same equivalence class.

(Manifold) A topological space $\mathbb{M}$ is a d-manifold if every element $e \in \mathbb{M}$ has an open neighborhood homeomorphic to an open Euclidean d-ball.

(Simplex) A k-simplex σ is the convex hull of $k+1$ linearly independent points. A k-simplex is a k-dimensional subspace of $\mathbb{R}^d$. A vertex is a 0-simplex of $\mathbb{R}^3$ defined by a point. An edge is a 1-simplex of $\mathbb{R}^3$ corresponding to the set of points on the segment [p1, p2]. It is of dimension 1. A triangle is a 2-simplex of $\mathbb{R}^3$ defined by three non-aligned points. The triangle is the union of the closed polygon $\{p1, p2, p3\}$ and its interior. It is of dimension 2. Finally, a tetrahedron is a 3-simplex of $\mathbb{R}^3$ defined by four non-coplanar points. The four triangles are its faces and the six segments are its edges. The tetrahedron is the union of the closed surface formed by its four faces and its interior. It has dimension 3.

(Face and coface) A k-face τ of the simplex σ is a simplex of dimension k defined by $(k+1)$ vertices of σ. It is denoted $\tau \leq \sigma$. If τ is a face of σ then σ is called a coface of τ. The empty



(a) (b) (c) (d)

Figure 3.2: The star $St(\sigma)$ and link $Lk(\sigma)$ in blue of a simplex σ in red. (a) Star of a vertex. (b) Star of an edge. (c) Link of a vertex. (d) Link of an edge.

set and σ are faces of σ.

(Closure) The closure of a subset of simplices B, noted $\bar{B}$, is the set of all faces of the simplices of B such that $B = \{\tau \in K | \tau \leq \sigma \in B\}$.

(Simplicial complex) A simplicial complex K is a finite collection of non-empty simplices, such that every face τ of a simplex σ is also in K, and any two simplices intersect in a common face or not at all.

(Star) The star of a simplex σ in a simplicial complex K, noted $St(\sigma)$, is the set of all cofaces of σ: $St(\sigma) = \{\tau \in K, \sigma \leq \tau\}$. Intuitively the star of a simplex σ can be interpreted as the smallest combinatorial neighborhood around σ as illustrated on Figure 3.2a and Figure 3.2b.

(Link) The link of σ is the set of all faces of the cofaces from simplices of $St(\sigma)$ that are disjoint from σ: $Lk(\sigma) = \tau \leq \delta, \delta \in St(\sigma), \tau \cap \sigma = \emptyset$. Intuitively, the link of a simplex σ is simply the boundary of the star of σ as illustrated on Figure 3.2c and Figure 3.2d. It is generally obtained by deleting the set of simplexes adjacent to σ in the star of σ.

(Triangulation) A triangulation $\mathcal{T}$ is a combinatorial structure on a continuous domain. A simplicial complex K is a (conforming) triangulation of a topological space $\mathbb{X}$, if the union of its simplices is homeomorphic to $\mathbb{X}$. In other words the intersection of two elements of $\mathcal{T}$ is either reduced to the empty set or to a k-face for $k = 0...d-1$ with d being the spatial dimension.

(Mesh) A mesh is a general coverage of an underlying space by a set of geometric cells. A triangulation is a specific mesh in which cells are simplices. There exist various manners of describing a mesh depending on the structure and the description of its cells and their connectivity. It can be structured, curvilinear, unstructured and have many properties such as blocking or partitioning. Many techniques can be used to generate such meshes [Frey and George, 2007].

3.2 Scalar Field Input Data

The simulation of physical phenomena is driven by the approximation of numerical values at specific locations within the domain. Meshes are a common representation for solving these

types of approximations, such as the Navier-Stokes equations in fluid dynamics and the Maxwell equations in electromagnetism. Thus, most simulation codes and experimental studies generate numerical values in multiple dimensions. These values can be scalars, such as temperature; vectors, such as fluid velocity; or tensors, such as material stress. These values can be expressed on any mesh entity, such as a vertex, edge, face, or cell.

In practice, meaningful analyses performed by scientists are often based on low-dimensional values. Indeed, variations of scalar values are easier to understand than those of tensors. Thus, most of my research projects have focused on scalar values. Sometimes, these are direct outputs of the simulation code, and other times, they are derived variables, such as the vorticity or enstrophy of a vector field. The choice of these derived variables is crucial to understanding and must be made in collaboration with a domain expert. Once this choice is made, however, the analysis can rely on this representation to capture most of the features sought by the experts.

All of our topological data analysis work will benefit from this scalar representation, which uses various computation topology tools on meshes and scalar fields defined on vertices. Interpolation between vertices uses barycentric coordinates.

(Barycentric coordinates) Let p be a point of $\mathbb{R}^n$ and σ a k-simplex. The barycentric coordinates $\alpha_0, \alpha_1, ..., \alpha_k$ are defined such that $\sum_{i=0}^{i=k} \alpha_i \sigma_0^i$ (where σ_0^i is the i^{th} zero dimensional face of σ) and such that $\sum_{i=0}^{i=k} \alpha_i = 1$.

Using this interpolation method between simplices, we can evaluate the scalar field originally defined at the vertices everywhere, introducing the main data representation used in most of our research projects: a piecewise linear scalar field.

(Piecewise Linear Scalar Field) Let a triangulation $\mathcal{T}$ and a function f that maps its 0-simplices v_i to $\mathbb{R}$. A piecewise linear scalar field f on $\mathcal{T}$ is a function that maps any point p of a k-simplex σ of $\mathcal{T}$ to a value $\sum_{i=0}^{k} \alpha_i f(v_i)$ with $\alpha_0, \alpha_1, ..., \alpha_k$ being the barycentric coordinates of p relatively to σ and $v_0, ..., v_k$ the 0-simplices of σ.

3.3 Simplicial Homology to Compare Data

As described earlier, analyzing complex simulation data requires advanced scientific visualization techniques. Comparing data sets is one way to understand hidden features or identify features in large amounts of data. The strategy used in my research is based on creating groups of data and classifying their underlying topological space based on certain properties. Homeomorphisms are one way to compare topological spaces. However, searching for homeomorphisms between spaces can be difficult. One approach is to study the topological invariants of these spaces that are preserved through homeomorphisms. If two spaces have different invariants, then they are not homeomorphic; however, the opposite is not true (spaces with the same invariants are not necessarily homeomorphic).

The main advantage of TDA is that studying topological properties is often equivalent to studying the main features or signature of a dataset. This abstraction allows us to perform

Figure 3.3: A coffee mug and a donut have the same topology with both of them having a hole in their shape (credits A. Gomez)

complex geometric analyses or sift through large amounts of noisy data. *Algebraic topology* provides tools for studying topological spaces and capturing their core features. More precisely, *simplicial homology* introduces topological invariants as holes in domains. Through homeomorphisms, holes cannot be changed, making them interesting and efficient topological invariants for comparing topological spaces. For instance, a disk is a two-dimensional manifold, similar to an empty sphere. However, the sphere's shape is obviously different due to its lack of a defined boundary and its internal void. Another example is the donut, which is a two-dimensional manifold with no boundary and a void resembling a handle. However, the torus is not similar to a sphere in terms of shape because of the hole, which looks more like a handle than a void. A coffee mug is a two-dimensional manifold with no boundary and a handle. Intuitively, the donut and the coffee cup are homeomorphic as illustrated on Figure 3.3. To further understand these features, *persistent homology* brings a complementary theory that captures evolution across different scales according to an input scalar field. Most of my TDA research projects will use this mathematical background to provide scientists with a better understanding of their simulations. We perform these analyses using workstations and supercomputers, taking full advantage of computational topology, which refers to topological analysis performed using computers [Zomorodian, 2010].

Simplicial homology uses collections of simplices called p-chains to detect holes. This notion sums how many times the simplex appears in the chain.

(p-chain) A p-chain of a simplicial complex K of a topological space $\mathbb{X}$ is a sum with modulo 2 coefficients of p-simplices of K.

Thus by examining the p-chains which create a cycle we can identify the holes depending on

their dimension. Holes of dimension 0 correspond to connected components of the simplicial complex; holes of dimension 1 correspond to loops; and holes of dimension 2 correspond to voids. If a p-chain is a cycle then we need to search for extremities, which are formally defined by the notion of boundary. The boundary of a p-chain yields the $(p-1)$-faces of the simplices in the chain that are not shared with another simplex in the chain.

(p-boundary) A p-boundary of a simplicial complex K of a topological space X is the boundary of a $(p+1)$-chain. The idea that $(p-1)$-faces shared by two p-simplices are not included in the boundary thanks to the usage of coefficients modulo 2.

(p-cycle) A p-cycle of a simplicial complex K of a topological space X is a p-chain with empty boundary.

(Homology group) The p^{th} homology group of a simplicial complex K of a topological space X is its p^{th} cycle group (group of all p^{th} cycles noted $Z_p(K)$) modulo its p^{th} boundary group (group of all p^{th} boundaries noted $B_p(K)$): $H_p(K) = Z_p(K)/B_p(K)$.

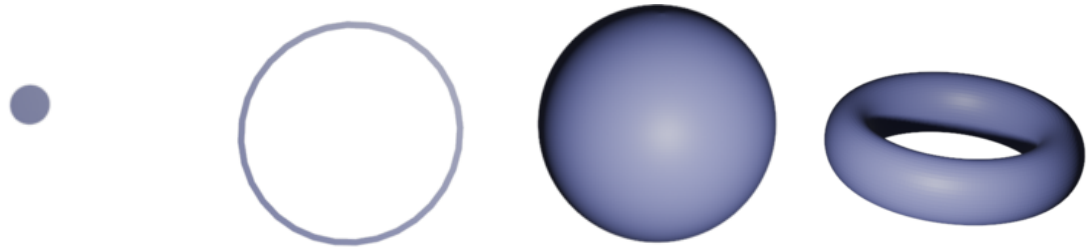
We can note that this quotient space Z/B can intuitively be viewed as the space Z where everything in B is equal to 0. Another example is the set $Z/2Z$ where we consider everything in $2Z$ (i.e. 2, 4, 6, . . .) to be equal to 0.

Within a homology group, p-cycles can be categorized into groups of equivalent p-cycles. In this context, the term equivalent refers to two p-cycles that can be continuously transformed into each other without collapsing to a point. Each group can be represented by a unique representative p-cycle, called a generator. This generator is equivalent to any other p-cycle of the group. Enumerating the generators of a homology group let us introduce a simple and useful topological invariant called the Betti numbers.

(Betti number) The p^{th} Betti number of a triangulation $\mathcal{T}$ of a topological space $\mathbb{X}$ corresponds to its number of independent groups and is computed using the rank of its p^{th} homology group: $\beta_p(\mathcal{T}) = rank(H_p(\mathcal{T}))$. In $\mathbb{R}^3$, the Betti number β_0 corresponds to the number of connected components of $\mathcal{T}$, β_1 to its number of handles and β_2 its number of 3-dimensional voids. Figure 3.4 shows examples of the Betti numbers β_0, β_1 and β_2 on a point, a circle, a sphere and a torus.

One useful tool derived from the Betti number is the Euler characteristic, which can be used to control a triangulation after a transformation. This concept is particularly useful in our mesh simplification work, as it ensures that topological features are preserved.

(Euler characteristic) The Euler characteristic of a triangulation $\mathcal{T}$ of a topological space $\mathbb{X}$ of dimension d, is the alternating sum of its Betti numbers: $E(\mathcal{T}) = \sum_{i=0}^{i=d}(-1)^i\beta_i(\mathcal{T})$. From a computational perspective, it has an interesting property that is generally used in the algorithms. Specifically, the Euler characteristic can be calculated by utilizing the alternating sum of the number of its i-simplices: $E(\mathcal{T}) = \sum_{i=0}^{i=d}\sigma_i$.



(a) Point : $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 0$

(b) Circle : $\beta_0 = 1$, $\beta_1 = 1$, $\beta_2 = 0$

(c) Sphere : $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 1$

(d) Torus : $\beta_0 = 1$, $\beta_1 = 2$, $\beta_2 = 1$

Figure 3.4: Examples of the Betti numbers β_0, β_1 and β_2 on different shapes.

3.4 Persistent Homology

Persistent homology is a key concept in most of my research projects. It captures the homological history of a space that is going to grow undergoing a transformation. This history is modeled for a space $\mathbb{X}$ through a transformation called a filtration represented as a sequence of nested spaces $\emptyset = \mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \ldots \subseteq \mathbb{X}_m = \mathbb{X}$.

To study the data defined on the mesh, we can introduce filtration on a scalar field with a notion of sub-level set. The filtration process involves adjusting the threshold from the lowest values of the scalar field to the highest ones. Additionally, it considers, at each step of the sequence, all points with a function value below the current threshold. For a given scalar field, it is possible to identify the homology classes that persist the most during the filtration process.

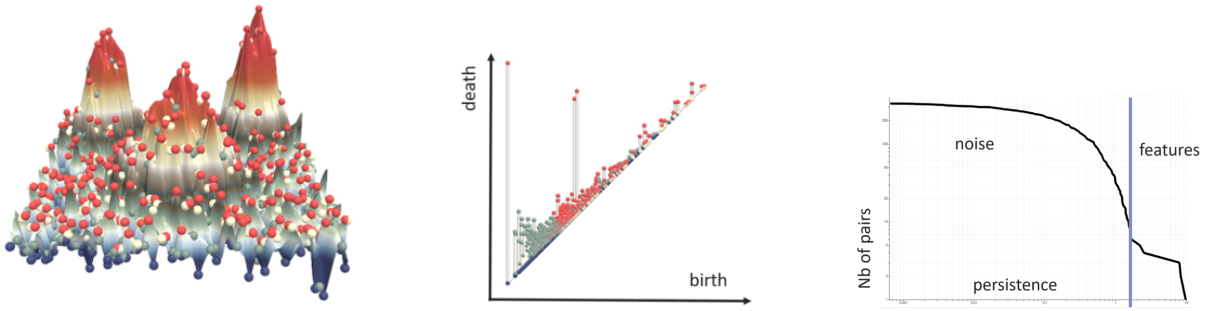
(Sublevel set) A sublevel set is defined as the pre-image of the interval $(-\infty, w)$ by f noted $f^{-1}_{-\infty}(w) = \{p \in \mathcal{M} \mid f(p) < w\}$. It can be used to track topological features in f. Specifically, the topology of these sub-level sets (in 2D, their connected components and cycles) can only change at specific locations. Conversely, a *superlevel set* is defined as the pre-image of the interval $(w, +\infty)$ by f noted $f^{-1}_{+\infty}(w) = \{p \in \mathcal{M} \mid f(p) > w\}$.

(Critical points) As w continuously increases, the topology of $f^{-1}_{-\infty}(w)$ changes at specific vertices of $\mathcal{M}$, called the *critical points* of f [Banchoff, 1970].

(Lower link) The lower link of v, noted $Lk^-(v)$, is given by the set of simplices of $Lk(v)$ which only contain vertices *lower* than v: $Lk^-(v) = \{\sigma \in Lk(v) \mid \forall v' \in \sigma,\ f(v') < f(v)\}$.

(Upper link) The upper link is defined symmetrically: $Lk^+(v) = \{\sigma \in Lk(v) \mid \forall v' \in \sigma,\ f(v') > f(v)\}$.

A vertex v is *regular* if and only if both $Lk^-(v)$ and $Lk^+(v)$ are simply connected. For such vertices, the sub-level sets do not change their topology as they span $St(v)$. Otherwise, v is a *critical point* of f [Banchoff, 1970]. These can be classified with regard to their *index* $\mathcal{I}(v)$. It is equal to 0 for local minima ($Lk^-(v) = \emptyset$), to 2 for local maxima ($Lk^+(v) = \emptyset$) and otherwise to



(a) (b) (c)

Figure 3.5: Noisy Gaussian dataset. (a) Terrain visualization with critical points as spheres (white: minima, blue: maxima, other: saddles). (b) Persistence diagram with prominent persistence pairs for the two Gaussians, while small oscillations due to noise induce pairs near the diagonal. (c) Persistence curve denoting the population of persistent pairs as a function of their persistence.

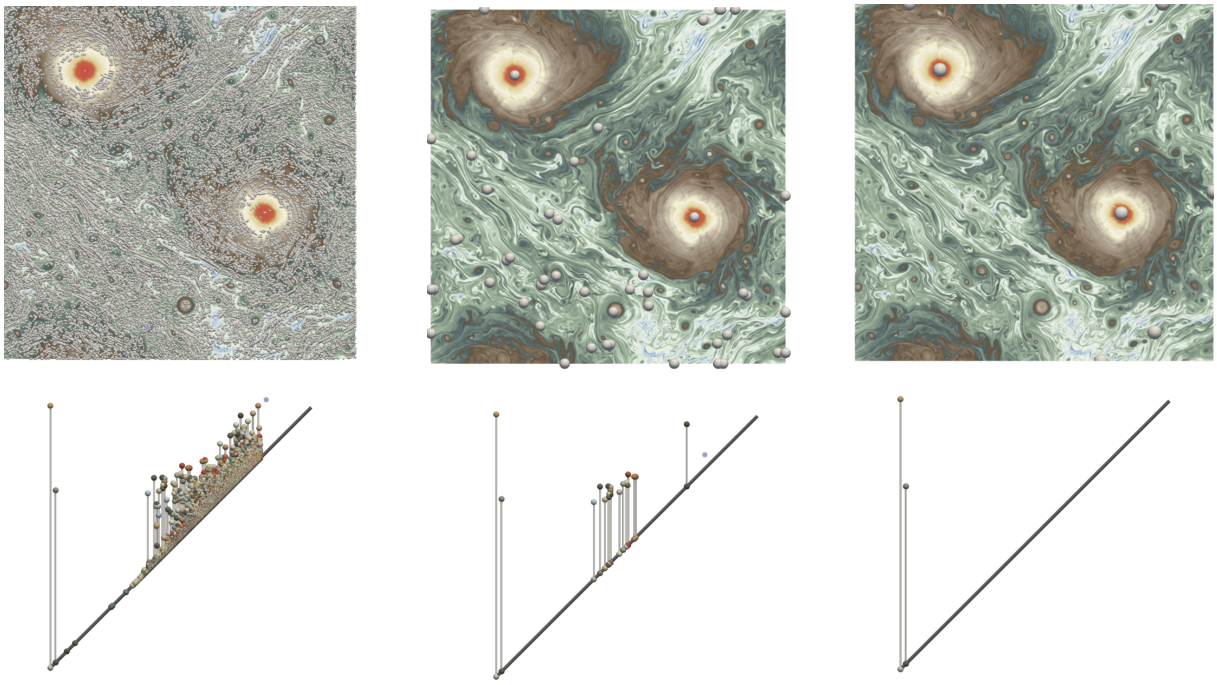
1 for saddles. In practice, f_i is enforced to contain only isolated, non-degenerate critical points [Edelsbrunner and Mucke, 1990; Edelsbrunner et al., 2003b].

In a multitude of real-world datasets, scalar fields are found to contain minor local variations. However, since the critical point characterization is based on a classification restricted to the link of each vertex, that makes it only local. As a result, the slightest oscillation in the data results in the appearance of spurious critical points as seen on Figure 3.5a. In situations involving noisy data, such as turbulent flows or instabilities, it is essential to implement a systematic approach to identify and filter these critical points. This motivates the introduction of an importance measure of critical points with the objective of disambiguating vortices from noise.

Several important measures for critical points have been studied [Carr et al., 2004] including local geometric measures based on computation of volume evaluated with the number of vertices or hypervolume with the sum of sample values. One of the most frequently used and effective concepts is the *topological persistence* [Edelsbrunner et al., 2002]. Persistence is a measure of the importance of a critical point, in relation to the lifetime of the topological feature it creates or destroys in $f_{-\infty}^{-1}(w)$, as the isovalue w is continuously increased.

Specifically, as w increases, new connected components of $f_{-\infty}^{-1}(w)$ are created at the minima of f_i. According to the Elder rule [Edelsbrunner and Harer, 2009] indicates that if two connected components, created at the minima m_0 and m_1 with $f(m_0) < f(m_1)$, meet at a given saddle s, the *youngest* of the two components (the one created at m_1) *dies* in favor of the *oldest* one (created at m_0).

In this case, a *persistence pair* (m_1, s) is formed and its *topological persistence* p is given by $p(m_1, s) = f(s) - f(m_1)$. All the local minima can be unambiguously paired following this strategy, while the global minimum is usually paired with the global maximum according to convention. The symmetric reasoning can be applied to characterize, with saddle/maximum pairs, the lifetime of the independent cycles of $f_{-\infty}^{-1}(w)$.



(a) Original data (b) Noise removal (c) Main features

Figure 3.6: Persistence simplification by removing critical pairs close to the diagonal on the persistence diagram.

3.5 Various Topological Representations

(Persistence diagrams) Persistence pairs are usually visualized with the *persistence diagram* $\mathcal{D}(f)$, which embeds each pair (c, c'), with $f(c) < f(c')$, as a point in the 2D plane, at the location $\big(f_i(c), f_i(c')\big)$. The value $f(c)$ generally referred to as the *birth* of the feature, while $f(c')$ is designated as its *death*. The pair persistence can be visualized as the height of the point to the diagonal. Features with a high persistence stand out, away from the diagonal, while noisy features are typically located in its vicinity. Several algorithms exist to compute the diagram such as [Bauer et al., 2014; Edelsbrunner and Harer, 2009] via matrix reduction.

Figure 3.5b shows the two main persistence pairs tracking the two main Gaussian maxima of the example dataset. All the critical points generated by local extrema due to the noise are represented as small bars near the diagonal. The conciseness, stability and expressiveness of the persistence diagram have made it a popular tool for data simplification [Edelsbrunner et al., 2002]. It provides visual hints about the number, ranges and salience of the features of interest. Based on the height of the pairs of critical points on a diagram, it is easy to adjust the amplitude of what could be classified as noise. The simplification process deletes all pairs on the diagram close to the diagonal and simplifies the other structures. Figure 3.6 shows a persistence simplification based on the diagrams. The bottom views show the persistence diagrams used for the simplification with the original turbulence Figure 3.6a, a diagram after noise removal Figure 3.6b (*e.i* all the pairs close to the diagonal) and a simplified diagram with only the two main pairs corresponding to the two main vortices of this turbulence Figure 3.6c.

(Persistence curves) A persistence curve, denoted as $C(f_i)$, plots the population of persistent pairs as a function of their persistence. Specifically, it encodes the number of pairs (Y axis) whose persistence exceeds a threshold value ϵ (X axis).

Figure 3.5c shows the persistence curve of the two Gaussian dataset. For $X = 0$, Y is equal to the total number of persistence pairs, while for the largest values of X, Y indicates the number of prominent, high-persistence features. In practice, large plateaus in this curve will indicate *stable* persistence ranges, for which no (or few) topological features are present in the data. These correspond to *separations* between populations of topological features of distinct persistence scales, typically the noise (low X values) and the persistent features (high X values).

(Morse function) A Morse function is a smooth function defined on a manifold such that all critical point p (section 3.4) is non-degenerate and no two critical points have the same function value. It means that its Hessian matrix of p is non-singular ($det\ H(p) \neq 0$).

(Contours) Given a manifold $\mathbb{M}$ and a Morse function $f : \mathbb{M} \to \mathbb{R}$, let $f^{-1}(w)$ be the level set of the isovalue w. A contour is a connected component of this level set f^{-1} at value w. Contours in two-manifolds are loops called isolines and voids called isosurfaces for three-manifolds.

(Reeb graphs) A Reeb graph is a graph whose nodes correspond to the critical points of a Morse function and whose edges encode the evolution of the connected components of the level sets between these critical points [Reeb, 1946]. It captures how the topology of the contours changes as the function's value varies. It is constructed by contracting each contour to a point [Biasotti et al., 2008]. For manifolds with complex topology such as a torus the Reeb graph can have loops (cycles). Several efficient algorithms have been documented [Doraiswamy and Natarajan, 2013; Parsa, 2012; Pascucci et al., 2007; Tierny et al., 2009], including parallel algorithms [Gueunet et al., 2019a].

(Contour trees) The contour tree is a Reeb graph without a cycle that tracks the evolution of the contours of the level set as they split, join, appear, and disappear [Tarasov and Vyali, 1998]. This acyclic nature arises when the manifold is simply connected, or more generally, when the function's behavior doesn't create "loops" in the connectivity of its level sets.

The construction of the contour tree can use intermediate structures trees [Carr et al., 2003] that are simpler forms of Reeb graphs, each focusing on a specific aspect of level set connectivity.

(Join tree) The join tree also known as the merge tree captures how connected components of sublevel sets $f^{-1}_{-\infty}(w)$ merge together as the function value r increases. A threshold plane is swept from $-\infty$ upwards across the function's range. During this process the connected components of the region where $f(x) \leq w$ are described. The root of the join tree is the global maxima of the function where all components merge. A new connected component in the sweep process is created at a minima. Two or more pre-existing connected components of the sublevel set merge into a single component at a saddle point.

(Split tree) The split tree captures how connected components of superlevel sets $f^{-1}_{+\infty}(w)$ split apart as the function value w decreases. It can be thought of as the join tree of the function $-f$. In this tree, a new connected component in the sweep process is created at a maxima and the global minimum of the function corresponds to the root.

The join and split trees allow to efficiently construct the full contour tree. The contour tree itself is then formed by combining (or "merging") the join tree and the split tree. The join tree captures all the "merging" events of level sets whereas the split tree captures all the "splitting" events. The contour tree relies on the values of the function and the connectivity of its level sets. It is always a 1-dimensional graph regardless of the dimension of the underlying manifold and does not show geometric information about the features. Efficient algorithms have been documented [Carr et al., 2000; Tarasov and Vyali, 1998] and parallel algorithms have also been described [Acharya and Natarajan, 2015; Carr et al., 2016; Gueunet et al., 2019b; Maadasamy et al., 2012].

In order to add a geometrical description of the features, the Morse-Smale complex provides a full partitioning of the manifold based on gradient flow, giving a rich geometric and topological understanding of the data. This representation is based on gradient flow [Gyulassy et al., 2012].

(Integral line) An integral line is a maximal path whose tangent vectors agree with the gradient. They have an origin and a destination to critical points and are always disjoint or the same.

(Ascending manifold) Let f be a Morse function defined on a manifold. The ascending manifold of a critical point p of f is the set of points belonging to integral lines whose origin is p.

(Descending manifold) Let f be a Morse function defined on a manifold. The descending manifold of a critical point p of f is the set of points belonging to integral lines whose destination is p.

(Morse-Smale Complex) The Morse-Smale complex is a graph representation partitioning the entire manifold into regions based on where gradient flow lines originate and terminate. The vertices (0-simplex) of the complex are the critical points of the function f. The edges (1-simplex) of the complex are the integral lines of the gradient vector field of f. These are the paths going always toward the direction of steepest ascent or steepest descent. The Morse-Smale complex is constructed by the intersection of segmented regions of the ascending and descending manifold. Thus, these cells of the complex are regions with monotone gradient flow behaviors that are very interesting for the understanding of a scalar field. The Morse-Smale complex has been studied for many different inputs [Bremer et al., 2003; De Floriani et al., 2015; Edelsbrunner et al., 2003a]. Based on Discrete Morse Theory [Forman, 1998], many robust and efficient algorithms have been introduced for its computation [Gyulassy et al., 2018; Robins et al., 2011; Shivashankar and Natarajan, 2012].

This chapter has presented the core concepts of computational topology used in my research. It's important to note that the field of computational topology is much broader and richer, as detailed in [Edelsbrunner and Harer, 2009] and [Zomorodian, 2010]. For some projects, I have been programming the data structures and algorithms related to these core concepts, as shown in chapter 4. For some others, we've been utilizing the Topology ToolKit (TTK) [Tierny et al., 2017a], a software platform designed for topological data analysis (TDA). TTK provides user-friendly topological abstractions and algorithms that are integrated as plug-ins in ParaView and available through various bindings (Python, C++). In many of my research projects, we've used

TTK to create domain-specific TDA workflows, as described in chapter 6.

CHAPTER 4

Computational Topology for Mesh Simplification

Computational topology has been the core concept of my PhD [Vivodtzev, 2005] in the domain of scientific visualization of simulation data based on mesh simplification. Topological criteria, independent of the geometric complexity of the models have been studied in order to guarantee the relevance of low-resolution visualizations, particularly at the scale of embedded features in the volume.

The innovative idea behind this work stems from the extension of the simplicial complex representing the mesh. This extension is done by creating a cone of simplices based on the simplices to be preserved in order to represent all the multidimensional features in the same extended complex. This allows us to identify and guarantee edge contractions that preserve the topology of the mesh and that of all its substructures [Vivodtzev et al., 2005a]. The success in the use of computational topological criteria has motivated the investigation of these approaches on numerical simulation data carried by meshes in the domain of electromagnetism [Vivodtzev et al., 2010] and neurosciences [Vivodtzev et al., 2005b].

Numerical simulations and imaging cameras generate large amounts of data that are far greater than the available resources on a workstation with advanced visualization techniques such as parallel or in-situ strategies. In electromagnetism simulations, the finite element meshes are composed of several million of volumetric cells that have embedded sub-structures of varying dimensions (surfaces and linear features) such as thin material-boundary layers. Unstructured tetrahedral meshes with substructures such as wires and patches are represented as triangles and edges. In neurosciences, cortex surfaces, described as isosurfaces taken from MRI images, present specific topological connectivity between the main features.

4.1 Topological Criteria for Preserving Embedded Features

This section presents the topological abstraction and specific simplices that are added to the original simplicial complex in order to build a multiresolution framework based on mesh simplification with topological consideration.

4.1.1 Towards a Multiresolution Representation

The contribution of my PhD work [Vivodtzev, 2005] is a pre-processing step for this massive data to reduce the required amount of data while preserving the topology of embedded features to ease interaction on a single workstation. However, the tools in scientific visualization in the 2000s did not allow for, or only partially achieved, the interactive exploration goals for this type of data (i.e., large number of cells, substructures, thin layers ...). The same topological criteria are used to simplify surfaces to highlight main topological features and to ease the comparison of complex surfaces.

A pre-processing step was introduced that involves a hierarchical organization of data in order to construct a multiresolution representation. This step is based on a mesh simplification algorithm that uses iterative edge collapses while preserving both mesh topology and the topology of all embedded substructures. The robust topological criteria introduced in this work are derived from theoretical notions in computational topology.

The visualization step then uses the multiresolution representation produced in the first step to speed up rendering time of the data. In a progressive, invertible, and local manner, the method adapts dynamically to the required resolution, specified by the user, and the hardware resources available.

4.1.2 Topology-preserving Simplification

Mesh simplification algorithms transform a general simplicial complex into another. It has been extensively studied in computer graphics to reduce the resolution of 2D and 3D objects while preserving geometric or topological features [Dey et al., 1999; Garland and Heckbert, 1998; Garland and Zhou, 2005]. Edge collapse [Hoppe et al., 1993] is the most widely used operator for removing sub-simplices from input because it allows for efficient and accurate control of the deformation introduced in the initial mesh. Some of these algorithms in 2D preserve the topology of the simplified mesh by investigating the organization of triangles around an edge, which leads to a number of specific cases. In some situations, the edge collapse is rejected as it would introduce topological errors. A similar treatment can be performed on feature lines based on local tests between the edges around a collapse and the triangulation.

Many studies [Dey et al., 1998] have proven that the complex obtained after an edge collapse is homeomorphic to the first one if the neighborhood of the edge collapse satisfies the link condition [Vivodtzev et al., 2005a]. This result, based on concepts of computational topology,

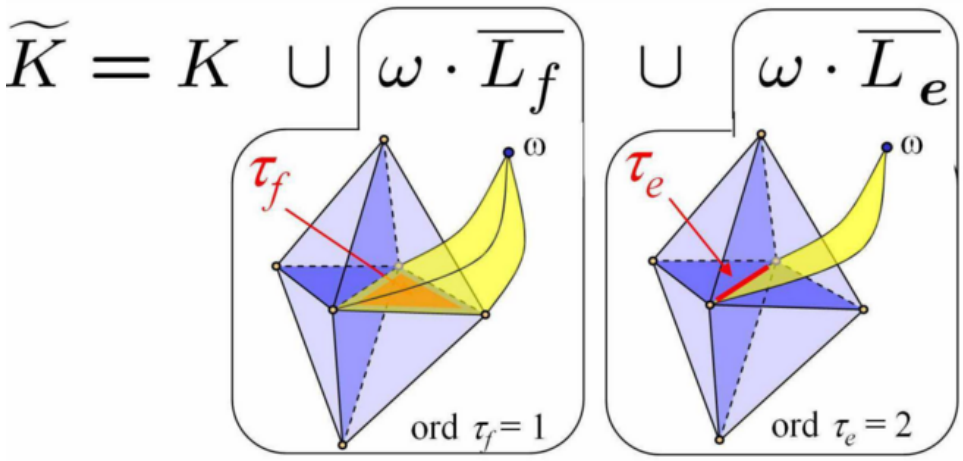


Figure 4.1: Construction of the extended complex. The red simplices correspond to the substructures L_f and L_e. The yellow simplices are the elements added to the original blue complex in order to extend K to $\widetilde{K}$.

has found increasingly widespread use, even in volume simplification, as it is extensible to 3-complexes. Non-manifold surfaces can also be treated with this method using the link condition.

In our work on mesh simplification [Vivodtzev et al., 2005a], we first used the link condition introduced in [Dey et al., 1998] on 2-complexes. We then extended this approach to tetrahedral meshes with 2D and 1D substructures. The link conditions are expressed using basic topological concepts. If the mesh is a manifold 3-complex K, the link conditions reduce to:

$$L_k u \cap L_k v = L_k e \tag{4.1}$$

If the 3-complex is non-manifold, the link condition has to be tested on three successive smaller complexes of dimensions 3, 2 and 1.

4.1.3 Link Conditions in the Extended Complex

A mesh is represented by a simplicial complex K. The 2D (resp. 1D) substructures are defined as a set L_f (resp. L_e) of faces (resp. edges) in K. We define an extended complex $\widetilde{K}$ by adding to K the cones from a dummy vertex, ω, to each simplex in the closure of the substructures, as illustrated in Figure 4.1. The key idea is to consistently encode the topology of the mesh and its substructures in a single extended complex. Using results established in computational topology on this extended complex, it is possible to determine whether an edge can be contracted without modifying the topology of the mesh or that of all its substructures. This check is performed using the link condition on the extended complex $\widetilde{K}$ [Vivodtzev et al., 2010].

To preserve the topology of the mesh and its substructures, we apply the link conditions to the extended complex. Even if the original mesh is manifold, the extended complex is non-manifold. Thus, the link condition of equation 4.1 must be evaluated respectively on the

following complexes:

$$\begin{aligned} \widetilde{K}^{\omega} &= \widetilde{K} \cup (\omega \cdot \mathrm{Bd}_1 \widetilde{K}) \\ \widetilde{G}^{\omega} &= \mathrm{Bd}_1 \widetilde{K} \cup (\omega \cdot \mathrm{Bd}_2 \widetilde{K}) \\ \widetilde{H}^{\omega} &= \mathrm{Bd}_2 \widetilde{K} \cup (\omega \cdot \mathrm{Bd}_3 \widetilde{K}) \end{aligned} \tag{4.2}$$

where $\mathrm{Bd}_i \widetilde{K}$ is the set of all simplices of order i or less in $\widetilde{K}$. The link of a simplex in $\widetilde{K}^{\omega}$, $\widetilde{G}^{\omega}$ and $\widetilde{H}^{\omega}$ are denoted as Lk_0^{ω}, Lk_1^{ω}, and Lk_2^{ω}, respectively. Simplices of dimension k in a 3-complex have an order not greater than $3 - k$. Therefore $\widetilde{K}^{\omega}$ is a 3-complex, while $\widetilde{G}^{\omega}$ contains only edges and $\widetilde{H}^{\omega}$ only vertices. Note that our algorithm can deal with non-manifold tetrahedral meshes. However, for simulation data in electromagnetism and neurosciences explored next we restricted to the case where the tetrahedral mesh is a manifold with boundary. In this case, $\mathrm{Bd}_1 \widetilde{K}$ contains the boundary faces of K and the faces of L_f.

It also contains the edges and vertices of these faces. $\mathrm{Bd}_2 \widetilde{K}$ contains the edges of L_e and the edges adjacent to exactly one, or three or more faces of L_f (i.e., the boundary and the non-manifold edges of L_f). It also contains the vertices of these edges. Finally $\mathrm{Bd}_3 \widetilde{K}$ contains the vertices adjacent to one or three or more edges of $\mathrm{Bd}_2 \widetilde{K}$ (i.e., the boundary and non-manifold vertices of $\mathrm{Bd}_2 \widetilde{K}$). This includes vertices at the intersections created when a polyline in L_e crosses a surface in L_f. From our experience, the simplices of high order correspond to regions of interest that the user visualizes in detail. Important features in the mesh (detected by cells marked as simplices of high order) are usually correlated with high variations in the data. This point is clearly illustrated in the electromagnetic simulations where the field varies greatly around the linear structure or the points of contact between this element and a mass plane.

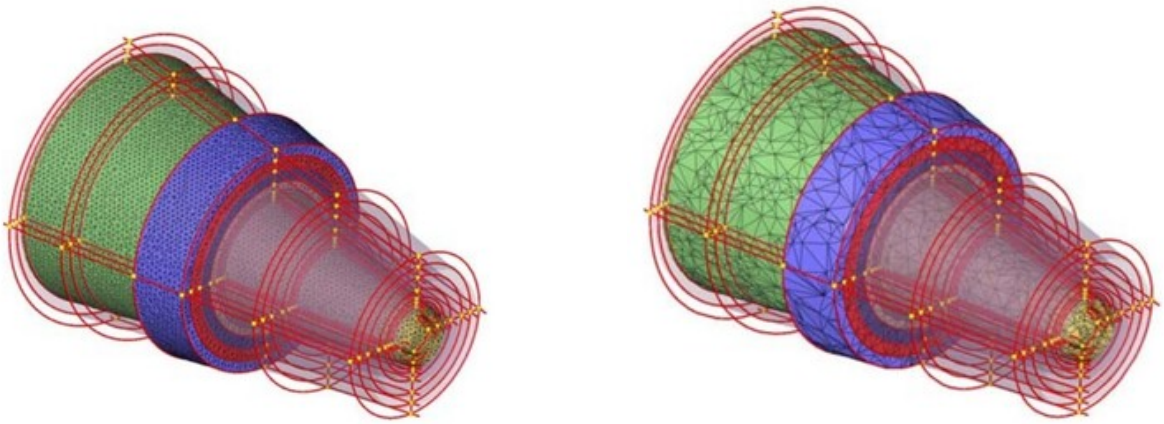
These effective techniques demonstrate the importance of topological approaches in many fields [Bennett et al., 2015] as illustrated in electromagnetism and neurosciences in the next section.

4.2 Applications in Electromagnetism

4.2.1 Volumetric Simplification

Volumetric simplification of electromagnetism simulation results needs to handle domain-specific features. The geometry is composed of approximately twenty thin layers of materials encapsulated in each other. The geometry is meshed with about 2 million tetrahedra illustrated in Figure 4.2a. To study the circulation of the electromagnetic field at the boundary of each material, all interfaces are explicitly described as embedded surfaces. Additional linear structures with self-intersections are inserted in the model.

After a strong substructure-topology-preserving simplification as shown on Figure 4.2b, the simplified mesh is left with only 10% of the original vertices. This example illustrates the strength of the algorithm on data composed of a large amount of substructures of different dimensions with multiple self-intersections (e.g. which increases the topological complexity). Achieving



(a) 100% (b) 10%

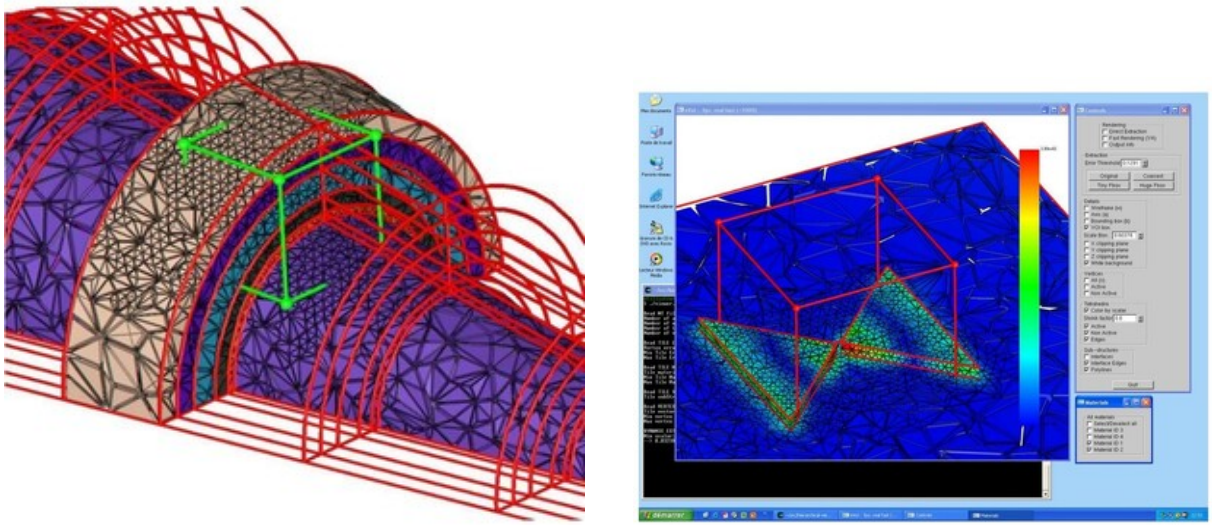
Figure 4.2: Volume simplification of a cone. Some materials are represented by their colored interfaces, while others use semi-transparent surfaces. After 90% simplification, the substructures of the assemblies are preserved.

aggressive simplification rates becomes challenging when dealing with large volumetric meshes organized into thin material layers. The link condition driving the candidate edges to collapse guarantees the preservation of all the domain-specific features.

4.2.2 Multiresolution Visualization

To provide a visualization system that takes advantage of the substructure-topology-preserving criteria, the algorithm has been integrated into an existing multiresolution framework. The MT library Floriani et al. [2000], freely available and based on the formal approach of Floriani et al. [1997], provides the tools to build and exploit a multiresolution representation from a sequence of valid transformations on a mesh. MT stores a partial order among the modifications in a directed acyclic graph (DAG) which can then be queried with static or dynamic criteria to produce various resolutions.

In [Vivodtzev, 2009], we integrated topological criteria with the MT library. Global or local extractors allow the user to extract meshes at variable resolutions centered around a volume of interest (VOI). Figure 4.3a and Figure 4.3b show a multiresolution mesh at high resolution only inside a VOI that is interactively manipulated by a user. The mesh at low resolution reduces the use of the graphics processor allowing the application to maintain an interactive frame rate even with an original mesh of several millions of cells. As the substructure preserving topology criteria have been used to build this representation, the topology of the substructures is guaranteed to be preserved on the multiresolution visualizations.



(a) (b)

Figure 4.3: (a) Variable resolution visualization of a volume mesh with multiple linear features. The topology of the substructures is guaranteed to be preserved. (b) Snapshot of the multiresolution visualization tool to explore simulation data with embedded structures on a desktop PC.

4.3 Applications in Neurosciences

Data analysis in neurosciences is also a field for the application of mesh simplification where topology-preserving criteria to preserve structures of interest can be considered. This work, carried out between 2002 and 2006, is part of the international project *The Human Brain Project*, which aims to help neurologists diagnose brain diseases. This work has been done in collaboration between the IDAV Institute at the University of California at Davis, the Neurology Center in the Department of Psychology at UC Davis and the GRAVIR laboratory at Inria in France as presented in [Fuller et al., 2003].

One of the challenges is to unify information collected from different brains. In particular, it is important to be able to combine the many studies carried out on magnetic resonance images (MRI) within a single atlas. This requires segmentation and association of features from one brain model to another (from the atlas to the patient or vice versa).

The first results obtained in [Vivodtzev et al., 2003] deal with the segmentation of a cortex surface using curvature estimation and a multi-resolution approach. In this project, features are not identified with computational topology but rather with geometric properties. The topology of the features will be considered later, as explained next. The main features of the human cortex are the networks of folds (sulci and gyri) that make up its surface. The projection of information from one brain to another can use these features to calibrate the projection and compare two corresponding regions. To do this, it is necessary to be able to segment the irregular surface of a cortex correctly.

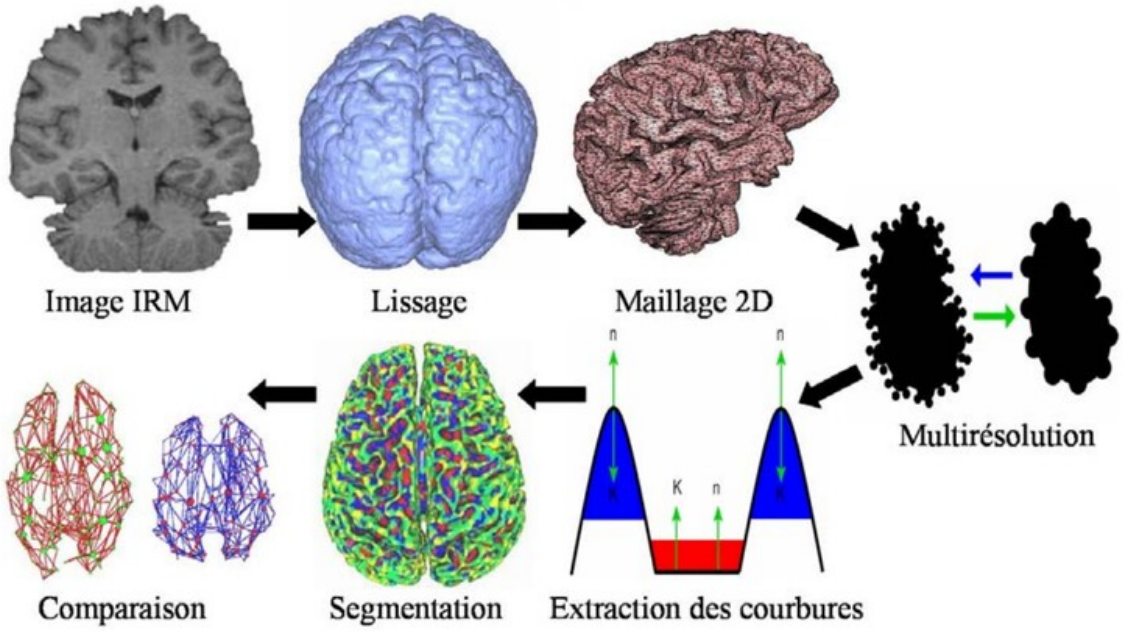


Figure 4.4: Processing pipeline from a volume data set to a segmented and topologically analyzed isosurface.

One way to describe the characteristics of a surface is to segment it into regions of uniform curvature behavior and construct an abstract representation given by a (topology) graph. We introduce a surface segmentation method based on discrete mean and Gaussian curvature estimates. The surfaces are obtained from three-dimensional imaging data sets by isosurface extraction after data smoothing and post-processing the isosurfaces by a surface-growing algorithm. We generate a hierarchical multiresolution representation of the isosurface. Segmentation and graph generation algorithms can be performed at various levels of detail.

At a coarse level of detail, the algorithm detects the main features of the surface. This low-resolution description is used to determine constraints for the segmentation and graph generation at the higher resolutions. We have applied our methods to MRI data sets of human brains. The hierarchical segmentation framework can be used for brain mapping purposes.

4.3.1 Pipeline from MRI to Simplified Triangular Surfaces

From the MRI images, a sequence of operations transforms a model of the cortex so that it can be annotated using an atlas model. Figure 4.4 illustrates each stage of this sequence. The initial dataset corresponds to a volume created from a few hundred MRI images. A Gaussian filter is used to smooth out the image noise. An isosurface is generated with a choice of isovalue validated by the experts to extract the cortex.

The noise remaining in the MRI images reveals numerous small regions isolated from the main surface. Because these regions are useless for cortex segmentation, they are removed in order to obtain a single closed surface per cortex model. A surface growing algorithm is used to

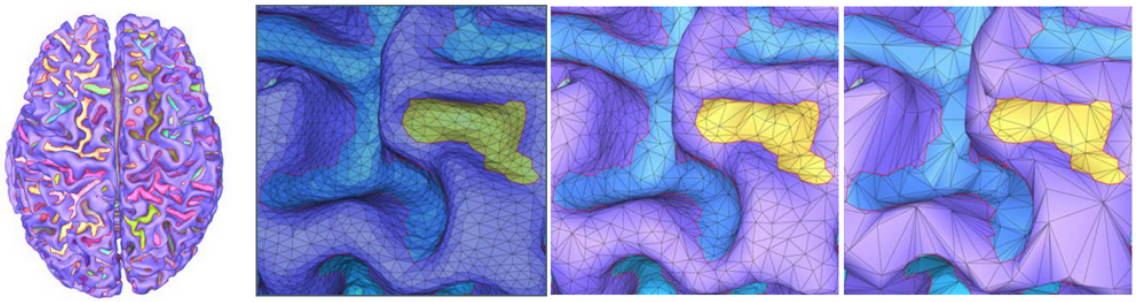


Figure 4.5: Simplification of a cortex surface with preservation of the topology of the features. Sulci boundaries form linear substructures and sulci triangles form surface substructures.

detect the largest closed surface, corresponding to the cortex. At this stage, a triangular manifold mesh is created to efficiently manipulate these surfaces.

However, the surface of a human cortex is highly non-convex with multiple sulci and gyri. The large gyri (bumps on the surface) are deformed by a multitude of small sulci (folds). In order to isolate the main features, the multi-resolution approach presented in section 4.1 has been applied on the cortex surface using mesh simplification with iterative edge collapse. The sulci can be represented as surface substructures of the triangular mesh of a cortex model, however, it is important to note that for this first simplification step, substructure topology preservation techniques are not used because we want to simplify the surface and not to preserve its details.

In contrast to the electromagnetism application example presented in section 4.2, the cortex geometries are deliberately deformed. To achieve this, full edge collapse is used, with the vertex resulting from contraction positioned in the middle of the edge. In order to maintain speed of execution, it was decided not to optimize the position of the vertex along the edge. The order of priority of the contractions is based on the length of the edges. The shortest edges on the mesh represent fine details of the surface, such as small sulci. Since the aim is to filter out these features, this simple heuristic allows the mesh cells representing these sulci to be contracted and deleted first.

Starting from the original model, which represents the cortex at high resolution, the triangular mesh is then iteratively simplified. At certain levels of resolution, the surface is segmented, based on an estimate of the discrete curvatures of the surface as described in subsection 4.3.2 to extract the main sulci and gyri. This approach provides a hierarchy of segmentations with low resolution defining constraints for more complex features detected at high resolution.

After an initial segmentation, it may be useful to further simplify the surface while preserving the topology of the sulci. This second simplification is used to speed up visualization and optimize the shape of the surface to ease segmentation based on an estimate of the discrete curvatures in order to work with a smooth surface. However, it is important to preserve certain characteristics when smoothing the surface, in particular by preventing the fusion of two sulci that are close to each other and each describe a different region of the cortex. The topology preservation tests presented in section 4.1 are then used.

The brain shown on Figure 4.5 is the result of a simplification that preserves the topology of

the surface and its characteristics. This simplification follows a segmentation step in which the main sulci were marked. The boundary edges between a sulcus and a gyrus are the polylines to be preserved and the sulci triangles are the faces of the surface substructures. The results show that the sulci are well-preserved even after significant simplification as shown on three different resolutions (100%, 50%, 25%) on Figure 4.5. The use of topological tests certifies that no sulcus has been fused with one of its neighbors.

4.3.2 Surface Segmentation with Discrete Curvature

A surface's behavior can be described by dividing the surface into disjoint regions of elliptic paraboloid behavior and hyperbolic paraboloid behavior. Let p be an arbitrary point on the surface. The surface's behavior at p can be determined by considering the curves on the surface that pass through p in the direction of the principal curvatures. The regions of elliptic behavior can further be classified into convex and concave regions by considering the direction of the surface normal at p. When considering the cortex of a human brain, the gyri contain convex elliptic regions, and the sulci contain concave elliptic regions, whereas the blending area between gyri and sulci is a hyperbolic region.

When dealing with triangle meshes, the principal curvature directions cannot be uniquely determined. Therefore, in [Vivodtzev et al., 2003] we use estimates for so-called discrete mean and Gaussian curvature. Many approximation schemes for curvature estimates on meshes have been developed such as [Hamann, 1993; Max, 1999] and we use some operators that were derived by [Meyer et al., 2002].

We use mean curvature estimates to distinguish between concave and convex regions. We simplify the definition of [Meyer et al., 2002] and use an operator $\mathbf{K}_{dir}(\mathbf{x}_i)$ only interested in the direction of the curvature. The vector $\mathbf{K}_{dir}(\mathbf{x}_i)$ for a vertex $\mathbf{x}_i$ is computed by a weighted sum of difference vectors emanating from $\mathbf{x}_i$ and ending at the vertices of $\mathbf{x}_i$'s 1-ring. The weight of the vector associated with edge e_{ij} between $\mathbf{x}_i$ and its neighbor $\mathbf{x}_j$ depends on the cotangents taken from the opposite angles of its adjacent faces. Its definition is:

$$\mathbf{K}_{dir}(\mathbf{x}_i) = \sum_{j=1}^{N_i} (\cot \alpha_j + \cot \beta_j)(\mathbf{x}_j - \mathbf{x}_i) \tag{4.3}$$

where N_i is the number of neighbors constituting the 1-ring of $\mathbf{x}_i$, and α_j, β_j are the opposite angles of e_{ij} with respect to its adjacent faces. We use $\mathbf{K}_{dir}(\mathbf{x}_i)$ to define the Boolean operator $mean(\mathbf{x}_i)$, which distinguishes between convex and concave regions. It is defined as

$$mean(\mathbf{x}_i) = \begin{cases} \text{convex} & \text{if } \mathbf{K}_{dir}(\mathbf{x}_i) \cdot \mathbf{n}_i \leq 0 \\ \text{concave} & \text{if } \mathbf{K}_{dir}(\mathbf{x}_i) \cdot \mathbf{n}_i > 0 \end{cases} \tag{4.4}$$

where $\mathbf{n}_i$ is a discrete approximation of the normal vector at $\mathbf{x}_i$. In concave areas, the operator $\mathbf{K}_{dir}(\mathbf{x}_i)$ and the normal vector $\mathbf{n}_i$ point in roughly opposite directions, whereas in convex areas they point in roughly the same direction. This operator enables us to use a first classification of the surface.

Then, to distinguish between elliptic and hyperbolic regions (i.e., separate local extrema from blending regions), Gaussian curvature is considered. An operator $\kappa_G(\mathbf{x}_i)$ is used, whose length is a discrete approximation of the Gaussian curvature [Meyer et al., 2002]. This operator $\kappa_G(\mathbf{x}_i)$ is then used to create a Boolean operator $Gauss(\mathbf{x}_i)$, which is true if the vertex $\mathbf{x}_i$ is a local extremum (minimum for a sulcus or maximum for a gyrus). The operator compares 2π with the sum of inner angles θ_j of all the adjacent faces of a vertex $\mathbf{x}_i$. In the planar case, the angles sum up to 2π.

When $\mathbf{x}_i$ is an extremum, a plane through $\mathbf{x}_i$ exists where all neighbor vertices of $\mathbf{x}_i$ lie on one side of that plane. In this situation, the angles sum to a value smaller than 2π. When $\mathbf{x}_i$ is not an extremum and the best fitting plane is computed in the least-squares sense through $\mathbf{x}_i$, the neighbor vertices are both above and below that plane. In this situation, the angles sum up to a value larger than 2π.

Hence, we are only interested in the sign of the operator $\kappa_G(\mathbf{x}_i)$ and define it as:

$$\kappa_{Gsign}(\mathbf{x}_i) = 2\pi - \sum_{j=1}^{N_i} \theta_j, \tag{4.5}$$

where θ_j is the angle between the difference vectors $\mathbf{x}_j - \mathbf{x}_i$ and $\mathbf{x}_{j+1} - \mathbf{x}_i$ from vertex $\mathbf{x}_i$ to its neighbors $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$. In summary, our Boolean operator used to define surface type is:

$$Gauss(\mathbf{x}_i) = \begin{cases} \text{elliptic} & \text{if } \kappa_{Gsign}(\mathbf{x}_i) > 0 \\ \text{hyperbolic} & \text{if } \kappa_{Gsign}(\mathbf{x}_i) \leq 0 \end{cases} \tag{4.6}$$

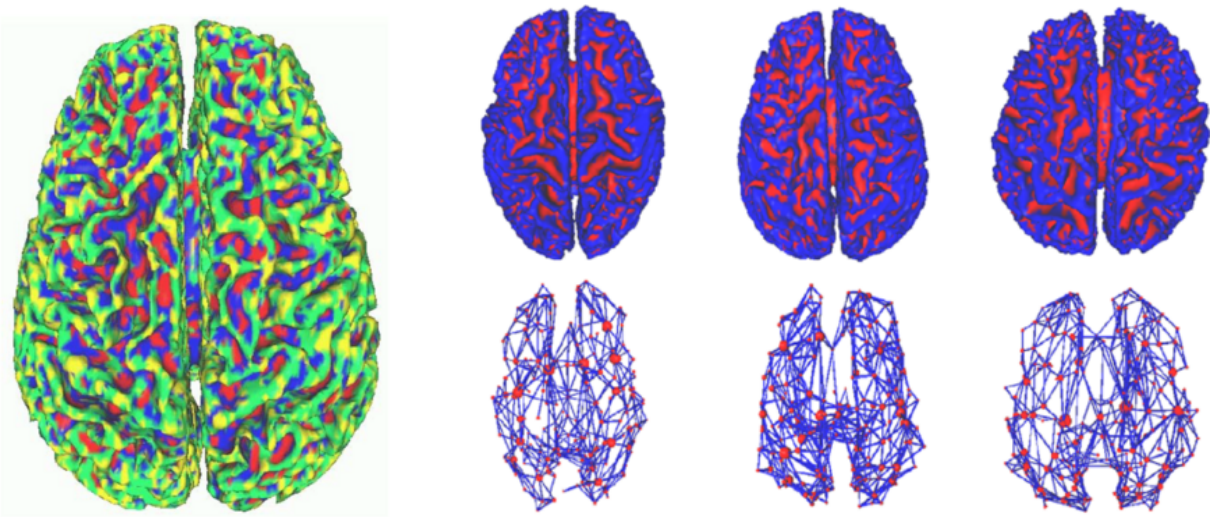
4.3.3 Brain Mapping with a Topology Description of Segmented Features

By combining the *mean* and *Gauss* operators, a surface can be segmented. Utilizing both Boolean operators leads to four distinct scenarios, as shown in Table 4.1. Detecting extrema with *Gauss* in an identified sulcus or gyrus area allows for precise localization of sulcus minima and gyrus maxima. Figure 4.6a visually represents this concept by assigning colors to surface types according to Table 4.1.

Table 4.1: Possible combinations of surface types according to curvature.

$mean\ (\mathbf{x}_i)$	convex	convex	concave	concave
$Gauss(\mathbf{x}_i)$	hyperb.	elliptic	hyperb.	elliptic
color	blue	red	green	yellow

This segmentation is then used to construct a topological graph [Vivodtzev et al., 2005b] describing the connectivity of the sulci on the surface of the cortex as shown in Figure 4.6b. A node in the graph corresponds to a sulcus, and the arcs between the nodes are the neighborhood relationships on the surface. The notion of neighborhood is chosen as a distance along the surface between the boundaries of the sulci. This graph is used to compare the atlas and patient models. However, because of the initial noisy MRI data, the correspondence between two nodes remains delicate even using topological neighborhood relations.

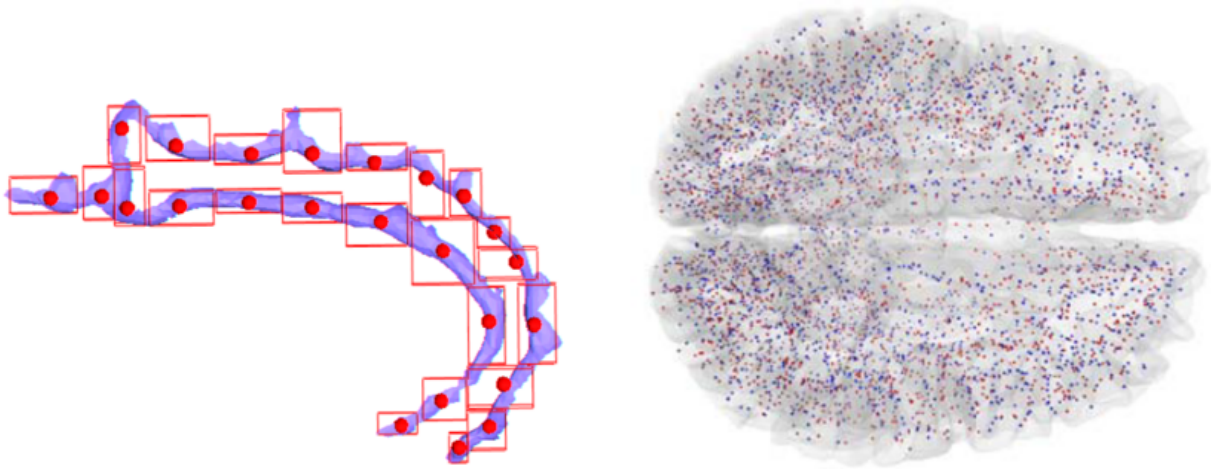


(a) (b)

Figure 4.6: (a) Surface segmentation based on the $mean$ and $Gauss$ operators to extract concave, convex, elliptic and hyperbolic regions. (b) Segmented surfaces and topology graph with sulci as graph nodes

In order to improve the comparison between two cortex models, an approach based on scatter plots and principal component analysis was proposed [Vivodtzev et al., 2006]. Curvature estimation was first improved by using a continuous approximation for each vertex of the surface as proposed by [Cohen-Steiner and Morvan, 2003]. The surface segments, corresponding to the sulci and obtained by partitioning using the curvatures, are converted into a point cloud. This is done by grouping the triangles of the same segment into a certain number of vertices representing the point cloud illustrated on Figure 4.7a. Once this point cloud has been constructed, an iterative association process is used to establish correspondences between the vertices of the atlas point cloud and those of the patient's point cloud as illustrated on Figure 4.7b. After this association of points between sulci of the two sets, a voting system is set up to determine the correspondence between the regions of the two models.

The process of associating two point clouds uses several calibration and deformation methods. The patient's point cloud is projected and distorted into the atlas space using principal component analysis and the thin-plate spline method. The voting system uses heuristics to determine the association of regions based on the unassociated vertices in each set and their distances from the vertices already associated. The result of our pipeline is a mapping between topological components of the input surfaces allowing transfer of annotations. By detecting features based on curvature and using them to establish correspondences between the surface, our method is capable of matching complex surface geometries such as brain cortices.



(a) (b)

Figure 4.7: (*a*) Extraction of the points representing a sulcus to form the point cloud. (*b*) Two point clouds superimposed in the same space. The blue points correspond to the points in the atlas and the red points correspond to the patient's points.

CHAPTER 5

Topological Data Analysis for Development of Flow Simulation Code

Flow turbulence is a phenomenon of major importance in fluid dynamics. It is characterized by chaotic changes in the motion of a flow, as illustrated in Figure 5.1. In this example, the turbulence moving the clouds is generated by a difference in the wind speed or direction between two wind currents in the atmosphere.

The study of turbulence has a drastic impact on numerous applications such as aeronautics, weather forecasting, climate modeling, material sciences, and astronomy. Although turbulence has been studied since the early stages of modern physics, its theoretical mastery remains incomplete [Baez, 2006], and the understanding of the Navier-Stokes equations, central to the description of fluid motion, is still considered a major open challenge in mathematics and physics, as proven by the Clay Mathematics Institute selecting it to be among its celebrated

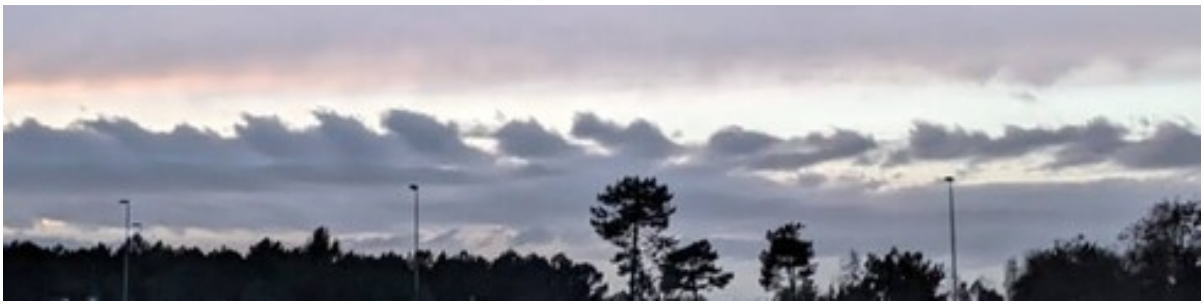


Figure 5.1: Kelvin-Helmholtz clouds look like waves in the ocean. These clouds form when there is a difference in the wind speed or direction between two wind currents in the atmosphere. This photograph of Kelvin-Helmholtz clouds was taken near Bordeaux, France.

Millennium Prize problems [Fefferman, 2000]. Thus, in engineering applications, the main practical solution available for the study of turbulence remains numerical simulation.

However, many different numerical solvers can be used to simulate a given flow configuration, with each solver being subject to several input parameters such as domain resolution, interpolation scheme and order, etc. When faced with such a wide variety of inputs, the main problem for users becomes the configuration of the simulation parameters itself. In particular, domain experts not only want to identify the solver configurations which produce the most realistic simulations, but they also want to discover configurations resulting in degraded but fast computations, which still produce simulations of acceptable realism. section 5.1 details the numerical components used in a simulation code developed at CEA. The fundamental problem behind such comparative analyses is that of comparing turbulent flows *quantitatively*. For instance, the quantitative realism of a simulation could be evaluated by comparing its outcome to a reference, either obtained by acquisition or by a highly detailed simulation considered as a ground-truth.

Analysis of turbulent flow simulations is tedious with standard imaging tools due to the chaotic nature of this phenomenon. For instance, turbulent flows which are considered similar at a high level by domain experts (in terms of the number and size of their vortices) are reported by classical metrics such as the L_2 norm as being very distant, as such pointwise measures are sensitive to mild geometrical variations in the data and miss global structural similarities in such a chaotic context. This observation motivates the consideration of alternative similarity estimation tools, which focus on the *structure* of the flow, rather than its raw geometry.

Computational topology and, more generally, Topological Data Analysis (TDA) as introduced in chapter 3 provide efficient techniques whose purpose is to precisely recover hidden implicit structural patterns and to enable their reliable comparison. As such, they provide potentially relevant alternatives to standard comparison measures used in scientific imaging, such as the L_2 norm.

5.1 Flow Simulation Code

This section presents the numerical components that code developers need to evaluate and handle when describing flow motion. The description can be general to any simulation code but it comes from the development of one production code at CEA [Bridel-Bertomeu, 2021] which benefited from this work on TDA to better handle the equations, the interpolation schemes, and the solvers. The data presented in section 6.1 also comes from this flow simulation code.

5.1.1 Euler Equations for Flow Simulation

We consider the two-dimensional compressible unsteady Euler equations for inviscid flows [Masatsuka, 2013]:

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = 0, \tag{5.1}$$

where the subscripts indicate differentiation, $\mathbf{U}$ is the vector of conservative dimensionless variables, and $\mathbf{F}$ and $\mathbf{G}$ represent the inviscid fluxes in the x and y directions, respectively. Those vectors are defined as:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}. \tag{5.2}$$

In the above expressions, t denotes time and x and y are the Cartesian coordinates. ρ denotes density, u and v denote the $x-$ and $y-$ coordinates of the velocity vector $\mathbf{w}$, E denotes the specific total energy, and p denotes the static pressure. The aforementioned mathematical model is described as it is implemented in an in-house simulation code whose primary capabilities are as a massively parallel structured solver using immersed boundary conditions [Bridel-Bertomeu, 2021].

The present study uses only regular Cartesian grids with constant grid spacings (grid of pixels) in both directions of space, Δx and Δy, and will not rely on any immersed boundary condition during the computations presented later. This being said, the finite-volume method [LeVeque et al., 2002; Toro, 2013; Trangenstein, 2007] is then employed for space discretization of the compressible Euler Equation 5.1.

The 2D turbulence investigated in our work at the heart of this study is generated using a Kelvin-Helmholtz instability, as described in [San and Kara, 2015] and simulated with high-order, low-dissipation reconstruction schemes of 5^{th}- and 7^{th}-order. The numerical fluxes between the cells are obtained using a variety of Riemann solvers detailed at the end of this section. Table 5.1 summarizes all the input parameters that the developer of the simulation code evaluates during design. To emulate turbulence in an infinite medium, all boundary conditions are set as periodic. To lower down the dimension of the field data, one common measure of turbulence in two dimensions that we will rely on is the local enstrophy $\mathcal{E}$, defined locally as the square of the flow vorticity:

$$\mathcal{E} = 0.5 \left| \nabla \times \mathbf{w} \right|^2 . \tag{5.3}$$

When solving the Euler equations (Equation 5.1) numerically, we start by interpolating the values of the flows at the cell interfaces. Then, we have to use an approximate Riemann solver [Toro, 2013] to solve the eponymous problem on those interfaces. The different numerical methods used to make these two calculations are exposed in subsection 5.1.2 and subsection 5.1.3.

5.1.2 Interpolation Schemes

One problem in numerically solving the schemes is being able to capture the strong discontinuities while capturing the small scales of the turbulence. In addition, we want to be as accurate as possible in our interpolation. To do this, researchers and engineers have developed several high-order reconstruction methods.

A common scheme for solving compressible flows in the presence of strong discontinuities

is the *Weighted Essentially Non-Oscillatory (WENO)* scheme [Liu et al., 1994]. Several variants of this scheme have been introduced to improve its performance [Jiang and Shu, 1996] [Hu et al., 2010] [Henrick et al., 2005]. A second family of interest is that of the *The Targeted Essentially Non-Oscillatory (TENO)* schemes developed by [Fu et al., 2016] is a high-order scheme that attempts to capture turbulence better than WENO-Z while capturing strong discontinuities (shocks). It uses new ingredients such as a better discrimination between strong discontinuities and small-scale turbulence [Hu and Adams, 2011]. We are particularly interested in these two families: the well-known robust but dissipative WENO-Z [Borges et al., 2008] and the TENO (T for *Targeted*) [Fu et al., 2016], which better discriminates scales [Hu and Adams, 2011].

5.1.3 Solvers

Riemann problems at the interfaces between the cells of the mesh have to be solved. One solution is to use an exact Godunov solver [Toro, 2013] which takes into account a large number of nonlinear operations too expensive, however, when calculating complex flows. Rather, researchers and engineers are interested in approximate Riemann solvers.

The most used approximate solvers can be grouped into three large families: Flux Difference Splitting (FDS), Flux Vector Splitting (FVS), and Flux Type Splitting (FTS) [Toro, 2013]. In this section, we will focus on two types of solvers in particular: Flux Difference Splitting solvers that work as a finite volume method to solve the Riemann problem, and Flux Splitting Riemann solvers that combine the qualities of the other two families by separating kinematic and acoustic scales. This type of solvers captures the strong discontinuities and calculates the boundary layers accurately.

HLL ***(Harten, Lax, and van Leer)*** FDS scheme developed by Harten et al. [Harten et al., 1983]. This scheme does not take into account contact discontinuities, i.e., lines crossing two states. For turbulent phenomena, the interface between vortices will therefore be less described.

Roe and HLLC ***(Harten, Lax, and van Leer with Contact)*** FDS type schemes developed by [Toro et al., 1994] [Roe, 1981]. These two schemes are robust and thus allow reproducing the strong discontinuity (shock) and taking into account the discontinuities of contact. Thus, with these schemes, the reconstruction of the vortices represented in our flows is performed with more accuracy than with the HLL solver. In his study on low-speed Riemann solvers, [Qu et al., 2014] noticed that the solvers were unable to obtain physical solutions. Therefore, there is a need for approximate Riemann solvers to accurately reconstruct both low and high-speed flows. This is why we are interested in two flux type splitting (FTS) solvers, which take into account all velocities to obtain low Mach and high Mach physics solutions.

AUSM$^+$-UP ***(Advection Upstream Splitting Method +UP)*** By adding improvements [Liou, 2006] to the AUSM+ [Liou, 1996] solver, Liou increases its level of accuracy for all speeds. This new solver takes into account contact discontinuities, also reconstructs strong discontinuities, and gives physical solutions for all speeds.

SLAU2 ***(Simple Low-Dissipation AUSM 2)*** The analysis of the dissipation pressure term of the AUSM+ [Shima and Kitamura, 2009] shows that it is too high for low speeds. The author

Parameter	Resolution	Order	Time	Solver	Scheme	Total
Value	256 512 1024	5 7	t_0 t_1 t_2	HLL SLAU2 AUSM$^+$-UP Roe HLLC	TENO WENO-Z	
Number	3	2	3	5	2	180

Table 5.1: Parameter space of the simulation code leading to a total of 180 members for the ensemble dataset used in this study.

decided to control the pressure flux and implemented the SLAU solver. This has been extended [Kitamura and Shima, 2013] so that the dissipation becomes proportional to the Mach number. This solver takes into account contact discontinuities, also reconstructs strong discontinuities, and gives physical solutions for all speeds.

5.2 Case Study on a Kelvin Helmholtz Instability

An ensemble dataset representing Kelvin Helmholtz Instabilities (KHI) has been generated for which we state the challenges in understanding such phenomena and provide theoretical hypotheses that our experimental protocols have to verify.

5.2.1 Data Description

The initialization of the KHI for the dataset was generated with two fluids of different densities (ρ_1, ρ_2) (Figure 5.2(a)). The different velocities of opposite direction ($\{u_1, v_1\}$, $\{u_2, v_2\}$) of the fluids create a shearing zone where the turbulence appears with the KHI (Figure 5.2b). While the instability develops over time, the main vortices grow (Figure 5.2c). After a longer simulation time, the main structures keep evolving (Figure 5.2d), and a large number of small-scale vortices appear in the vicinity of large-scale vortices (Figure 5.2e), leading to a complex turbulent flow. This variation in vortex scale, in addition to the chaotic flow geometry, is notoriously challenging for the analysis of turbulent flows. This is why the topological analysis of the enstrophy field is studied to capture features in the KHI and thus help the code developers to compare them.

The ensemble dataset available at [Vivodtzev, 2022] has been generated on the CEA's supercomputers [CEA EXA] using the numerical ingredients presented in subsection 5.1.1 and summarized in Table 5.1. Each simulation was executed in parallel using 16 MPI processes and was distributed over the supercomputer. The total simulation took about 745 CPU hours. The raw data has been dumped on disk with the metadata stored in XDMF files and the scalar fields in HDF5 files, leading to 14 GB for the entire ensemble dataset. We processed these results to extract the enstrophy scalar field (Equation 5.3) and stored it to a VTK file format [Kit, 2003] using an image data structure for regular grids (VTI). This reduces the entire ensemble to 600 MB.

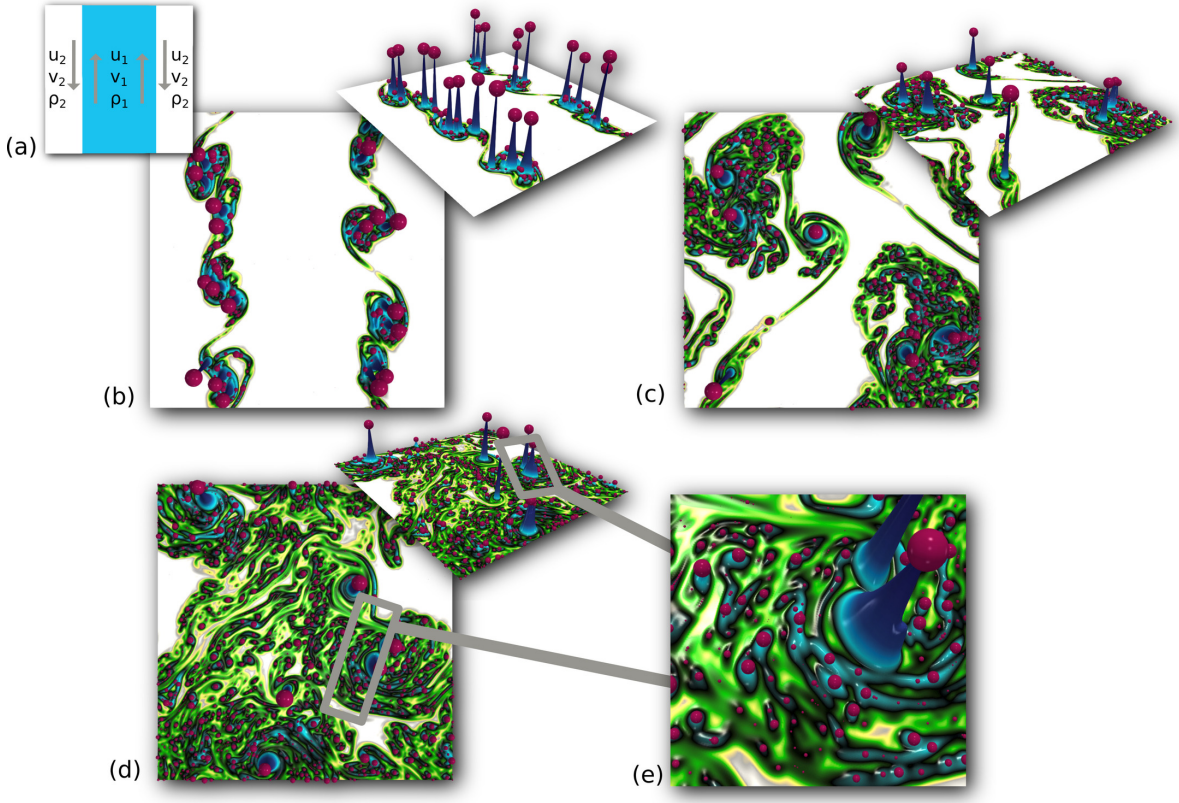


Figure 5.2: Initialization of the Kelvin-Helmholtz instability (a). This simulation was obtained with the AUSM$^+$-UP solver with a TENO 5 order interpolation at physical times 0.25(b), 0.75(c) and 1.25(d). Red spheres scaled by the persistence represent the maximum critical points. Zoom of the turbulence structures (e).

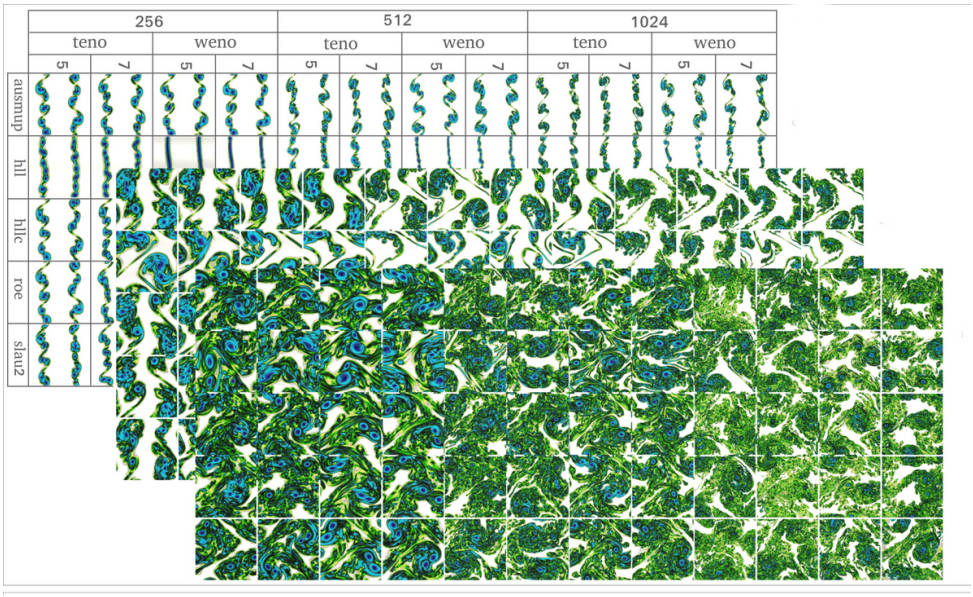


Figure 5.3: The 180 members of the ensemble obtained with variations of timesteps, interpolation schemes, orders, resolutions and Riemann solvers summarized in Table 5.1

The ensemble dataset corresponds to different computational configurations for the same turbulent instability. The simulation code handles different parameter types such as scalars or enumerations, which allow the users to compute various numerical simulations in the same parametric study. The resolution of the 2D regular grid, the simulation time, the interpolation scheme, the order of interpolation, and the Riemann solvers presented in section 5.1 are our different parameters. Table 5.1 details the parameter types and values, as well as the number of samples per parameter, leading overall to an ensemble of $3 \times 2 \times 3 \times 5 \times 2 = 180$ members, illustrated in Figure 5.3.

Each parameter value of Table 5.1 used to run the simulation has been stored as meta-data in the VTI files (i.e., *Field Data* in the VTK terminology) to keep track of the computational configuration for later analysis down the pipeline. In order to ease the exploration of the ensemble dataset, we defined a SQL-type database using the cinema database feature of TTK [Bin Masood et al., 2019]. This representation facilitates the extraction of sub-samples of the ensemble based on standard SQL queries on the simulation parameters.

5.2.2 Problem Statement

Flow understanding in the context of aerodynamic vehicle design is well known for its high number of constraints that are nowadays most often handled with the help of computational techniques. In the aeronautical world, for instance, engineers today face an incredible challenge wherein they have to be able to predict, at the same time, integral quantities at the wall of the vehicle, such as heat flux or pressure, as well as three-dimensional phenomena, such as flow discontinuities and turbulence. In other words, engineers have to deal with multiple types of physics and phenomena that have markedly different length- and time-scales, but whose

interactions are still of great importance to the accuracy of their predictions. With limited time and resources to conduct the computer-aided simulations, the traditional approach is to rely on numerical strategies that temporally average most of the three-dimensional phenomena and rely more or less on models of turbulence to yield a fast and reasonable forecast.

Even in such a context of approximate simulations, the choice of the ingredients of the numerical recipe matters - methods of reconstruction, Riemann solvers, etc. Making the right choices can indeed bring a significant increase in fidelity to the engineer, especially in terms of turbulence, by lessening the need for modeling and henceforth bringing more margin in the design of the vehicle. Turbulence is, however, by nature a chaotic phenomenon, and conducting a systematical study of the impact of the different numerical ingredients thereupon might prove tricky for a simple reason: beyond a certain level of accuracy, everything will *look* the same. Detecting the benefits of one method compared to another in that situation will be next to impossible - that is, with traditional techniques.

In [Nauleau et al., 2022b] we propose to use the ability of topological analysis to discern features that stay otherwise hidden in traditional fluid dynamics postprocessing to help with the choice of the right numerical ingredients.

5.2.3 CFD Hypotheses

This section introduces the hypotheses provided by CFD experts, documenting their expectations about ensemble flow variability.

(Hypothesis H1) TENO induces more turbulence (i.e. more critical points) than WENO-Z, for all configurations, no matter what the resolution, time, or solver is.

(Hypothesis H2) Order 5 and 7 are equivalent for Kelvin Helmholtz instabilities. We therefore try to find an independence of the orders using our topological analysis.

(Hypothesis H3) The HLL solver should provide a significantly distinct description for all configurations. It's a dissipative solver and doesn't take into account the dissipative contact discontinuities.

(Hypothesis H4) The HLLC and Roe solvers should provide equivalent outputs for all configurations. These solvers are of FDS type; they take into account the contact discontinuities but do not adapt to all Mach.

(Hypothesis H5) The SLAU2 and AUSM$^+$-UP solvers should provide equivalent outputs for all configurations.

The above hypotheses are direct consequences of observations or design choices. For instance, the TENO scheme has been reported to capture turbulence more accurately [Peng et al., 2021], which is expressed by Hypothesis H1. Similar kinetic energy curves (Figure 5.4) have been reported for orders 5 and 7, which is expressed by Hypothesis H2. The HLL solver, which is a dissipative approach, is known to model contact discontinuities poorly in contrast

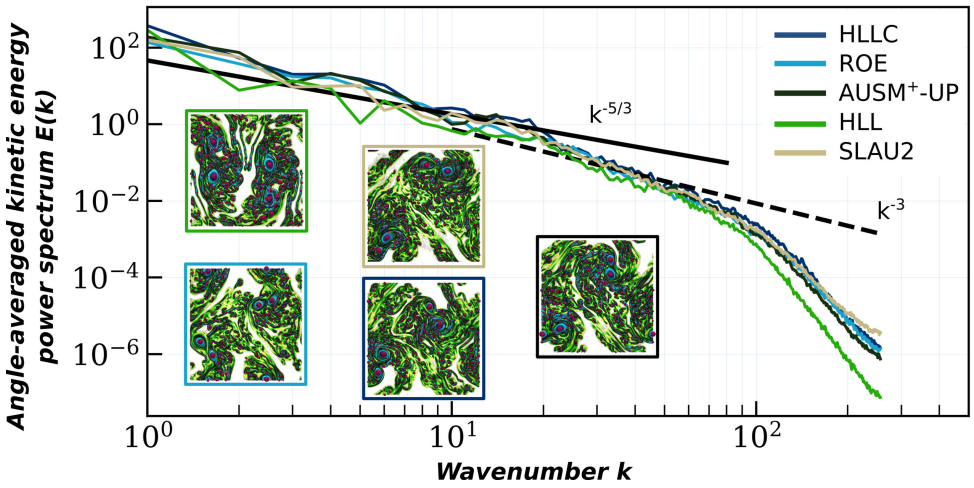


Figure 5.4: Baseline analysis by angle-averaged kinetic energy power spectrum for different solvers and various fixed simulation parameters (WENO-Z, order 7, t_2, 512×512).

to more recent solvers, which is expressed in Hypothesis H3 [Toro, 2013]. Finally, unlike the SLAU2 and AUSMUP (FTS type) solvers, the HLLC and Roe (FDS type) solvers have been reported to provide unphysical results at both low and high velocities (resulting in local oscillations in pressure and density), which is expressed in Hypotheses H4 and H5.

From a practical point of view, the validation of these hypotheses has a major impact for engineers when setting up their simulations. For instance, the validation of Hypothesis H1 would justify the usage of a more computationally expensive scheme (TENO), while the validation of Hypothesis H2 would enable the usage of less computationally expensive orders (5 instead of 7). Finally, the validation of Hypotheses H3, H4, and H5 would help engineers properly select the most appropriate solvers based on their flow characteristics. Then, overall, the validation of these hypotheses would provide reliable rules-of-thumb for the tuning of the solvers to achieve the best balance between accuracy and speed.

5.2.4 Baseline Analysis

Traditional approaches for turbulent data analysis (Figure 5.4) are based on an average of quantities of interest, such as flow energy (**??**). The L_2 norm is another established distance for comparing scalar fields. Both strategies bear similarities in their averaging artifacts: they cannot distinguish the contribution of small structures from the global flow, because these are masked by the weight of larger vortices. Moreover, the L_2-norm is also very sensitive to mild geometric variations, whereas the chaotic nature of turbulent flows induces major geometric variations between ensemble members. This motivates the usage of topological methods to capture features in the KHI that will help us compare the members (section 5.5). In the remainder, we will systematically compare our protocols based on topological distances (section 5.4) to the L_2 norm, considered as the baseline approach.

5.3 Related Work on TDA for Turbulent Flow

In the work presented in [Nauleau et al., 2022b] we provide a comprehensive experimental evaluation of the above CFD hypotheses, i.e., to assess the suitability of topological data representations and their associated analysis tools for the quantitative comparisons of turbulent flows. We shall focus on a specific type of turbulence that expresses itself in two dimensions (Kelvin-Helmholtz instability): it is a fair representative of generic turbulence (*a.k.a.* three-dimensional viscous turbulence) and allows for affordable high-resolution simulations to feed our study.

5.3.1 Turbulent Flow Simulation

Turbulence is ubiquitous in nature, at all scales, from Higgs-Boson condensates [Kraichnan, 1967] to a stirred cup of coffee to geophysical flows [Kraichnan and Montgomery, 1980] to galaxy formation. While a significant body of literature in graphics [Bai et al., 2021; Kim et al., 2008; Zhang et al., 2014] has focused on the efficient generation of visually plausible turbulence, we focus in this work on the direct numerical simulation of the underlying physical equations, for engineering applications. One motivation to study turbulence, and in particular two-dimensional turbulence, shared by all investigators is that despite its practical applications, its omnipresence, and the years of scrutiny it was subjected to, it remains one of the most important unsolved problems of classical mechanics. A special distinction of 2D turbulence is that it is never realized in nature that, unless strongly constrained, always has some degree of three-dimensionality, but rather it only exists in computer simulations.

Two-dimensional turbulence has thus been studied extensively by the latter means, *e.g.*, for its importance as an idealization of meteorological flows [Boffetta and Ecke, 2011], its role in the confinement of thermonuclear plasmas [Kraichnan and Montgomery, 1980], but also as a cost-effective numerical testing ground for three-dimensional flow dynamical theories [Tabeling, 2002]. Most such studies focus either on validating predictions of theorists [Kraichnan, 1967; Lilly, 1989] or on providing insights into the dynamic behavior of 2D eddies thanks to high-resolution simulations [Lilly, 1989; Maltrud and Vallis, 1991].

The simulations are usually analyzed by considering macroscopic quantities such as the enstrophy, as presented in Equation 5.3, or by considering the Fourier decomposition of the 2D field, similar to Figure 5.4. Note that these integral indicators make it near impossible to compare and/or classify the results of such a parametric study. These classical indicators all integrate the data to some extent and, faced with two numerical realizations produced using different numerical settings, make it near impossible to classify the results. Some efforts have been made recently [San and Kara, 2015] to provide users with some guidelines to best choose the numerical methods and parameters for the simulation of 2D turbulence, but still using, mostly, the aforementioned integral, inaccurate indicators.

5.3.2 Evaluation of Topological Representations

The utility of TDA has already been demonstrated in a number of analysis and visualization tasks [Heine et al., 2016], with examples of successful applications in combustion [Bremer et al., 2011; Gyulassy et al., 2014; Laney et al., 2006], material sciences [Favelier et al., 2016; Gyulassy et al., 2007, 2015], bioimaging [Anderson et al., 2018; Bock et al., 2018; Carr et al., 2004], quantum chemistry [Bhatia et al., 2018; Guenther et al., 2014; Olejniczak et al., 2019], or astrophysics [Shivashankar et al., 2016; Sousbie, 2011].

In particular, the critical points of flow vorticity indicators have been reported to appropriately capture the center of vortices [Bridel-Bertomeu et al., 2019; Kasten et al., 2011], as well as their importance, with the notion of *topological persistence* [Edelsbrunner et al., 2002]. Such results provide additional evidence and consolidate the intuition that TDA could be a relevant framework for comparing turbulent flows. Specifically, we document the usage of the persistence diagram of the maxima of flow enstrophy (an established indicator of vorticity for two-dimensional flows, section 5.1) for the topological representation of 180 members of an ensemble of hydrodynamic turbulent flows, generated by a coarse sampling of the parameter space of five distinct solvers. We hereby propose five main hypotheses, as detailed in Section section 5.2, which are reported by domain experts. These hypotheses describe their expectations with regard to the variability among the flows generated by the distinct solver configurations. Then, we describe three evaluation protocols in Section section 5.4 designed to assess the validation of the above hypotheses by standard comparison measures (L_2 norm) on one hand, and by topological methods on the other. Specifically, these protocols exploit the persistence curve, the L_2-Wasserstein distance between persistence diagrams [Turner et al., 2014] detailed in subsection 5.3.5 , and k-means in the L_2-Wasserstein metric space [Vidal et al., 2019]. Finally, we document the results of these protocols on the input ensemble in section 5.5.

We believe that the insights reported by our study bring strong experimental evidence of the suitability of TDA for representing and comparing turbulent flows, thereby providing confidence in its usage by the fluid dynamics community. Moreover, our flow data and evaluation protocols provide to the TDA community an application-approved benchmark for the evaluation and design of further topological distances in the future.

5.3.3 Topological Methods for Flow as Vector Fields

A large body of literature has been dedicated to the analysis and visualization of flow data as a vector field with topological methods, and we refer the readers to a series of surveys on the topic [Bujack et al., 2020; Garth and Tricoche, 2006; Laramee et al., 2007; Pobitzer et al., 2011; Scheuermann and Tricoche, 2005; Wang et al., 2016], including a recent iteration [Günther and Baeza Rojo, 2021]. A substantial line of work [Otto et al., 2010, 2011; Petz et al., 2012] focused on extending topological techniques to uncertain vector fields, where flow variability is encoded via a pointwise estimator (*e.g.* an histogram) of an *a priori* vector distribution, but only a few techniques explicitly focused on the analysis of flow variability in an ensemble.

Specifically, several comparative visualization techniques have been proposed [Guo et al.,

2016; Hanser et al., 2018; Hummel et al., 2013; Jarema et al., 2016; Lohfink and Garth, 2020; Schlemmer et al., 2007; Schneider et al., 2012; Zheng et al., 2019]. [Ferstl et al., 2016] investigated the global structure of flow ensembles by proposing a clustering approach of the members based on a measure of streamline similarity. However, these techniques assume a mild geometrical variability within the ensemble and are therefore not suited to highly turbulent flows as studied in this work, where the geometry of the features (streamlines, vortices) chaotically changes from one ensemble member to the other (Figure 5.3), even upon only slight variations of the simulation input parameters.

5.3.4 Topological Representations for Flow as Scalar Fields

In certain application contexts, CFD experts often prefer to focus their analysis on scalar descriptors generated from the flow, such as the kinetic energy (Figure 5.4) or the enstrophy (Equation 5.3). Such a transition to a scalar descriptor enables them to leverage the existing tools for scalar data analysis. For instance, several authors [Bridel-Bertomeu et al., 2019; Kasten et al., 2011] have shown that, given a relevant vorticity scalar descriptor, the center of the flow vortices could be reliably extracted and tracked over time, which supports the idea that topological methods for scalar data can be useful for the description of specific flow features. As presented in chapter 3, many popular topological representations can be used to represent flows described as scalar fields. Those include persistence diagrams [Edelsbrunner and Harer, 2009], Reeb graphs [Biasotti et al., 2008], contour trees [Tarasov and Vyali, 1998], or Morse-Smale complex [Edelsbrunner et al., 2003a].

5.3.5 Comparison Methods for Ensembles

The comparison of scalar field ensembles can be achieved with a large variety of metrics, as surveyed in [Yan et al., 2021] for different topological descriptors. Interesting metrics, inspired by the literature in optimal transport [Kantorovich, 1942; Monge, 1781], the Wasserstein distance between persistence diagrams [Edelsbrunner and Harer, 2009], and its variant, the Bottleneck distance [Edelsbrunner et al., 2002], have been extensively studied. In practice, it enables users to compare ensemble members based on their persistence diagrams.

Several techniques have been proposed for summarizing the topological features in an ensemble or analyzing their variability. For instance, [Favelier et al., 2018] and [Athawale et al., 2019] introduced approaches for analyzing the variability of critical points and gradient separatrices, respectively. Other approaches aimed at summarizing an ensemble of topological descriptors by computing a notion of average descriptor, given a specific metric.

The L_2-Wasserstein distance [Cohen-Steiner et al., 2005] allows comparing two datasets f_i and f_j with a metric between them. This distance is derived from a bipartite assignment optimization problem between the points of the two diagrams to compare. First, the input diagrams $\mathcal{D}(f_i)$ and $\mathcal{D}(f_j)$ are typically augmented by adding to each diagram the diagonal projection $\Delta(p)$ of each off-diagonal point $p = (x, y)$ of the other diagram. From a practical

point of view, this augmentation introduces dummy features with zero persistence. Therefore, it does not modify the topological description provided by the diagrams. However, it presents the advantage of appropriately balancing the size of the two input diagrams, i.e., $|\mathcal{D}(f_i)| = |\mathcal{D}(f_j)|$, which will facilitate their comparison. We define the L^q-Wasserstein distance, or *Earth Mover's distance*, used originally in the field of Optimal Transportation Theory [Kantorovich, 1942; Monge, 1781] as:

$$W_q\big(\mathcal{D}(f_i), \mathcal{D}(f_j)\big) = \min_{\phi \in \Phi} \left(\sum_{p_i \in \mathcal{D}(f_i)} d_q\big(p_i, \phi(p_i)\big)^q \right)^{1/q}, \tag{5.4}$$

where Φ is the set of all possible assignments ϕ mapping each point $p_i \in \mathcal{D}(f_i)$ to a point $p_j \in \mathcal{D}(f_j)$. Since the augmented diagrams are balanced ($|\mathcal{D}(f_i)| = |\mathcal{D}(f_j)|$), W_q can be efficiently computed via assignment optimization, for which exact [Munkres, 1957] and approximate [Bertsekas, 1981; Kerber et al., 2016] implementations are publicly available [Bin Masood et al., 2019; Tierny et al., 2017b] with a fast approximation scheme [Vidal et al., 2019].

Once the L_2-Wasserstein distance between two diagrams $\mathcal{D}(f_i)$ and $\mathcal{D}(f_j)$ is available (noted $W_2\big(\mathcal{D}(f_i), \mathcal{D}(f_j)\big)$), more advanced geometrical objects can be considered, such as *Wasserstein barycenters* [Turner et al., 2014; Vidal et al., 2019]. They are representative diagrams as they minimize the sum of their distance to the ensemble. Consequently, they can be considered reliable representatives of the ensemble. The concept of the barycenter is instrumental in the development of clustering algorithms. In particular, the k-means algorithm [Celebi et al., 2013; Elkan, 2003] can be easily extended by employing the W_2 to measure distances between diagrams and by designating the barycenter of the cluster as the cluster centroid at each iteration of the k-means.

5.4 Evaluation Protocols

In this section, we present 3 protocols which can be used to verify the hypotheses detailed in subsection 5.2.3. One can directly use these algorithms on the ensemble dataset. It corresponds to (i) the separation of the schemes and the independence of the orders, (ii) the unique behavior of the HLL solver, and (iii) similarities in the class of solvers.

5.4.1 Persistence Curves

With this protocol, we want to validate hypothesis H1 (subsection 5.2.3) to discriminate between the TENO and WENO-Z interpolation schemes based on differences in the enstrophy field. Figure 5.5 illustrates this protocol on two input scalar fields (X-axis: persistence threshold, Y-axis: number of maxima more persistent than X). S_1 is generated with two Gaussian functions with noise, and S_2 with 10 Gaussian functions with a stronger noise. The maxima critical points are represented by red spheres scaled by persistence, and the vertical lines correspond to small

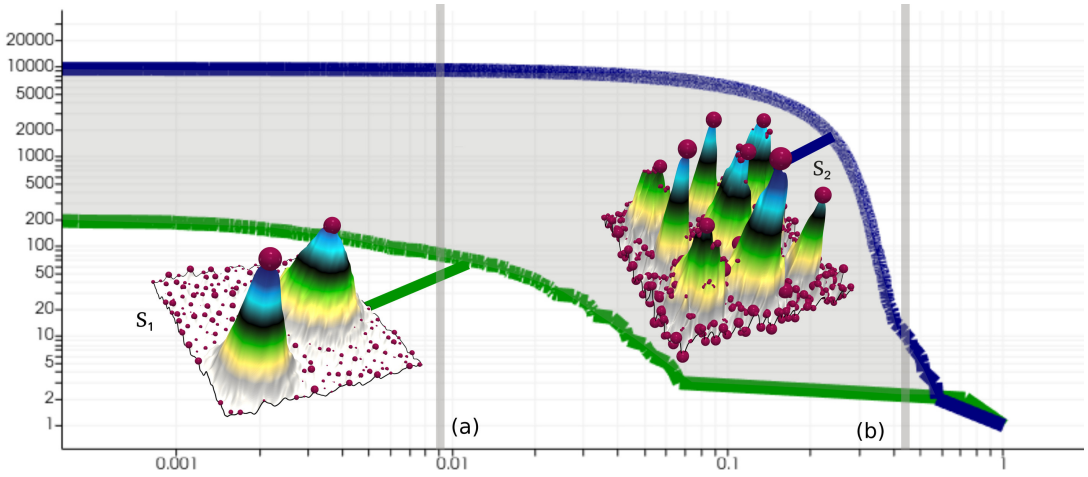


Figure 5.5: Persistence curves for two input scalar fields based on Gaussian functions and noise. Maxima critical points are represented by red spheres scaled by persistence. Vertical line (a) corresponds to small persistence critical points and (b) to high persistence.

persistence critical points, line (a) and high persistence, line (b). The gray area is the integral difference between the two curves.

With this protocol, we also want to validate hypothesis H2 (subsection 5.2.3) to confirm the independence of the orders. To better characterize the vortices influencing the turbulence, we use persistence curves (section 3.5). These curves will allow us to threshold the structures (the eddies) at different scales and thus to easily compare the number of small (Figure 5.5a) and large (Figure 5.5b) eddies using the integral of the persistence curve.

For the differentiation of the schemes, we take 5 simulation configurations where the physical time (t_0, t_1, t_2), the resolution (256×256, 512×512, 1024×1024) and the order (5,7) are fixed per sample (Table 5.1). The variation is the interpolation scheme (TENO, WENO-Z). For the order independence, 5 configurations are also chosen by fixing the physical time (t_0, t_1, t_2), the resolution (256×256, 512×512, 1024×1024), and the scheme (TENO or WENO-Z). The variation is done on the order (5,7). Besides different input variations, this protocol is the same for testing H1 and H2.

The persistence curves are generated for all the samples. Then, we average the 5 persistence curves (one per solver) to obtain 2 average persistence curves with respect to the variable parameters (schemes or orders). Finally, we compute the difference of the integrals between the two averaged curves (the gray area on Figure 5.5). The small values on the curves under a persistence of 10^{-6} correspond to numerical noise coming from the different simulation steps. They are removed from the computation of the integral with a threshold at 10^{-6} (Figure 5.5a). The integral curve difference corresponds to our metric, which allows for a precise description of the similarity in the topology of the critical points. The larger the integral, the more different the topology of the flow is. Thus, to verify hypothesis H1 related to the scheme, we want the difference of the integrals to be high. To verify hypothesis H2 related to the orders, we want the difference of the integrals to be close to zero.

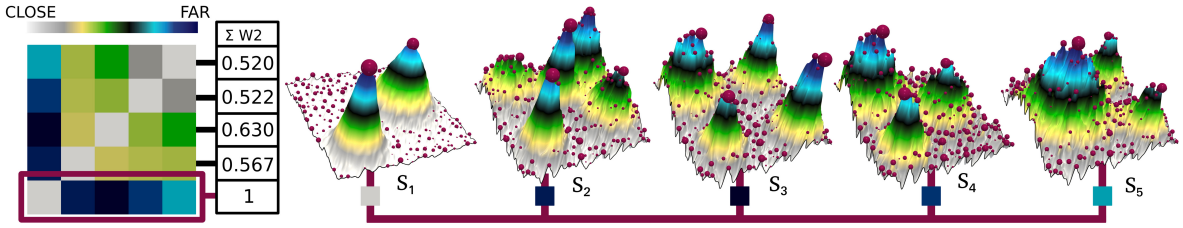


Figure 5.6: Wasserstein distance matrix for five input datasets generated with Gaussians and varying noise. The distance matrix is computed with respect to the scalar-field that maximizes the distances.

5.4.2 Outlier Distance Profile

With this protocol, we want to validate hypothesis H3 (subsection 5.2.3), which means that for all the simulation configurations, the HLL solver will be very different from other solvers in its description of the Kelvin-Helmholtz instabilities. Figure 5.6 illustrates this distance matrix for five inputs: S_1, S_2, S_3, S_4, and S_5 generated respectively with two, five, four, and three Gaussians with varying noise. The sum of each matrix line is computed and normalized with respect to the scalar field that maximizes the distances, here S_1. We see that S_1 with only two Gaussians is very far from the other datasets.

For this protocol, we take 5 simulation configurations where we fix the reconstruction (TENO or WENO-Z), the physical time (t_0, t_1, t_2), the mesh (256×256, 512×512, 1024×1024), and the order (5 or 7) and vary the solvers. The 5 different computations describing the same turbulent flow obtained with the solvers (HLL, SLAU2, AUSM$^+$-UP, HLLC, and Roe) are analyzed regarding the enstrophy.

A distance is used to compare the topology of the enstrophy. Many methods can be used to compute such a distance, but in this protocol, we focus on 2 metrics: the L_2-norm distance directly on the values of the enstrophy and the Wasserstein distance on the persistence diagrams. One can inject other distances if needed. For the Wasserstein, the saddle-maximum persistence diagram is computed on each result. Then, they are grouped in a unique dataset to compute a persistence diagram distance matrix (Figure 5.6). For the L_2-norm, a distance matrix is also created where a line corresponds to the distance in the enstrophy field from one solver to the others.

Thus, the sum of the distances from one solver to the others is computed by summing the distances on one line of the matrix. The total distance of one solver to the others, for all configurations, is simply the sum of all these sum distances for every line of the matrix which corresponds to the same solver. We finally obtain one global distance per solver for all configurations. Finally, the difference between the distance of the HLL and the distance of the maximizer (the second value if HLL is the maximum) gives a separation score. If the difference is positive, then hypothesis H3 is verified, whereas it is not if negative, because it means that another solver generates a flow topologically more different than the HLL. With this protocol, the best separations are obtained for high absolute values.

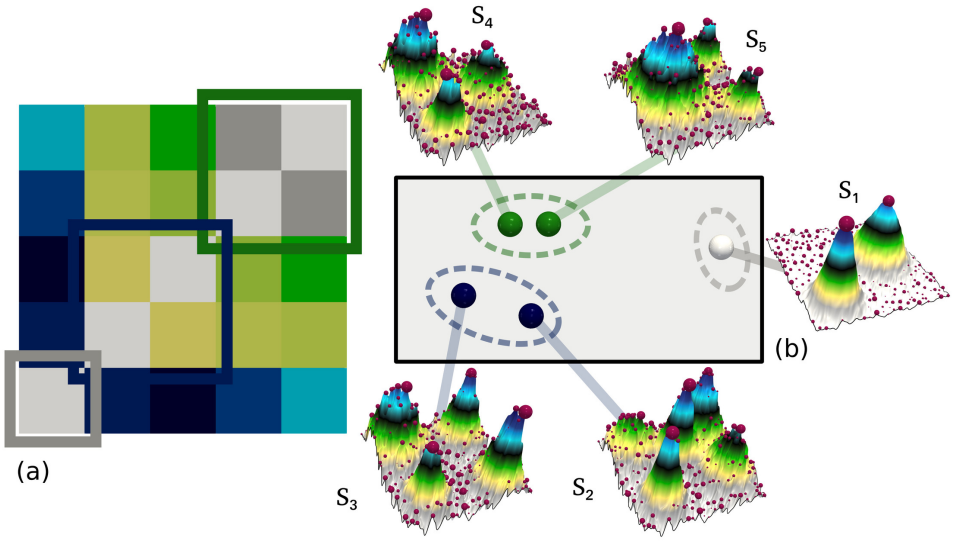


Figure 5.7: (a) Wasserstein distance matrix for five input datasets generated with Gaussian functions with different noise levels. (b) Point cloud of the inputs in the Wasserstein distance space colored according to the clusters obtained with the k-means clustering method.

5.4.3 Unsupervised Classification

With the last protocol, we want to validate hypotheses H4 and H5 (subsection 5.2.3). We want to verify that the simulations with the Roe and HLLC solvers are topologically close (hypothesis H4) and the simulations with the AUSM$^+$-UP and SLAU2 solvers are topologically close (hypothesis H5). To do so, three clustering methods will be used based on Wasserstein distances and the L_2-norm. Figure 5.7 illustrates the Wasserstein distance matrix for five inputs: S_1, S_2, S_3, S_4, and S_5, generated respectively with two, five, four, and three Gaussian functions with different noise levels. The point cloud of the inputs in the Wasserstein distance space is colored according to the clusters obtained with the k-means clustering method. We can see that each terrain in a cluster has the same number of Gaussians and level of noise.

For the first two clustering methods, we start by computing the distance matrix with the protocol of the outlier distance profile (subsection 5.4.2) using successively the Wasserstein distance and L_2-norm matrices (Figure 5.7a). We apply a dimension reduction to project the distances of the matrix according to 2 components. This projection is used to generate clusters of the matrices with a k-means algorithm as illustrated on Figure 5.7b. The third clustering method uses the persistence diagrams directly without using the distance matrix. All the persistence diagrams are merged into a single dataset to compute the Wasserstein distances between each diagram. The barycenter of the persistence diagram is then used to directly compute a cluster, without dimension reduction, in the Wasserstein metric space [Vidal et al., 2019] with the W_2 distance. Then a k-means algorithm is applied.

With these three classification methods, we obtain different associations of our configurations. Each association is going to be scored with a measure of similarities between the clusters regarding a reference cluster using the Rand Index [Rand, 1971]. This Rand Index has a value between 0 and 1, with 0 indicating that two clusters do not agree on any pair of points and 1

indicating that the data clusters are exactly the same. Based on the properties of the solvers used in the simulation code and detailed in subsection 5.1.1, we define our reference cluster such that the first partition contains the AUSM$^+$-UP and SLAU2 solvers, the second partition the HLLC and Roe solvers, and the third partition the HLL solver. The Rand Index is computed for each configuration and averaged per clustering method. This enables the ranking of the different solver behaviors. If the average Rand Index score is close to 1, then both hypotheses H4, showing similarity between the AUSM$^+$-UP and SLAU2 solvers, and H5, showing the isolation of the HLL solver, are verified.

5.5 Interpretation of the TDA Protocols

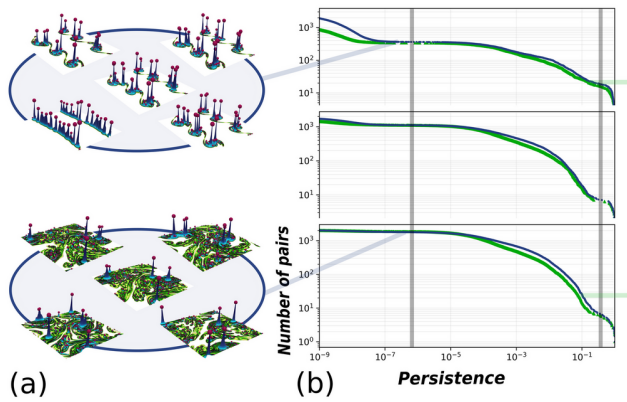
This section presents our experimental results and their interpretations for the protocols presented in section 5.4, applied on the ensemble data described in section 5.2.

5.5.1 Persistence Curve Study

We applied protocol 1 using the persistence curves, on our ensemble dataset of Kelvin-Helmotlz instability to verify the hypotheses of separation of the schemes (H1) and the independence of the orders (H2) as described in subsection 5.2.3. The input parameters are setup as detailed in section 5.4, generating 36 studies.

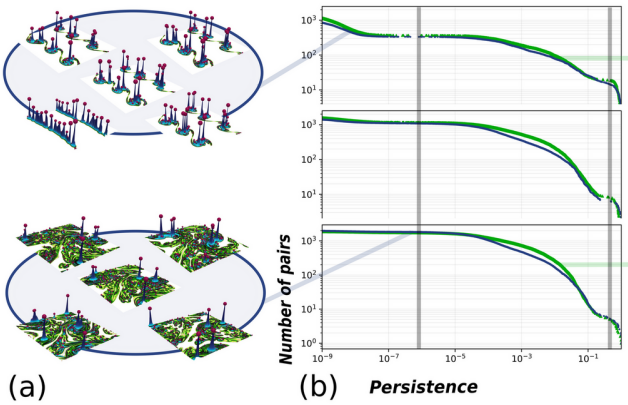
The Figure 5.8 shows the average persistence curves for the TENO (blue curves) and WENO-Z (green curves) at t_0 *(top)*, t_1 *(middle)*, t_2 *(bottom)*. The vertical lines on the curves correspond to critical points of small *(left)* and high *(right)* persistence. The integral differences between average persistence curves for all variations are shown in the table of Figure 5.8d. The integral difference, between the average curves, obtain results between $[-5.6, 5.2]$, which demonstrates differences on the topology of the enstrophy between the interpolation methods as expected. For the scheme comparison, most of the averaged persistence curves for the TENO schemes (blue curves) are above the WENO-Z curves (green curves). Hypothesis H1 is verified on the KHI ensemble dataset.

For the study on the independence of orders illustrated on Figure 5.9, we see that the averaged persistence curves are often close. However the integral differences obtained for this study show larger values for the WENO-Z, *i.e* in between $[-0.5, 8.1]$ in the table of Figure 5.9d. This analysis highlights that orders play a more important role, in terms of topology of the vortices, for WENO-Z than for TENO. Moreover, we observe that this difference tends to increase at t_2 for both studies confirming that the flow is composed of a larger number of vortex as the simulation evolves. Hypothesis H2 is verified for the TENO solvers but not for the WENO-Z.



ORDER	GRID	TIME	INT
5	256	0	-3.12
		1	-2.05
		2	-0.81
	512	0	-2.42
		1	-1.91
		2	-1.19
	1024	0	0.38
		1	-4.15
		2	-5.21
7	256	0	-1.66
		1	-1.81
		2	0.43
	512	0	0.76
		1	-0.67
		2	1.74
	1024	0	0.24
		1	3.25
		2	5.06

Figure 5.8: Scheme comparison between TENO and WENO-Z showing terrain views, average persistence curves and integral differences between persistence curves for all variations.



SCHEME	GRID	TIME	INT
TENO	256	0	0.42
		1	0.83
		2	2.24
	512	0	0.43
		1	-0.1
		2	1.2
	1024	0	-0.34
		1	-1.97
		2	-2.11
WENO-Z	256	0	1.89
		1	1.07
		2	3.49
	512	0	3.63
		1	1.13
		2	4.15
	1024	0	-0.47
		1	5.42
		2	8.1

Figure 5.9: Order comparison between 5^{th} and 7^{th} order for TENO and WENO-Z schemes showing terrain views, average persistence curves and integral differences between persistence curves for all variations.

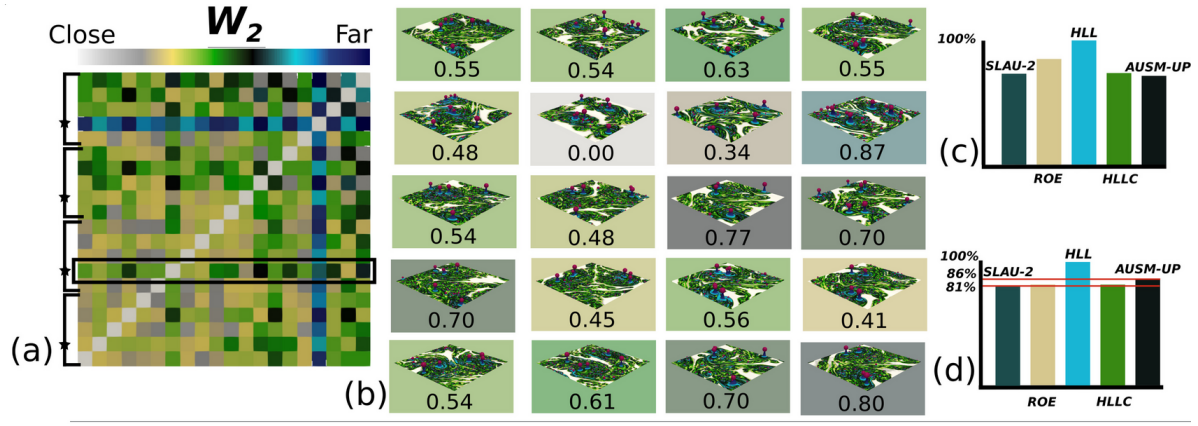


Figure 5.10: The W_2 metric for isolating the HLL solver. (a) Distance matrix between the TENO 7^{th} order with the HLL for 20 configurations at t_2 at 512×512.(b) Matrix view of the distance between the TENO 7_{th} order with the HLL and the other configurations. (c) Histogram of the percentage average of the sum distance matrix. (d) Histogram of the percentage of the sum distance for all variations.

5.5.2 Outlier Distance Profile Study

To verify the HLL isolation stated in hypothesis H3 (subsection 5.2.3) on our ensemble dataset, we implemented our protocol 2 based on the Wasserstein distance and the L_2-norm (subsection 5.3.5). For this study, we apply protocol 2 where time and resolution are fixed. The parameters that vary are the schemes ($\times 2$), the orders ($\times 2$), and the solvers ($\times 5$), generating 20 cases according to Table 5.1. All the distances have been computed according to the protocol of the outlier distance profile. These distances are represented by a global distance matrix where a line represents the 20 configurations (Wasserstein Figure 5.10a and L_2-norm Figure 5.11a) compared to the HLL solver, chosen as the reference. The matrix view of Figure 5.10b and Figure 5.11b shows the KHI terrains and the distances of all configurations to the HLL solver.

The study has been done for all time steps and all resolutions, generating nine 20×20 distance matrices for each distance. The histograms of Figure 5.10c and Figure 5.11c show the average of these nine distance matrices for the Wasserstein distance and the L_2-norm, expressed in terms of percentage according to the distance of HLL to the other solvers (HLL being the reference at 100%). The histograms of Figure 5.10d and Figure 5.11d show the percentage of the sum distance for all variations. In this case, the percentage difference in distances to HLL is about 18% for the Wasserstein and 13% for the L_2. These large percentages confirm that HLL is a solver that behaves differently from others.

As it does not take into account contact discontinuities, the interfaces between the vortices are much less defined than with the other solvers, resulting in a different number of vortices. From a physical point of view, this result confirms the isolation of HLL in all cases. From a topological point of view, it shows that the Wasserstein distance is the best at differentiating the HLL solver from the others (the distance gap is always bigger than the L_2). For this large study of 18 distance matrices 20×20, hypothesis H3 is verified.

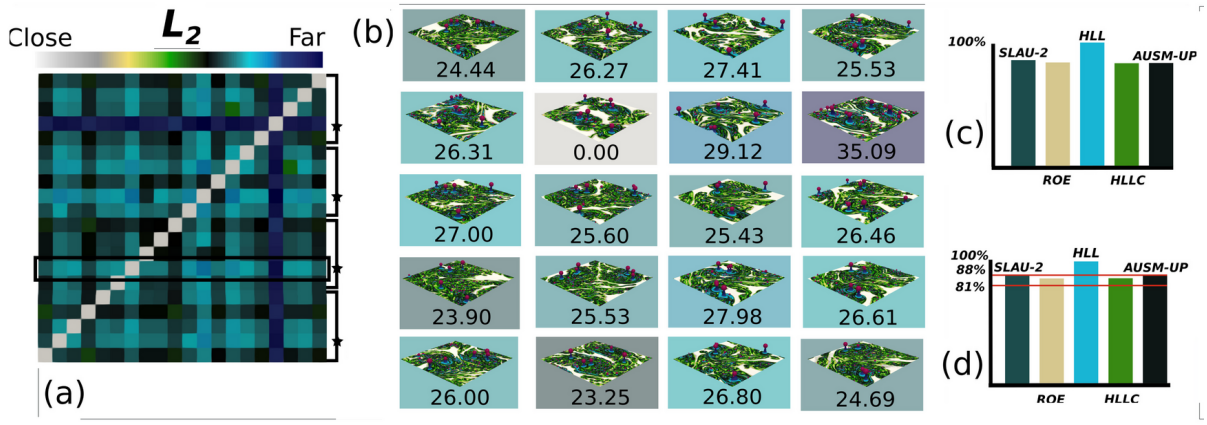


Figure 5.11: The standard L_2-metric for isolating the HLL solver. (a) Distance matrix between the TENO 7^{th} order with the HLL for 20 configurations at t_2 at 512×512.(b) Matrix view of the distance between the TENO 7_{th} order with the HLL and the other configurations. (c) Histogram of the percentage average of the sum distance matrix. (d) Histogram of the percentage of the sum distance for all variations.

5.5.3 Unsupervised Classification Study

To improve our understanding of the behavior of the solvers in our simulation code, we implemented protocol 3 on unsupervised classification to verify hypotheses H4 and H5 (subsection 5.2.3). The goal is to identify the separation of FDS type solvers from the FTS type solvers (subsection 5.1.1). We are interested in the low Mach reconstructions.

The challenge comes from the fact that small vortices are reconstructed on only a few cells. So, we implemented protocol 3 with the distances and clustering method leading to 5 simulation configurations (Table 5.1) with variable solvers. To focus on the small vortices we used a threshold of 0.38 persistence for the topological methods. On the KHI ensemble, we generated 36 clusters from the threshold persistence diagrams and obtained the Rand Index for all of them. On Figure 5.12, point clouds are generated for the time steps t_0, t_1, t_2 with a first-order scheme at 256×256. The point cloud is a representation of the five scalar fields in the distance space colored according to the clusters obtained. The Rand Index is computed with the five configurations S_1 (SLAU2), S_2 (HLL), S_3 (AUSM$^+$-UP), S_4 (Roe), S_5 (HLLC). The W_2* and L_2 are shown on Figure 5.12.

Histogram Figure 5.12 h shows the average Rand Index for the three methods with a value of 0.63 for L_2, 0.66 for W_2, and 0.71 for W_2*. There is very little difference between the topological and geometric results, and each of the methods struggles to get the right cluster. Hypotheses H4 and H5 are not verified for high orders. However, to highlight the differences between solvers, it is necessary to use a reference reconstruction that barely captures small-scale turbulence due to order dissipation (subsection 5.1.1). Thus, we applied protocol 3 (section 5.4) on a more restricted dataset at order 1. Histogram Figure 5.12g shows the average Rand Index at order 1 with the three methods leading to 0.63 for L_2, 0.71 for W_2, and 0.78 for W_2*. In this case, we notice that for any reconstruction, the topological methods obtain better clustering. Moreover, the study with order 1 shows that the W_2* method enhances solver isolation. With this high

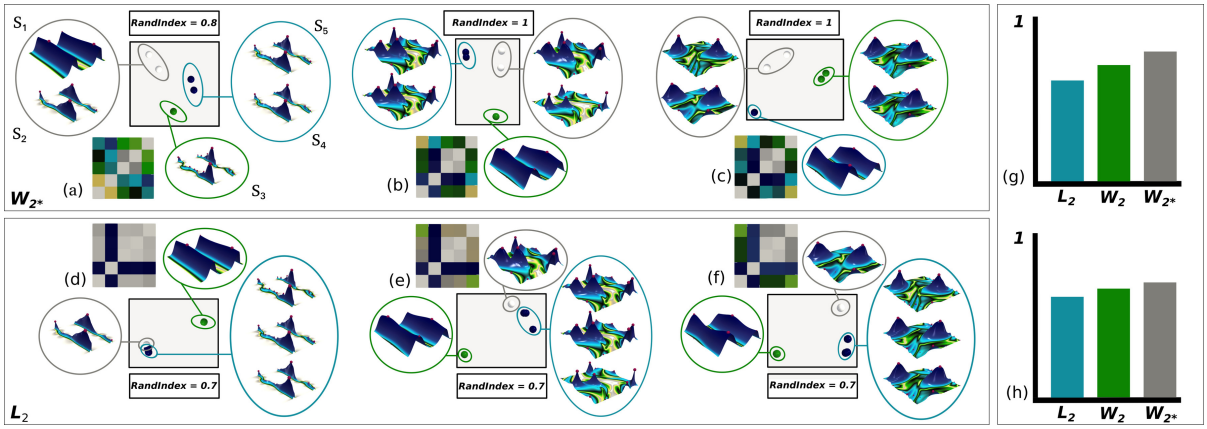


Figure 5.12: Comparison between the clustering on the Wasserstein metric space (top frame) and a clustering based on the traditional L_2 norm (bottom frame) for distinguishing FDS solvers from FTS solvers. (g,h) Average Rand Index for all variations for the high orders (bottom) and the first order (top).

score of the Rand Index, hypotheses H4 and H5 are verified with the first order.

5.6 Contributions of TDA to code developpment

5.6.1 Unanticipated Insights

During the analysis of the persistence curves generated by our protocol 1, we found significant differences in the topology of the enstrophy between the orders for the WENO-Z. By increasing the order, we increase the accuracy of our calculation, which generates more structures into the turbulent flow. On the other hand, there is no difference between the orders obtained with the TENO. This means that other ingredients in the TENO reconstruction play an important role in the computation of the turbulence, such as the separation of the scales. In addition, the persistence curves also allowed us to observe that the WENO-Z schemes produce more numerical errors than the TENO. As presented in subsection 5.5.3, H4 and H5 hypotheses have not been verified for high orders. This means that the topological analysis does not capture the differences between the solvers. This may be due to the reconstructions, which are accurate enough to calculate all velocities in the Kelvin-Helmholtz instability.

5.6.2 Limitations

As discussed in subsection 5.5.2, in comparison to the L_2 norm, the Wasserstein distance improves the separation of the HLL solver, but only by 5% (distance difference percentage). While this improvement may seem marginal, we would like to stress its significance given such challenging data, in particular with regard to the traditional approach based on kinetic energy, shown in Figure 5.4, where the five solvers can hardly be distinguished from each other.

Similarly, we can see that the Rand Index score for the three clustering methods detailed in subsection 5.5.3 are quite close to each other, as illustrated on Figure 5.12h. These close scores are due to the interpolation schemes (subsection 5.1.1), which cover up the differences between the different solvers. In other words, the variations in vortex distributions induced by the choice of solver are too subtle, given the importance of the interpolation order on the outcome. As shown in Figure 5.12 (top), we were still able to overcome this limitation by considering a reconstruction that is not dedicated to turbulence, i.e., an upwind scheme of order 1. This enabled us to exaggerate the impact of the solvers, thereby allowing us to validate hypotheses H4 and H5 as reported in subsection 5.5.3.

5.6.3 Towards New Applications

The results obtained in [Nauleau et al., 2022b] demonstrate the viability of topological methods for representing and comparing Kelvin-Helmholtz instabilities. The key advantage of these topological protocols is that numerical method comparisons are based on physical differences rather than on unreliable, low-level, pointwise measures. They can be used as guidelines for analyzing any complex features in ensemble data with the Topology ToolKit and ParaView as illustrated on [Tierny et al., 2023]. Even though improvement could have been made on the topological distances, we rather focus on other application domains where chaotic data is ubiquitous and where TDA can help to better understand the physics. The next chapter illustrates these projects.

CHAPTER 6

Topological Data Analysis for Chaotic Data

Advances in numerical simulation and hybrid parallelism for code development enhance the level of description of complex physical phenomena. Regarding computational capacity, advances in high-performance computing have made it possible to achieve massive computations at an accessible cost of time and energy consumption. Heterogeneous architectures based on processors (CPUs) and graphics cards (GPUs) also make it possible to increase the number of operations per second.

For example, in the June 2025 ranking of the TOP 500 most powerful supercomputers in the world [TOP500.org, 2025], El Capitan at DOE/NNSA's Lawrence Livermore National Laboratory (LLNL) is in first place with nearly 1,700 PFLOP/s of Linpack performance [Dongarra, 1989] using AMD 4^{th} generation EPYC processors. At CEA, the EXA1-HE computer with 500 hybrid nodes (Nvidia Grace Hopper Superchip processors) achieves Linpack performance of around 90 PFLOP/s, making it one of the 25 most powerful computers in the world. Moreover, with the utilization of advanced sensors and imaging capabilities in large facilities such as the Laser MegaJoule (LMJ) at CEA or the National Ignition Facility (NIF) at LLNL, complex experimental data can be acquired.

Over the last decade, therefore, my research activities have been driven by the need to develop scientific visualization strategies for various physicists [Vivodtzev et al., 2023]. When dealing with large and complex data obtained with the capacities described above, Topological Data Analysis becomes particularly efficient using the appropriate abstraction introduced in chapter 3.

Thus, chaotic turbulence has been analyzed with 2D vortex segmentation [Bridel-Bertomeu et al., 2019] with an extension to 3D viscous flow with an in-situ approach [Nauleau et al., 2022a]. To go further into the understanding of local turbulent vortices, a classification based on

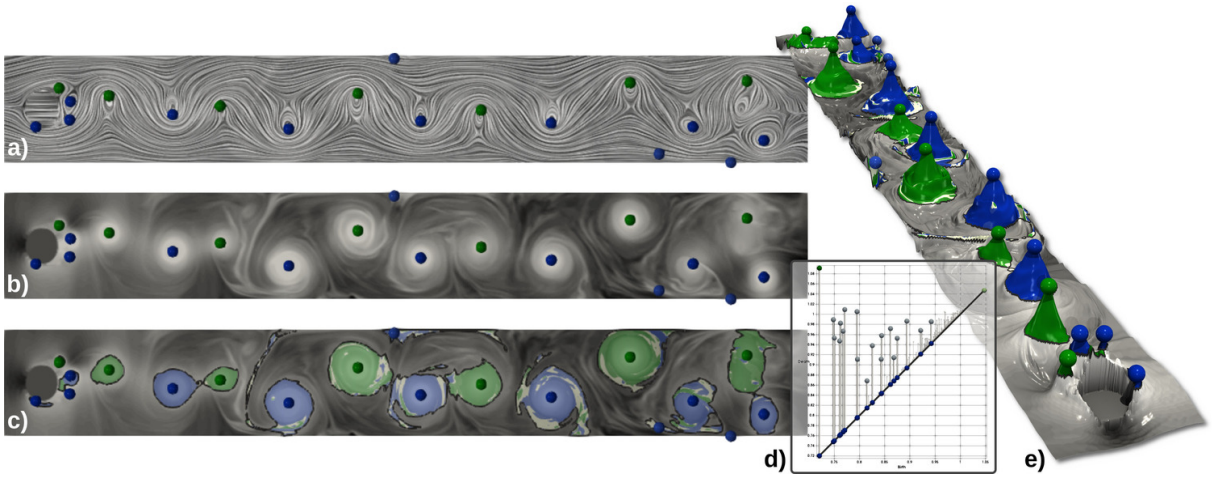


Figure 6.1: Topological analysis of a single time step. (a) Line Integral Convolution of the input vector field. (b) Flow density. (c) Merge tree based vortex segmentation. (d) Persistence diagram. (e) Terrain view with rotation colors.

CFD properties has been used in a topological segmentation [Vivodtzev et al., 2024]. Feature detection methods have also been proposed to better characterize hydrodynamic instabilities in x-ray images taken on the NIF [Vivodtzev et al., 2023].

6.1 2D Vortex Segmentation in Turbulent Flows

Many flow simulations conducted at CEA involve turbulent flow. To improve the understanding of the results and avoid costly traditional analysis methods, we introduced several TDA pipelines on 2D turbulent flows described in [Bridel-Bertomeu et al., 2019].

6.1.1 Data Description

The use case considered is a von Kármán street, as illustrated in Figure 6.1. The fluid enters the domain behind a cylindrical obstacle, on the left, at Mach 0.475 and with a Reynolds number of 10^5, and can only exit at the other end of the domain. The rest of the boundaries are set as periodic. The computation is run up to $t_f = 8.5 \times 10^{-3}$ seconds in physical time by 15, 625 steps. A total of 625 snapshots have been taken in order to realize the present analysis, separated from each other by a constant time step of 1.35×10^{-5} seconds, *i.e.* at a frequency of about 75 kHz. Each snapshot stores ρ the density, ρV with V the velocity vector and ρE with E the energy in double precision on a 5000×400 2D grid, yielding a total amount of data of 14 GB.

In terms of physics, it is of note that the mesh is not strictly speaking adapted to perform a complete Direct Numerical Simulation [Moin and Mahesh, 1998; Pope, 2001] down to the smallest Kolmogorov scales [Pope, 2001]. Nonetheless, it is fine enough to capture the largest eddies of the turbulent flow developing in the wake of the cylindrical body (see Figure 6.1b for

instance) and provides accurate enough data for the TDA to be applied thereupon.

6.1.2 Instantaneous Analysis

One result of the topological analysis for a single time step is shown in Figure 6.1, here close to the end of the run at $t \simeq t_f$. The comparison with the result of a pass of Line Integral Convolution [Cabral and Leedom, 1993] shows the richness of the TDA. It is capable of pinpointing the vortices by finding the regions of minimum density (spheres), which is in agreement with the LIC vector field treatment. The merge tree-based segmentation of the density field (Figure 6.1c) also allows for the immediate visualization of each vortex's region of influence and its rotation direction - green for clockwise, blue for counter-clockwise. By classifying the vortices using their persistence, as shown in Figure 6.1d, it is also possible to immediately distinguish between physically relevant structures and numerical noise.

6.1.3 TDA Abstraction

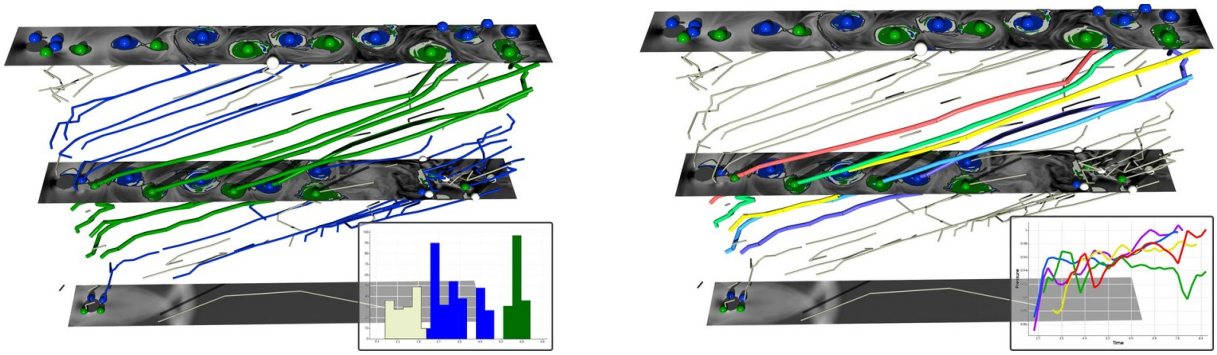
To adjust the parameters of our analysis for data scientists, we first considered a single time step as shown in Figure 6.1. In particular, vortices were identified as local minima of the density, with their region of influence given by a merge tree-based segmentation [Gueunet et al., 2019b]. The rotation direction of each vortex is estimated by the sign of the orthogonal component of the curl of the velocity. To discard noise, vortex importance within a single time step was assessed by the topological persistence [Edelsbrunner and Harer, 2009] of the corresponding minimum of the density.

To estimate an appropriate persistence threshold, we considered the persistence diagram drawn on Figure 6.1d which represents each minimum, denoting a vortex, as a vertical bar, whose height denotes the density amplitude of the vortex. Bars near the diagonal, corresponding to noisy structures, can easily be isolated (persistence below 10% of the global value range) and the data can be simplified [Tierny and Pascucci, 2012] to account for noise removal. Next, minima of the density are tracked through time by estimating an optimal assignment based on the Wasserstein metric between consecutive time steps as introduced in subsection 5.3.5.

6.1.4 Interpretation

As mentioned in the previous paragraph, topological assessment of a single instantaneous snapshot yields paramount information, such as a sorting criterion between noise and relevant vortices. The extraction and temporal tracking of the vortices thanks to TDA took 215 seconds overall (Xeon CPU, 2.6 GHz, 2x6 cores). The vortex tracking provides a valuable feature representation for further analysis and interpretation.

To take into account the time-varying aspect of the data, a topology-based tracking and analysis of vortices are done as shown on Figure 6.2. The flow density is used as a scalar field



(a) (b)

Figure 6.2: (a) Topology-based tracking and analysis of vortices showing 3 modes in terms of duration. (b) The 4 longest trajectories shown in colors are used to analyze the time evolution of flow density.

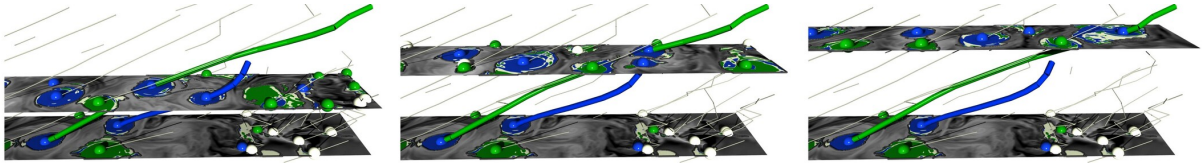


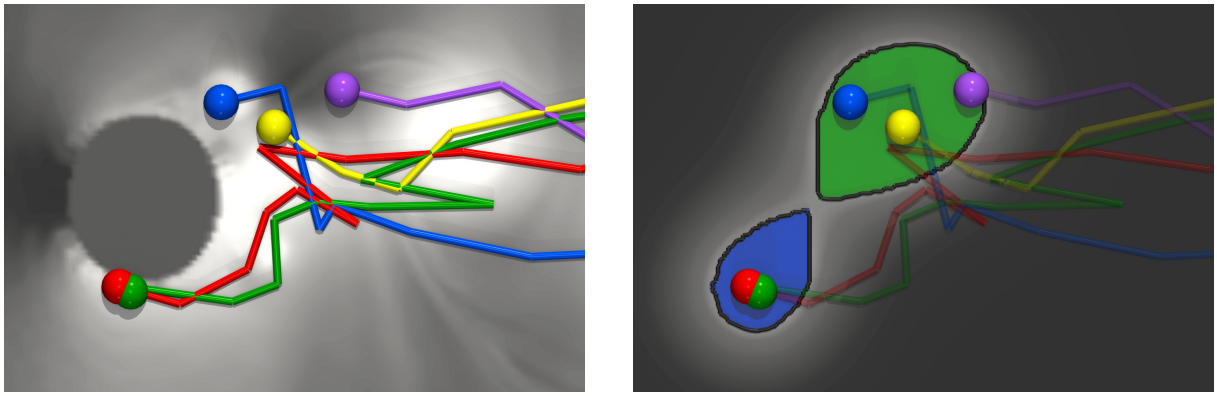
Figure 6.3: The two trajectories which maximize lateral movement correspond to 2 clockwise vortices which merge together.

and the time as height, highlighting the temporal evolution of the vortices moving throughout the domain. The TDA, realized at each time step, performs the same merge tree-based segmentation of the density field.

The vortex trajectories on Figure 6.2a exhibit 3 modes in terms of duration, corresponding to the 3 colors in the inset histogram: noise (white), intermediate length (blue), and full length (green). The 4 longest trajectories shown in color on Figure 6.2b are used in the remainder to estimate further statistics. For instance, we use the startup time of these trajectories to estimate the frequency of the vortex shedding ($1,500$ Hz), which matches the theoretical expectations [Blevins, 1977] without using Fourier decomposition and validates the relevance of our analysis pipeline.

The time evolution of the Y coordinate of a vortex center is used to denote its lateral movement, which assesses how much a vortex trajectory deviates from a straight line. The 2 trajectories which maximize their lateral movement are shown in Figure 6.3. These correspond to two interacting vortices rotating in the same direction, entering a coupled spiral movement and eventually merging together. This indicates that this kind of vortex merging can be systematically tracked thanks to TDA.

This analysis also enables investigation of the vortex start points, also known as *wavemakers* and illustrated on Figure 6.4. In hydrodynamics, determining wavemaker locations often requires stability analyses [Bridel Bertomeu, 2016; Chandrasekhar, 2013] that are tricky to develop and manipulate. This information can be, however, of great importance, for instance for the identification and resolution of sources of vibrations in an engine. Here, by considering a merge



(a) (b)

Figure 6.4: (a) Zoom in the shear-layer region behind the obstacle. (b) Merge tree segmentation of a Gaussian density estimation showing green clockwise vortices generated in the low-pressure region whereas blue anti-clockwise vortices in the high pressure region.

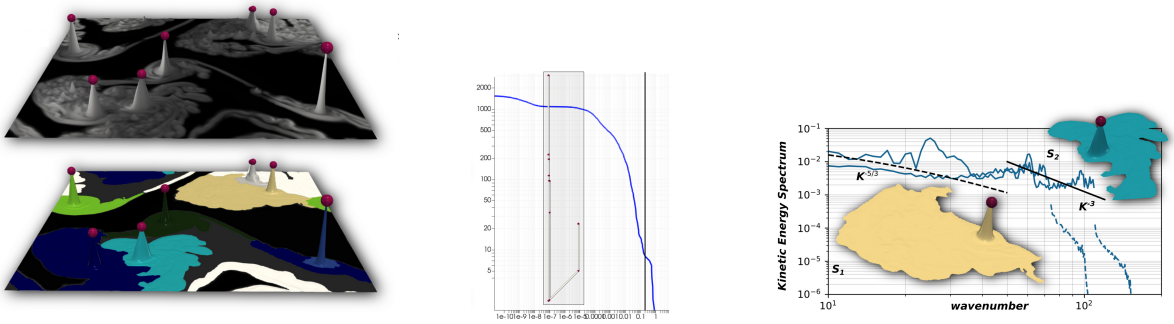
tree segmentation of a density estimation of the trajectory start points, locations for wavemakers can be estimated.

In [Bridel-Bertomeu et al., 2019], we detailed the TDA pipeline proposed to domain experts to provide an analysis support for the detailed investigation of high-velocity compressible turbulent flows, which would be difficult to conduct with traditional methods. This framework, in particular, enabled us to estimate the frequency of vortex shedding, locate wavemakers, and extract vortex merging events.

6.2 In situ Segmentation of 3D Viscous Flows

In this separate project, we focus on viscous flows specifically in 3D. The primary objective of this study is to identify the areas of influence of the most important vortices in a context of vehicle design for atmospheric reentry. Numerical simulations require running massive 3D calculations generating a set of important and complex data that must be analyzed to understand the physical phenomena involved.

To address the significant volume of 3D data generated by this turbulence study, we employed in situ analysis methods. With a topological segmentation of vortices in turbulent data, we show in [Nauleau et al., 2022a] that the vortices describe a physical solution by looking at its energy spectrum. This study would have been more difficult to perform with traditional methods due to the complexity of the turbulent flows and the finer-grained mesh required.



(a) (b) (c)

Figure 6.5: Kelvin-Helmholtz instability. (a) Enstrophy scalar field with a segmentation visualization. (b) Persistence diagram and curve. (c) Energy spectrum of two vortices.

6.2.1 Data Description and in situ Approach

The simulation code [Bridel-Bertomeu, 2021] described in subsection 5.1.1 generated 3D data on a Taylor Green Vortex with initial conditions (see [San and Kara, 2015]) on a grid of 64x64x64 cells. To emulate turbulence in an infinite medium, all boundary conditions are set as periodic. The local enstrophy is used as a scalar field as described in Equation 5.3

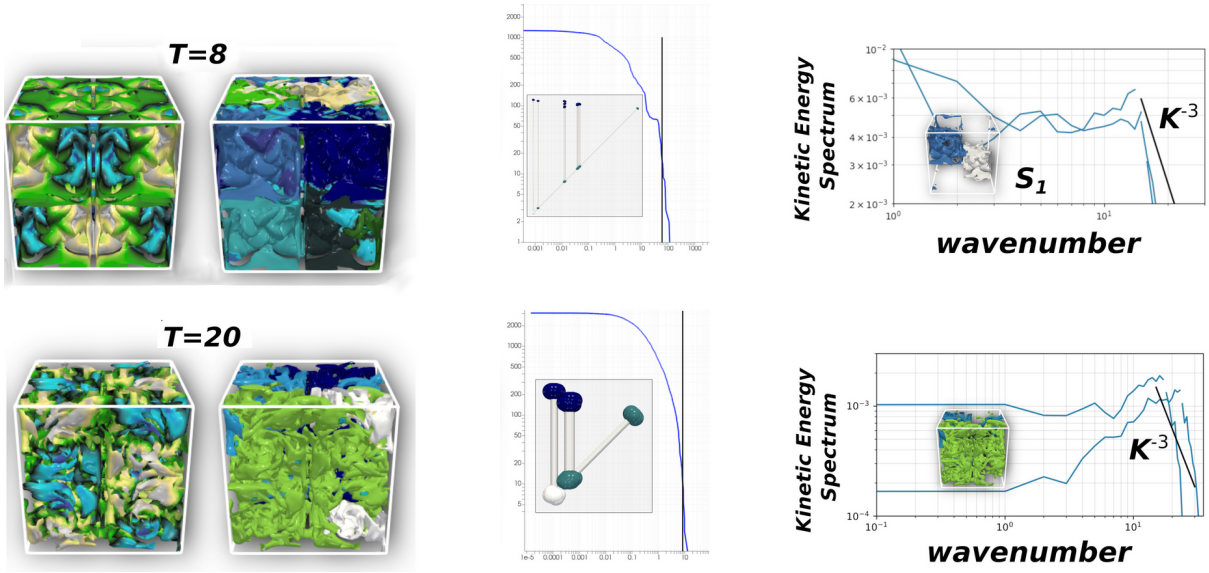
Running simulations at the exascale level means computing is getting cheaper while data transfer and storage is increasingly expensive. As a consequence, we replace the post-processing step – analysis and visualization of the simulation results in traditional scientific computing workflows – with in situ analysis. This approach minimizes the data transfer bottlenecks as all analyses are done in line with the simulation. To assist data scientists, the ParaView pipeline set up by a user and normally run on data saved on disk is transposed to a Python script generated by ParaView. An adaptor is implemented in the simulation code to map the simulation data structures to the Catalyst [Ayachit et al., 2015] data model. This script is finally given to the simulation code via the command line arguments to produce rendered images on the fly.

6.2.2 TDA for 3D Segmentation

The segmentation was first applied to 2D hydrodynamic turbulent flow simulations, where satisfactory results were obtained as shown on Figure 6.5. The segmentation was then applied to a 3D viscous flow simulation illustrated on Figure 6.6. Our method of separating different ensembles of connected vortices together gives accurate results and is easy to apply.

Our study relies on the enstrophy using Catalyst and TTK. From persistence curves (Figure 6.5b and Figure 6.6b), we set a threshold value on the enstrophy to keep the most important vortices. Then we use a Morse-Smale Complex [Tierny, 2018] to segment the enstrophy scalar field. Finally, we analyze the vortex ensemble with its energy spectrum [Kaneda et al., 2003] as exhibited in Figure 6.5c and Figure 6.6c.

Thanks to persistence curves, we easily identified the areas of influence of large vortices.



(a) (b) (c)

Figure 6.6: Taylor Green Vortex. (a) iso-contours of the enstrophy and segmentation. (b) Persistence diagram and curve. (c) Energy spectra.

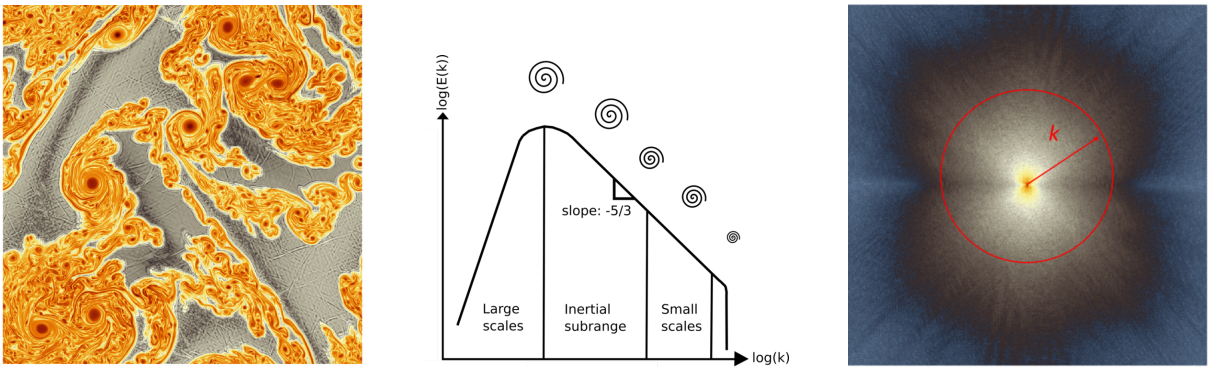
This allowed us to verify the segmentation on the enstrophy scalar field with energy spectra for each vortex ensemble. We notice that the energy transfer of vortex ensembles evolves in $K^{5/3}$ for 2D simulations and in K^{-3} for 3D simulations (see Figure 6.5c and Figure 6.6c).

This method that we introduced in [Nauleau et al., 2022a] is helping us verify that the turbulence reconstruction describes a physical solution during the simulation. This work using the energy spectrum of segmented vortices motivated a more precise analysis of local turbulent as described in the next section.

6.3 Local Turbulent Vortices Identification

The preliminary studies presented in section 6.2 showed that TDA could be correlated with traditional turbulence descriptors for high velocity compressible turbulent flow, such as the location of wavemakers or the frequency of vortex shedding. To go further in understanding the relevance of the features extracted with TDA with respect to traditional turbulence descriptors, we investigated Kolmogorov's work in [Vivodtzev et al., 2024].

In a typical turbulent flow as illustrated on Figure 6.7c, there is a wide range of vortex sizes that fluctuate at different frequencies. In a laminar flow, the velocity varies smoothly and predictably in space and time, whereas a turbulent flow exhibits a chaotic behavior. Turbulence is characterized by energy transfers between these small and large eddies. In the 1940s, Andrey Kolmogorov introduced several statistical properties for these flows for an intermediate range of eddy scales, called the inertial subrange illustrated on Figure 6.7c. The one used in this work is the energy spectrum of turbulence $E(k)$ related to the mean turbulence kinetic energy as a



(a) (b) (c)

Figure 6.7: Typical chaotic turbulent flow: (a) Enstrophy scalar field. (b) Kolmogorov energy cascade. (c) Shifted Fast Fourier Transform with radial averaging over wavenumber k.

function of wavenumber (a.k.a. the spatial frequency) k.

In the inertial subrange, the energy transfer from low to high wavenumber can be described [Obukhov, 1941] as the uniform form of E(k) $\sim k^{-5/3}$. In our work, the characterization of turbulence in the topological structures extracted by our pipeline is based on this concept. The kinetic energy spectrum is computed via a Fast Fourier transform and radial averaging [Navah et al., 2020] as illustrated on Figure 6.7b. The slope of the linear regression of the energy spectrum, in the inertial subrange, is used as an indicator of turbulence. Other indicators are also tested, such as the minimum and maximum of $E(k)$.

6.3.1 Case Study on a Turbulent Flow

The initialization of the turbulence was generated with two fluids of different densities and different velocities of opposite direction, creating a shear zone where the turbulence appears. This turbulence has been simulated with the simulation code [Bridel-Bertomeu, 2021] described in subsection 5.1.1. The boundaries are set to be periodic. A total of 421 snapshots were taken to capture the evolution of the turbulence on Cartesian grids of 1024×1024.

6.3.2 TDA Interpretation

A topological simplification is performed on the enstrophy scalar field based on a threshold of 0.3 on the normalized topological persistence. Then the vortices are extracted by collecting the basins of the ascending Morse-Smale complex (section 3.5) of the simplified enstrophy. The Morse-Smale complex provides an abstract representation of the gradient flow behavior of the enstrophy. It subdivides the scalar field into regions of uniform gradient flow, segmenting the flow such that every point in the same ascending manifold goes toward the same critical point. For the turbulence detailed in subsection 6.3.1 we threshold each ascending manifold above an arbitrarily low threshold of the original enstrophy (typically 10^{-6}), such that the boundaries

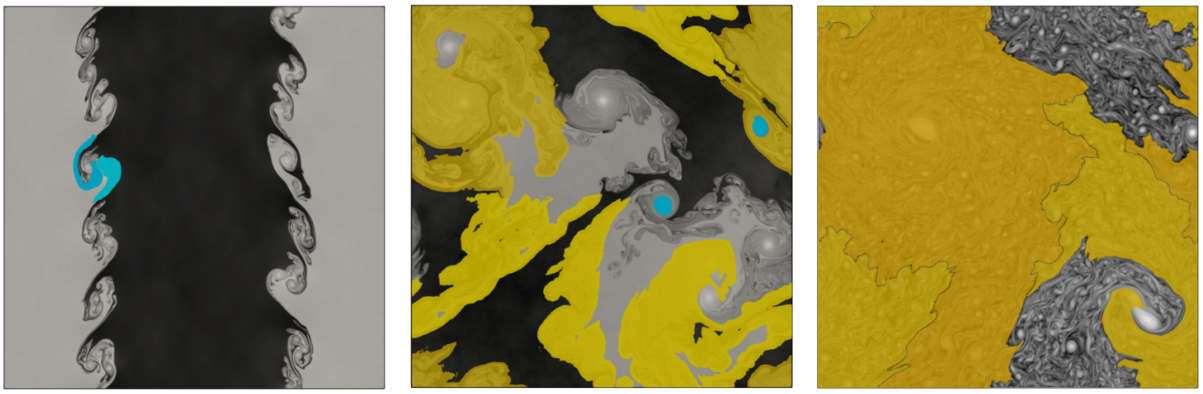


Figure 6.8: Ground-truth classification with laminar vortices in blue and turbulent vortices in orange annotated by CFD experts.

Time	Vortex	Slope	Emax	Emin	Type	diff k-E/3
20	6	-1,36	1,35E-07	1,03E-07	laminar	19%
	14	-1,45	1,29E-07	6,34E-08	laminar	13%
120	3	-1,18	1,79E-07	7,64E-08	laminar	29%
	7	-1,41	6,86E-08	3,37E-08	laminar	16%
	5	-1,51	3,93E-06	1,31E-06	turbulent	9%
	1	-1,79	3,08E-06	7,44E-07	turbulent	7%
310	9	-1,63	1,71E-06	6,03E-07	turbulent	2%
	1	-1,38	3,10E-06	1,07E-06	turbulent	3%

Table 6.1: Statistical vortex indicators computed on segmented topological features at different time steps.

of the extracted regions match the lowest level sets of enstrophy. Then a classification of the segmented vortices based on the Morse-Smale complex is performed by a CFD Expert as shown in Figure 6.8, which provides a ground truth description of the flow with laminar vortices in blue and turbulent vortices in orange.

The kinetic energy spectrum was calculated for each segmented vortex using a shifted Fast Fourier Transform with a radial averaging over the wavenumbers. Then several indicators are evaluated, such as the slope of the linear regression on the inertial subrange (wavenumbers between 120 and 500 in this case study) or the minimum and maximum of the kinetic energy. Table 6.1 summarizes these indicators for the vortices segmented in Figure 6.8. The slopes $E_{max}(k)$ and $E_{min}(k)$ are computed. Vortices with a slope close to $-5/3$ are successfully identified as turbulent, while others are identified as laminar. It successfully shows that TDA segmented vortices identified as turbulent by CFD experts respect the property E(k) $\sim k^{-5/3}$.

The analysis summarized on Figure 6.9 shows how TDA could be combined with flow frequency analysis for the detection of vortices which have reached a turbulent state. It demonstrates the relevance of our new indicators for distinguishing vortices which have reached a turbulent state (vortices 1 and 5) from those which have not yet (vortices 3 and 7).

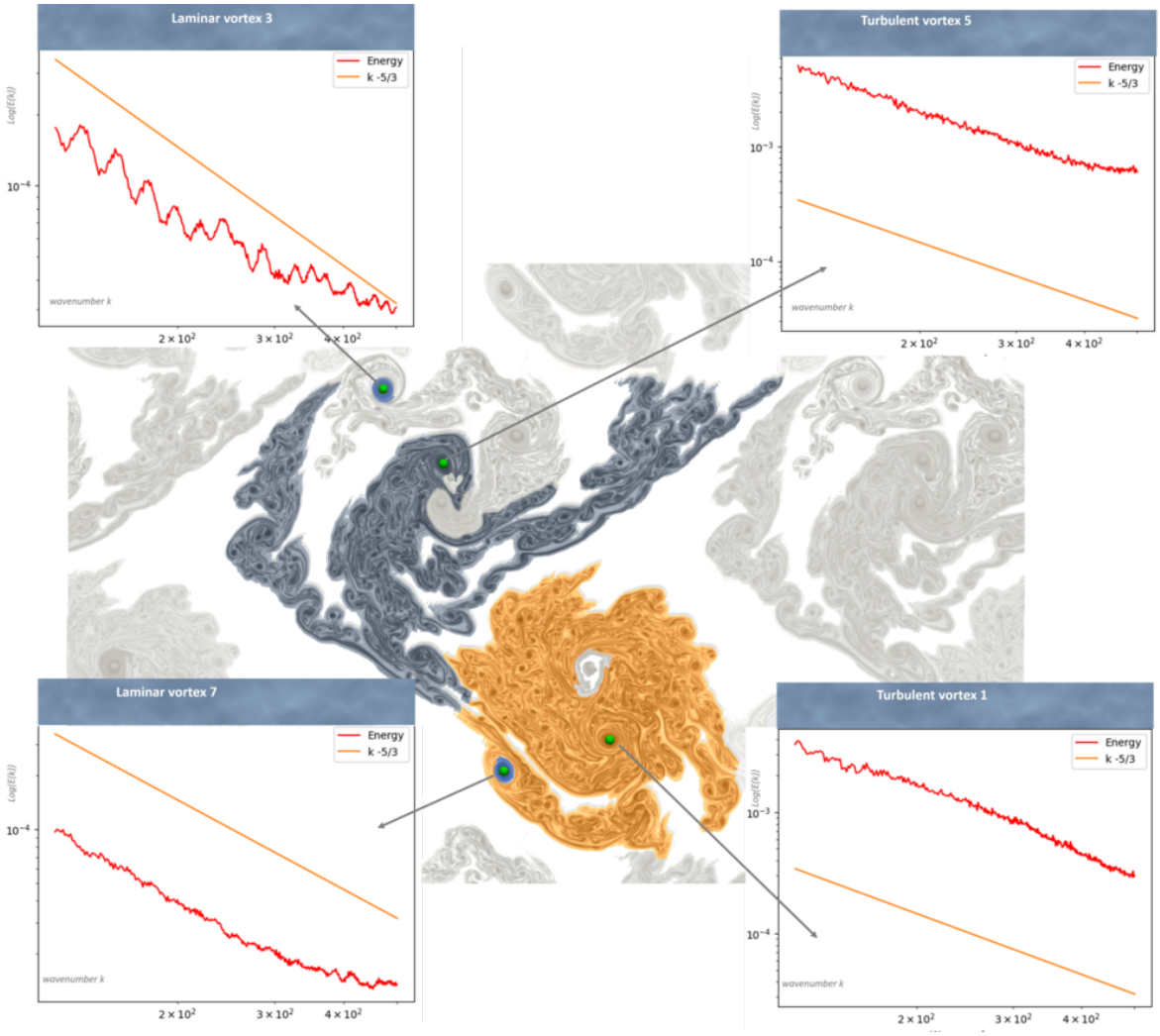
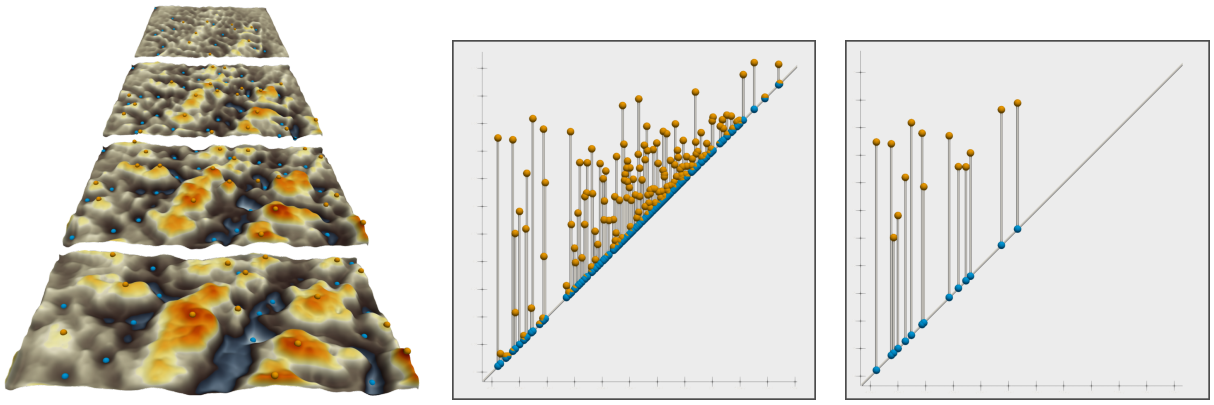


Figure 6.9: Classification of segmented vortices (laminar and turbulent) with the turbulence indicator. Thresholded Morse-Smale segmentation and kinetic energy spectrum of the vortices.



(a) (b) (c)

Figure 6.10: (a) Terrain views of the N180213-001 x-ray dataset, at 4 time steps with the most critical point of x-ray intensity. (b) Original persistence diagrams on the x-ray intensity. (c) Simplified persistence diagram.

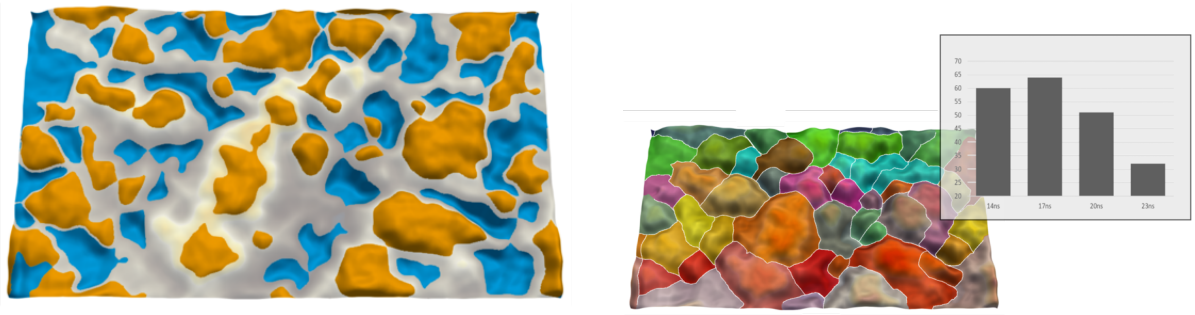
6.4 Feature Detection of Hydrodynamics Instabilities

Hydrodynamic instabilities are another type of chaotic data that the scientists need to overcome. In our work described in [Vivodtzev et al., 2023], we propose a TDA for the ablative Rayleigh-Taylor instabilities arising during Inertial Confinement Fusion (ICF). These particular instabilities during the convergence and implosion phases of ICF capsules [Casner, 2021] are one of the major drawbacks [McCall, 2023] on the path to the higher energy gains [Abu-Shawareb et al., 2022] needed for future inertial fusion energy power plants.

Dedicated Discovery Science campaigns [Casner et al., 2012] have therefore been conducted at the world's largest laser facility, the National Ignition Facility [Spaeth et al., 2016], in order to better understand the growth [Casner et al., 2015] and possibly control mechanisms of these detrimental instabilities. In particular, when starting from imprinted perturbations imposed by the laser focal spots [Ceurvorst et al., 2020], the ablative Rayleigh-Taylor instability gives birth at its highly nonlinear stage to complex 3D patterns. Typical data analysis methods based on threshold and watershed segmentation [Sadot et al., 2005] have been used to detect the peak-to-valley differences and the area of the bubbles. Such methods rely heavily on parameters defined by the observers. Therefore, we explore more robust methods based on TDA.

6.4.1 Data Description of Radiography

We explore several datasets such as the NI180212 or NI180213 through different visualization pipelines in Paraview [Ahrens et al., 2005b]. The topological analysis was done thanks to the Topology Toolkit [Tierny et al., 2017a]. First, we iterate on the x-ray images to better understand the correlation between the pixel intensity and the bubble structure created during the development of the Rayleigh-Taylor instability. We then construct a representation of the images as a discrete scalar field. The structured pixel image is triangulated to create a triangular



(a) (b)

Figure 6.11: (a) Contour tree segmentation of extrema area. (b) Ascending Morse-Smale complex segmentation with the histogram of the distribution of the number of bubbles into the hydrodynamic instability.

mesh defining a manifold surface. The cell-centered pixel values are interpolated on the mesh vertices. The size of experimental images in this study is 900×500. From that we use algorithms from the TDA to extract some critical points of interest as shown in Figure 6.10a by keeping all extrema pairs above a persistence threshold of 0.2. The original persistence diagram shown on Figure 6.10b is simplified by removing all pairs near the diagonal as shown on Figure 6.10c.

6.4.2 Analysis

Several topological representations have been evaluated in order to segment the features of the x-ray images corresponding to the bubble. First the discrete scalar field representing the pixel values is simplified according to a persistence threshold to preserve the main structures and remove small variations around bubbles. Then a contour tree is built on the scalar field and only the leaves of the trees are kept to segment the valleys (blue regions of Figure 6.11a) and the peaks (orange regions of Figure 6.11a) of the corresponding terrain.

We also evaluate a segmentation based on a Morse-Smale complex as shown in Figure 6.11b. We segment the x-ray by the 1-separatrices of the ascending Morse-Smale complex which leads to a very promising segmentation. To go further we compare the number of features segmented at each iteration and compare them with studies such as [Mailliet, 2018]. These topological structures successfully capture the expected decreasing number of bubbles of the instability over time as shown on the histogram Figure 6.11b.

In [Vivodtzev et al., 2024], we demonstrated that TDA of 3D Rayleigh-Taylor experimental radiographs is a promising approach to automatically analyze complex patterns in ICF. We show that Contour trees and Morse-Smale complexes extracted from a simplified x-ray based on persistence give promising segmentations.

CHAPTER 7

Visualization Software

Simulation data can be large and complex, as illustrated in chapter 6. An HPC approach often needs to be employed in order to be able to overcome the complexity of the physical phenomena to simulate. In this HPC context, the time-consuming computation codes generating the data are run on a supercomputer remote from the standard graphics workstations on the operating site. This separation can introduce a level of complexity in the visualization pipeline.

Over the years, we have developed various strategies, detailed in this chapter, by evaluating existing remote visualization software [Nassiet and Vivodtzev, 2012], integrating rendering views into simulation platforms [Vivodtzev and Carrard, 2014], and adapting the visualization capabilities of *VisIt* and *ParaView* for high-performance computing environments [Vivodtzev and Bertron, 2014].

7.1 Remote Visualization

Remote visualization techniques provide solutions for generating graphical rendering on a server as close as possible to the data, and for displaying results on a remote client connected to the same network but with limited graphical performance. In the 2010s, as network capacities increased, we evaluated various remote visualization solutions. The core of the problem lies in balancing the need for efficient data analysis and visualization from massive simulation outputs with stringent security constraints.

7.1.1 Industrial Challenges in Scientific Visualization

Many research laboratories or industrial architects of complex systems heavily rely on extensive simulations covering various physics (aerodynamics, electromagnetism, thermomechanics ...). These simulations generate colossal amounts of diverse data, demanding robust and secure solutions for visualization. These institutions operate within a distributed and heterogeneous computing landscape, including petaflop-scale production machines and user workstations running both Linux and Windows.

A key constraint is the physical distance between the main computing centers and the dedicated networks linking them, which are characterized by limited bandwidth. In the domain of defense, these dedicated networks need to be secured and isolated from the Internet, acting as a constraint in the global architecture of visualization solutions. Furthermore, users, while working on local machines, need to interact with remote resources. Also, they describe a wide variety of physical phenomena, which translates to many different simulation codes, each potentially using different data formats and visualization needs. In large research laboratories, it is common to have more than a hundred simulation software solutions that the overall analysis system needs to handle.

7.1.2 Existing Remote Visualization Approaches

To address these challenges, we worked on various aspects of the scientific visualization software as described in [Nassiet and Vivodtzev, 2012]. The first approach is to use existing software with remote capabilities when available. In the 2010s, different software were evaluated depending on the user's need and the simulation code they were using. We considered solutions with visualization capabilities in computing libraries (e.g., *Matlab* from MathWorks, *IDL* from NV5 Geospatial Software), simulation software (*Abaqus/Viewer* from Dassault Systèmes, *LS-PrePost* from ANSYS), or scientific visualization tools (*VisIt*, *Paraview*, *Tecplot* from Tecplot Europe). As an example, *Abaqus/Viewer* utilizes a client/server model, connecting to the stripping machines via SSH. This method primarily transfers geometry, considering client memory size and enabling real-time animations. The integration involves specific data storage and careful version management due to format dependencies

VisIt was configured to operate in a client/server mode, where the rendering engine and data server run directly on the high-performance computing clusters (like the supercomputer TERA [CEA TERA]), and only the geometry or images are transferred to the client workstation. The client-side components, on local machines, include the VisIt Viewer, Graphical User Interface (GUI), Command Line Interface (CLI), and Java Clients. These clients establish a network connection, via SSH, to remote supercomputers, where the server-side components operate as illustrated in Figure 7.1. On the supercomputer, *VisIt* engines work in parallel, utilizing MPI for communication, to process and filter data accessed through data plugins. This distributed processing allows for efficient handling of large datasets that would otherwise overwhelm local desktop machines. The connection setup to remote supercomputers is designed to be user-friendly, managed directly from the VisIt interface, allowing users to specify host settings, installation paths, and usernames.

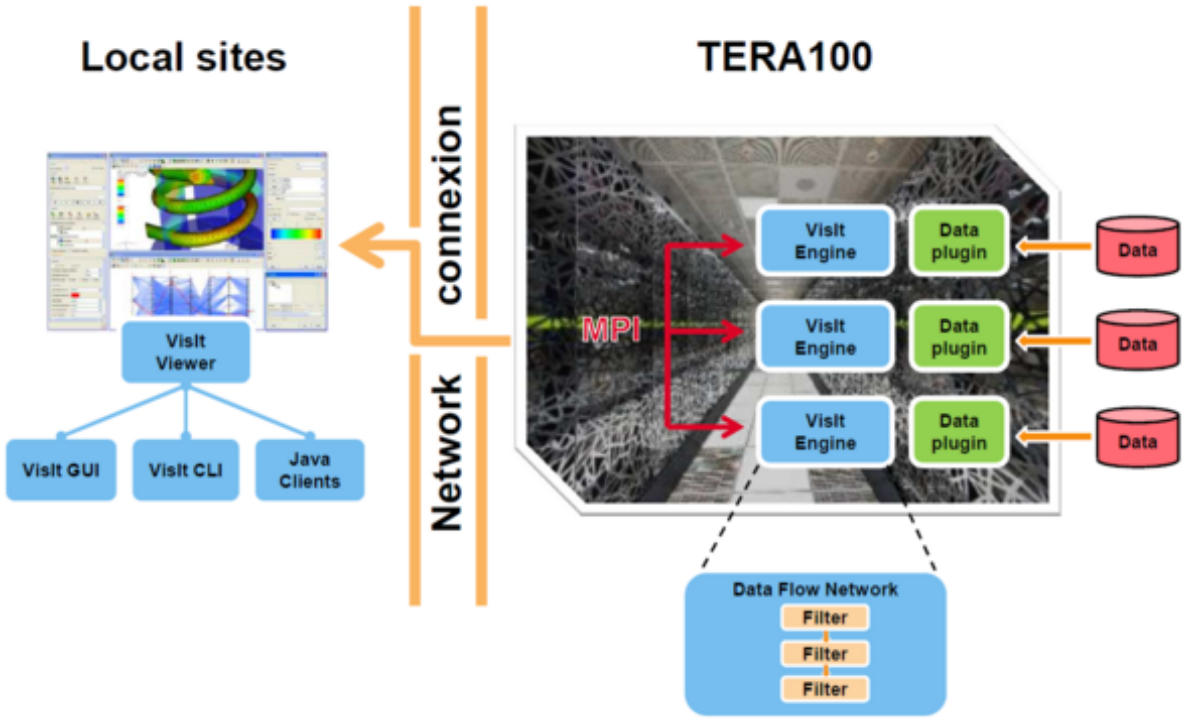


Figure 7.1: VisIt parallelized client-server architecture

Another approach that we used provided a secure connection to the stripping nodes via a web portal with the solution *VisuPortal* by Oxalya (now part of the company OVHcloud). This remote visualization tool facilitated temporary desktop deployment using *Java Network Launching Protocol* (.jnlp) files and optimized image transfer, interaction, and encryption (e.g., using a technology called HPRGS). This portal-based approach was designed for remote work, allowing users to interact with distant input data and calculation cases.

7.2 Integrated Visualization into Simulation Platforms

Even though existing software allows specific remote data exploration, it is not adapted to homogeneous solutions as explained in subsection 7.1.1. Thus, we started at the same time the development of an approach based on the integration of simulation tools with advanced visualization capabilities within an *Eclipse Rich Client Platform* (RCP) environment, requiring contributions to the *Visualization Toolkit* (VTK).

7.2.1 Requirements for a Unified Solution

To ease the overall process of the simulation cycle from the geometry to the computation, we work to create an adapted, easy-to-use, and homogeneous software environment. This work involves preparing and submitting calculations for multi-code and multi-machine contexts. Users need

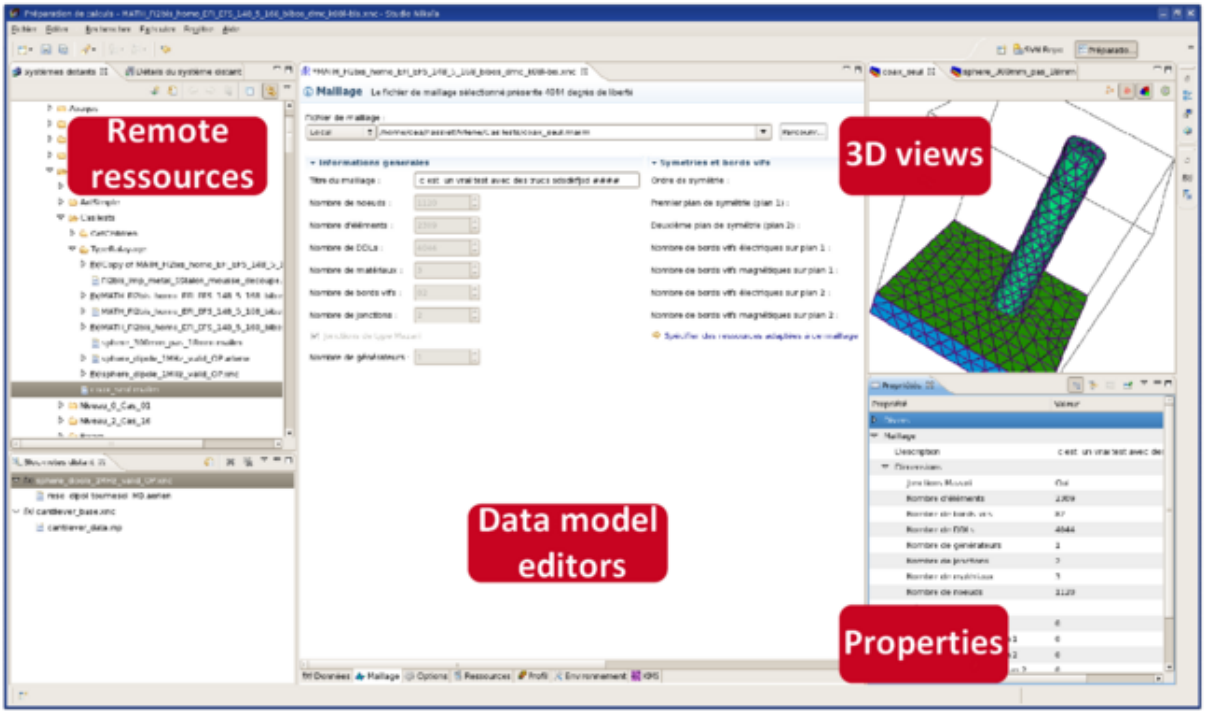


Figure 7.2: Integrated views of remote files, simulation data model editors, and 3D in an Eclipse RCP platform.

to easily input and validate data, manage local and remote files (meshes, material data ...), and submit calculations in interactive or batch modes, while dynamically allocating resources (time, memory, CPUs, or GPUs). Crucially, the solution must meet high security standards (supporting SSH connections, network identification/authentication protocols) and ensure portability across Linux and Windows client workstations. It must also be ergonomic, enabling users who are not batch specialists to seamlessly exploit distant resources from their local workstations.

7.2.2 Integrated Simulation Workbench

To address these complex requirements, we developed a simulation workbench. This software environment is an extensible *Eclipse RCP* (Rich Client Platform) application designed to separate generic services from specific functionalities. It offers generic services such as a remote system explorer, authentication, or 3D views as illustrated in Figure 7.2.

Based on the *Eclipse Modeling Framework* (EMF) conception model, the workbench is highly modular, which allows for flexible plugin integration. It also utilizes *Standard Widget Toolkit* (SWT) for graphical components and *JFace* for advanced GUI elements, ensuring a native look and feel across different operating systems. Furthermore, the *Eclipse* workbench defines mechanisms for views, perspectives, and editors, enabling a highly customizable user interface. The workbench provides extension points to integrate new computing machines, simulation codes, calculation case editors, and specialized views (e.g., for dataset properties, mesh properties, 3D visualization). This model-based approach eases the development of such

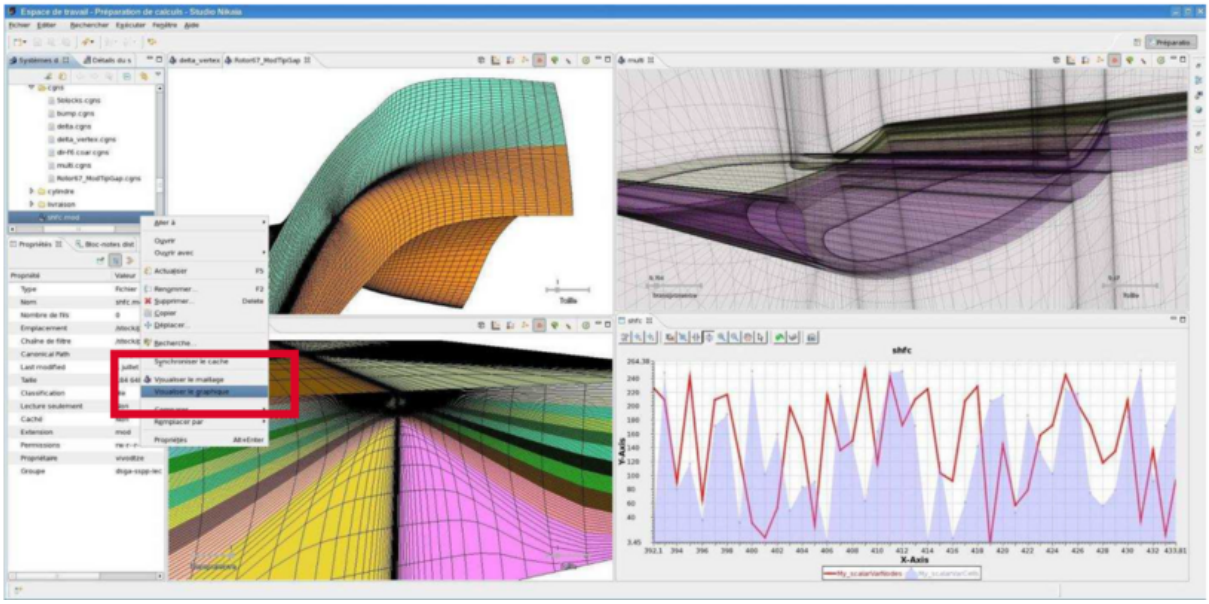


Figure 7.3: Example of embedded visualizations with the plugin integrated into the simulation workbench.

software by generating the components based on the data model [Nassiet et al., 2011], enabling dynamic configuration and manipulation of simulation parameters.

7.2.3 Core Features of the Integrated Visualization

We developed a specific visualization plugin integrated into the workbench described in subsection 7.2.2. It offers an integrated visualization view that supports specific data formats used by the simulation codes. Its key features include 3D interaction for data input, where users can interact with 3D views to perform data input, define properties, and assign materials directly on the mesh as illustrated in Figure 7.3.

This interaction facilitates tasks such as boundary extraction and initial condition setup, with picked information aiding in model editing. Before running computationally intensive simulations, this plugin allows users to preview meshes, ensuring correctness and identifying potential issues early in the workflow. The plugin also provides an interface for editing advanced properties associated with the simulation case. The integration within a remote explorer enables users to manage and visualize files located on remote computing machines.

7.2.4 Plugin Architecture and VTK Integration

The visualization plugin implementation heavily relies on VTK. It adopts a model-based approach where the 3D view is directly linked to a structured view of the input model (`IContentOutlinePage`). This means that any edition performed on the model directly modifies the input dataset, creating a cohesive and intuitive user experience. The plugin includes an implementa-

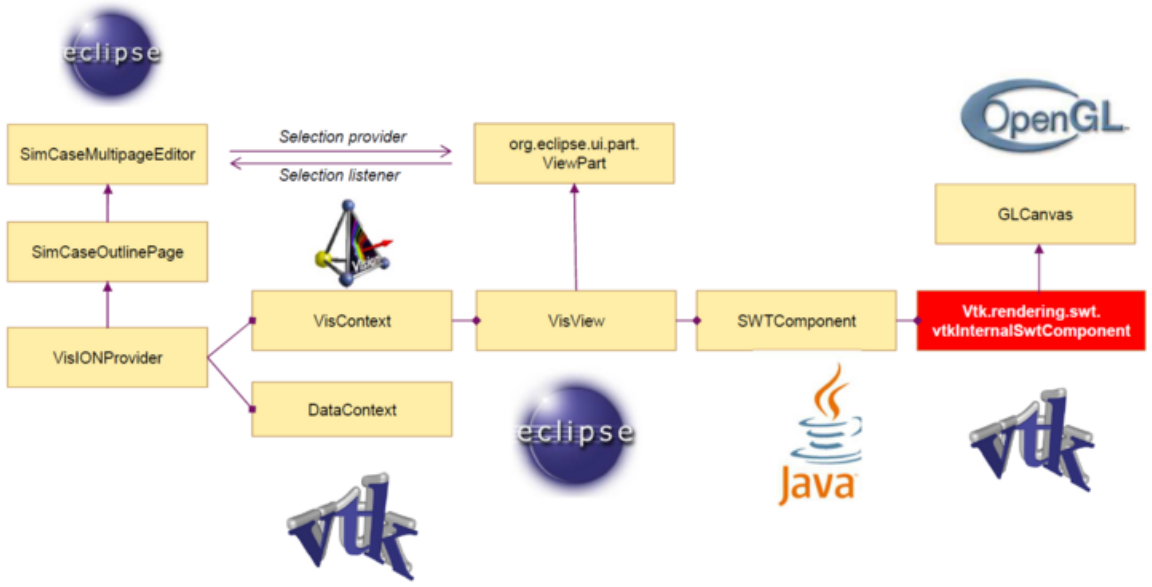


Figure 7.4: Integrated visualization plugin architecture relying on VTK and its JAVA interface.

tion of a multiblock dataset to handle datasets with multiple materials and blocks with enhanced picking and selection capabilities for identifying blocks, boundaries, and other geometric entities. It also leverages Eclipse extension points for integrating various readers, such as the CGNS reader (dedicated to fluid dynamics computations) implemented with *Java Native Interface* (JNI).

The integration of all the graphics layers (AWT, SWT, VTK, OpenGL) as illustrated in Figure 7.4 required evolution of the VTK library specifically for the JAVA interface. We collaborated with Kitware SAS to improve the rendering of a 3D view in the Eclipse environment through a SWT component [CEA and Kitware, 2012]. Indeed, the component, `Vtk.rendering.swt.vtkInternalSwtComponent`, acts as the bridge, allowing the VTK Canvas to be embedded within SWT components. It ensures that the 3D rendering of VTK can be displayed natively within the *Eclipse* RCP environment.

By integrating simulation workflows with embedded 3D visualization and analysis tools within a portable and secure *Eclipse* RCP environment, we provided scientists the tools to enhance productivity and streamline complex numerical simulations. However, for high-performance visualization requiring a high-performance computer, other solutions need to be considered.

7.3 Extending the Visualization Software *VisIt*

The software *VisIt* is an open-source scientific visualization and analysis tool developed primarily by Lawrence Livermore National Laboratory (LLNL) and for which we have been in collaboration [Childs et al., 2011]. It's especially efficient for enhancing data understanding, supporting extremely large datasets, and providing a robust, user-friendly product designed for

HPC environments. With data too large for desktop machines and the need for diverse analysis techniques, VisIt aimed to put visualization and analysis capabilities directly into the hands of end-users. It has since become a widely used tool around 2005.

We have been a contributor to plugin developments and a deployer of *VisIt*, focusing on adapting it for production-class visualization tools based on parallelized client-server architecture as described in subsection 7.1.2. Our work emphasizes leveraging *VisIt*'s parallel capabilities for exploring massive multi-block datasets and performing remote visualization [Childs et al., 2011] on the supercomputers like TERA [CEA TERA] and EXA [CEA EXA] as described in this section.

7.3.1 *VisIt* Architecture

VisIt is based on the third-party library VTK and employs a client/server architecture, with client-side programs (`gui`, `cli`, `viewer`) typically on local desktops for user interface and rendering, and server-side programs (`engine`, `mdserver`, `vcl`) on remote supercomputers for parallel data processing. This architecture allows for flexibility, including local execution, remote client access, multiple servers, batch processing, and in situ coupling with simulations. Its most frequent processing mode is pure parallelism, partitioning data across MPI tasks for native resolution processing. The user interface is built around five interoperable concepts: databases (readers), operators (data manipulation), plots (rendering), expressions (derived quantities), and queries (quantitative/debugging information). We added new functionalities as external C++ plugins, with *VisIt* providing code generation support.

7.3.2 Specific Database Readers

A key area of development that we investigated involves the creation of specialized database readers for *VisIt*. We adapted more than ten specific database readers to link the simulation code I/Os directly to *VisIt*. These C++ readers are deployed on both local and remote machines and are tightly integrated with the simulation data they are designed to handle, enabling on-demand computation of derived data.

Using *VisIt* pseudo code generation, an efficient `PopulateMetaData` function is crucial for quickly reading file metadata, emphasizing its importance in data processing. The *Single Time Single Domain* (STMD) format is prioritized to effectively leverage *VisIt*'s native parallelism. Some of the readers utilize libraries to handle various mesh and data types like HDF5, NetCDF, or SILO. This approach allows *VisIt* to read and process diverse and complex simulation outputs effectively, accommodating various mesh and material configurations.

7.3.3 Handling Ghost Zones

The accurate visualization of data at domain boundaries is vital, especially in parallel simulations where data is partitioned across multiple processors. We have developed methods for handling ghost zones, which are regions of overlap between computational blocks. This involves using knowledge from the simulation code to generate ghost cells according to the block connectivity in the input mesh. Specific functions like `avtCurvilinearDomainBoundaries` and `AddNeighbor` are used to manage the relationships between neighboring blocks, ensuring smooth visualization across domain boundaries.

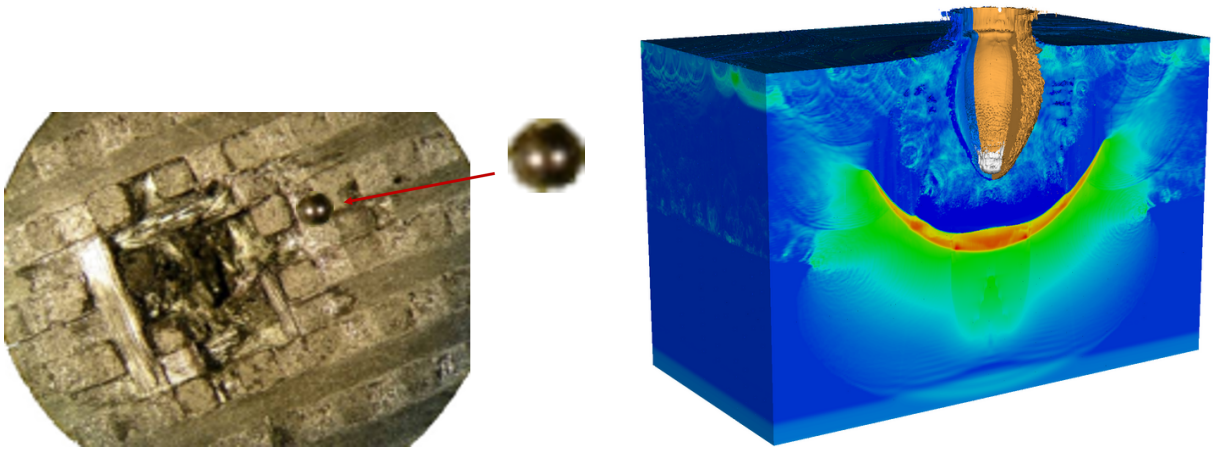
7.3.4 Material Interface Reconstruction

We have also made a specialized development into database readers related to the implementation of material interface reconstruction. Several methods of material interface reconstructions are available in *VisIt* [Meredith and Childs, 2010]. We adapted our reader to include `PopulateMetaData` to read only the first file and then gather all information from each file. Functions like `GetMesh(n)` and `ReadMetaData(n)` are used to retrieve mesh, material, scalar, and other metadata for specific domains. The process involves using `AuxiliaryData(n)` to categorize cells into pure, mixed (with material fractions), or empty. This allows for detailed analysis of material distribution within complex simulations.

7.4 Large Scale Visualization of Fast Dynamic Simulations

Debris shielding against hypervelocity impact (HVI) is a major concern for many applications such as spacecraft technology and high-power laser facilities. Indeed, meteoroids can impact satellites at several kilometers per second, possibly damaging or destroying some vital equipment. Similarly, the various instruments of the experiment chamber of large Laser facilities (e.g., the Laser MegaJoule (LMJ) at CEA or the National Ignition Facility (NIF) at LLNL) can be hit by a variety of shrapnel and debris originating from the target assembly. In a project started at CEA in 2011, we helped the scientists to better understand specific physical phenomena happening during debris shielding against hypervelocity impact. Benefiting from the extensions of *VisIt* presented in section 7.3, we focused on proposing scientific visualization tools to ease large-scale exploration of fast dynamic simulation as presented in [Vivodtzev and Bertron, 2014].

The scientists who led this study face several challenges in analyzing the large amount of data generated in a HPC context. We adapted *VisIt* to the exploration of such data using an approach generic enough to be extended to other fields. Based on the knowledge of the simulation code and the architecture where the computations are done, this solution provided a production analysis tool integrating the access to the remote data, the handling of the large amount of distributed solution files, the speed-up of a parallel visualization tool, and the easy off-screen movie generation.



(a) (b)

Figure 7.5: (*a*) Hypervelocity impact experiments on a composite material called EDM3 which is a grade of polycrystalline graphite. (*b*) Numerical simulations of crater formation with an impact of a steel projectile.

7.4.1 Visualization of Composite Materials under Impact

Due to their low density and high mechanical properties, composite materials are now more and more being used in the aerospace industry. Among other materials, carbon is of particular interest since it is widely used as an elementary component in composite materials. Our study relies on hypervelocity impact experiments as shown in Figure 7.5a and uses numerical simulations of crater formation shown in Figure 7.5b on a composite material called EDM3. This commercial grade of polycrystalline graphite is approximately 20% porous and macroscopically isotropic [Seisson et al., 2014].

The experiments consisted of impacts of a 0.5-mm-diameter steel sphere at around 4 km/s on a 30-mm-diameter and 15-mm-thickness graphite target. The projectile is launched by a two-stage light-gas gun. Post-mortem tomographies and Scanning Electron Microscope (SEM) observations helped us to adjust an isotropic damage model for porous and brittle materials in an Eulerian component of a hydrodynamic simulation code developed at CEA. This simulation code deals with multi-material flows in explicit mode on a Cartesian 2D or 3D grid using the BBC numerical scheme [PR and P., 1984].

For oblique impacts, 3D simulations are needed as shown in Figure 7.6 on a 15-degree impact at 4 different time steps. To obtain a good representation of the damaged area, the cell size used for computations was $12.5 \times 12.5 \times 12.5 \mu$m. It ensures both a good convergence level and a reasonable computation time but yields to 3D computations with billions of cells.

7.4.2 *VisIt* Configuration in a HPC Environment

The simulation code was run on the TERA supercomputer. In order to provide a user-friendly dedicated analysis tool, the scientific visualization software VisIt was adapted. The main

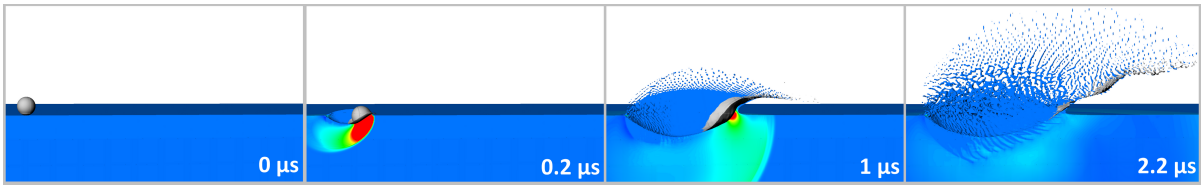


Figure 7.6: 15 degrees oblique impact of a steel projectile around 4 km/s in a graphite sample

challenges were to set up a remote and distributed environment and to be able to understand the simulation files. *VisIt* was deployed as described in subsection 7.1.2. This configuration allows the user to interactively choose the number of processors to use for computation on several partitions such as parallel, large (nodes with more than 512Go of memory), or hybrid (nodes with GPUs). Then, the user can browse and load their remote solution files with a specific database plugin without downloading them on the local site.

7.4.3 Database Plugin for Large Distributed Data

The Eulerian code gives post-processing data which contains the geometry description (a 3D fixed Cartesian grid), and for each cell, the volume of material present in it and the different thermodynamic quantities. Because of the Eulerian description, some cells are mixed (they contain different materials) in the interfaces between materials.

The environment introduced in this paper first extracts thousands of data files from archives. Each file of several Gigabytes corresponds to the computation of one processor for a sub-domain of the original grid. Using the development tools of *VisIt*, a specific database plugin has been implemented, taking advantage of the automatic code generation. The input file is interpreted as a *Single Timestep Multi Domain*. Using the metadata of only one data file in the `PopulateMeshMetaData` method of the reader, VisIt is initialized without any further computation and transfer operation.

Then, the parallel capability of *VisIt* is used through the loading of the sub-domain files distributed on each compute engine running on every node of the supercomputer. Figure 7.7 shows a parallel visualization of a 2-billion cells dataset where the 2048 sub-domains of the simulation shown as blocks are colored by the distribution on the 1024 processors used to explore the dataset using $1024 \times 2\text{GB} \sim 2\text{TB}$ of memory.

7.4.4 Optimization of the Visualizations with External Data

In order to help the interpolation needed by the visualization algorithms among blocks, we added ghost zones support to the reader. These extra cells, duplicated from the cells in the neighboring domain, enable smooth interpolation and are used to remove cracks in iso-surfaces at the boundaries of the blocks. We provided this extra information from the simulation code by transferring the neighboring block connectivity to the visualization algorithms (using the `avtCurvilinearDomainBoundaries`). The information needed describes the orientation of

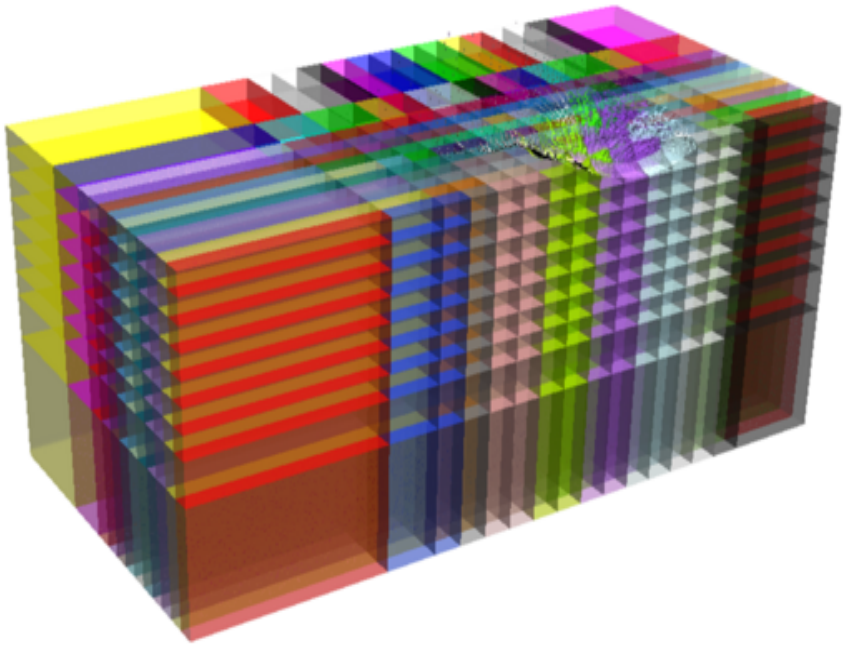


Figure 7.7: Extraction of 2-billion cells dataset where the 2048 distributed sub-domains of the simulation shown as blocks are colored by the distribution on the 1024 visualization processors using a total of 2 TB of memory.

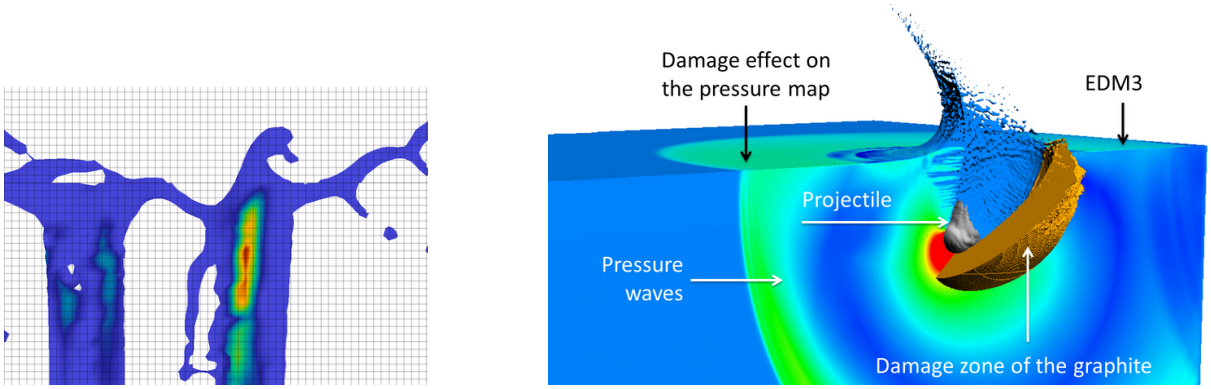
each block and the zone of the neighboring interval. On some data, complex inter-block connectivity can be addressed, such as T-intersections with this method.

Material Interface reconstruction is a key feature in analyzing Eulerian simulation code data as illustrated in Figure 7.8a. As explained in subsection 7.3.4, *VisIt* implements several visualization algorithms. The feature is enabled by providing to the database reader the precise description of the composition of each cell. The structure used in the `AuxilaryData` method is based on several link arrays describing the material of pure cells, the fraction of presence of each material in every mixed cell, and the empty cells. The implementation has to convert the data given from the input file to the *VisIt* data structure.

7.4.5 Analysis

The visualizations obtained in this project [Vivodtzev and Bertron, 2014] allow the scientists to analyze the effects of the impact such as crater volume, fragment velocities, and ejected fragmented volume. In addition to the crater depth and diameter, we can compare the crater shape with the experiment thanks to the visualization. We can also observe the deformation of the projectile and compare its penetration depth with experimental data. Visualizing pressure wave propagation and interaction, as shown in Figure 7.8b, with the material surface allows the understanding of the damage process in the graphite.

This interpretation can be complemented by visualizing the damage variable also shown in



(a) (b)

Figure 7.8: (a) Material interface reconstruction on a result of an Eulerian code. (b) Visualization of the pressure and the damage of a graphite sample with the 45-degree oblique impact of a steel projectile around 4km/s.

Figure 7.8b around the crater and its retroaction on the pressure wave propagation. During this campaign in 2014, the access and initialization of the 1024 compute engines on TERA [CEA TERA] took about 45 seconds. Then the 2048 data files were loaded in about 30 seconds. The computation of the visualization took less than 5 seconds depending on the algorithms used.

7.5 VR Software in a Linux Restricted Environment

As explained in the previous section, scientific visualization software such as *VisIt* and Paraview are adapted to the analysis of simulation data, most of the time realized on a 2D screen of a computer. However, in some cases, virtual reality (VR) technologies can be used to improve the understanding and exploration of scientific data. Many frameworks exist to handle the projection of visualizations into a VR scene such as Unity [Unity Technologies], Ansys EnSight[Ansys], or Unreal Engine [Epic Games]. It is also possible to project a ParaView render screen into a VR scene using dedicated middleware (like SteamVR [Valve] for example). However, most of these solutions work well with different head-mounted displays (HMD) connected to middleware and drivers but mostly in a Windows operating system environment.

For the needs of scientists working on restricted areas, we were interested in creating a VR environment for data exploration in the field of CAD and numerical simulation. The objective is to develop a VR workflow that runs on a restricted network isolated from the Internet and running mainly on Linux. The CAD to explore and the simulation results to analyze are generated on supercomputers connected to the same restricted network. In order to offer the users VR capabilities on their data, we are investigating solutions to set up an environment on Linux that minimizes the transfer between networks. We are also interested in open-source solutions such as Monado[Monado] and the Godot[Godot] engine in order to meet the security requirements of such a restricted network. The work on the software is targeting a workstation with an Nvidia GeForce GTX 1080 to which an HTC Vive Pro[Vive] is connected. We investigated several use cases on VR as described in the next sections.

7.5.1 Use Case : CAD Exploration

This first use case corresponds to a CAD visualization of an assembly line. The 3D model contains 7000 objects and 800,000 polygons. The goal was to create a VR environment directly from the CAD file exported from the engineering office via the CATIA [Dassault Systèmes, b] software, without spending too much time processing it or creating a hierarchy. Many software can be used to explore this type of CAD model, such as TechViz XL [Techviz], 3DX [Dassault Systèmes, a], or CAD-to-VR [Autodesk].

For this use case, we used the 3D game engine Unity [Unity Technologies] and various assets to load and manipulate the 3D model. First, the plugin PiXYZ [Unity Asset Transformer] allows us to load the hierarchy of the CAD file and create a Unity object with the right meshes and colliders to prepare the VR integration. Then we use the Unity plugin Interact [LS Group] to create the scene with interactions and some default material properties for the objects as shown in Figure 7.9.

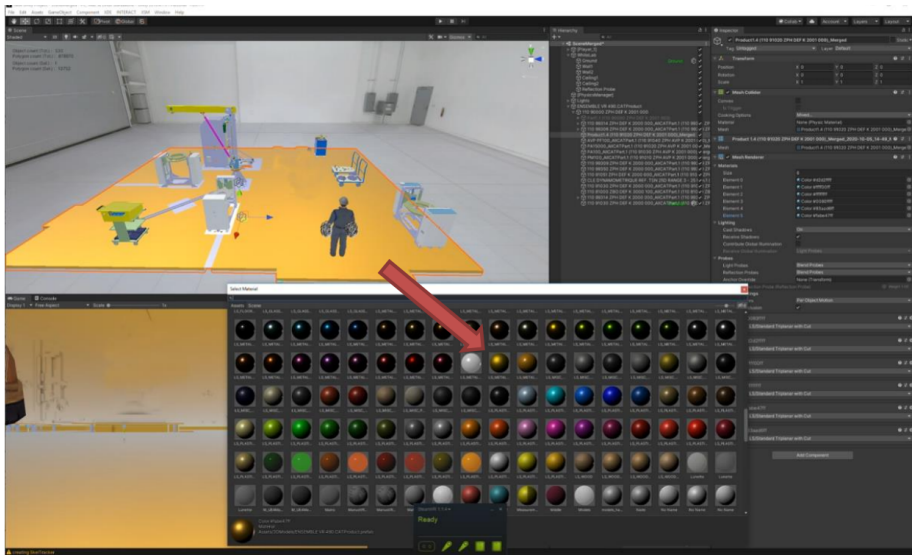


Figure 7.9: Automatic integration of a CAD model of an assembly line into the game engine Unity

7.5.2 Use Case : Identifying Mesh Failures

Our second use case is dedicated to methods to explore singularities or errors introduced into a mesh during the generation. To better track features into the meshes we implemented a specific interaction and render adapted to meshes using the VRTK [VRTK] Unity asset. The implementation allows us to define material properties, rigid bodies for physics, highlighting of objects, and volume definition for the colliders.

To answer the need of scientists, we focus on the design of simple 3D interactions for non-expert users in VR to allow them to easily get familiar with the visualization of a mesh in the

HMD. To facilitate the navigation, the workstation shows the user's view and some controls to change the size and the view in case the user gets lost. We also added a simple collaboration tool between the user and the controller to allow the user to point with a laser-style ray to some artifacts into the mesh as shown in Figure 7.10. Non-expert VR users have had a very positive experience with this immersive mesh exploration. They felt comfortable in the VR scene and appreciated the high visual impact of this visualization, especially around mesh inconsistencies.

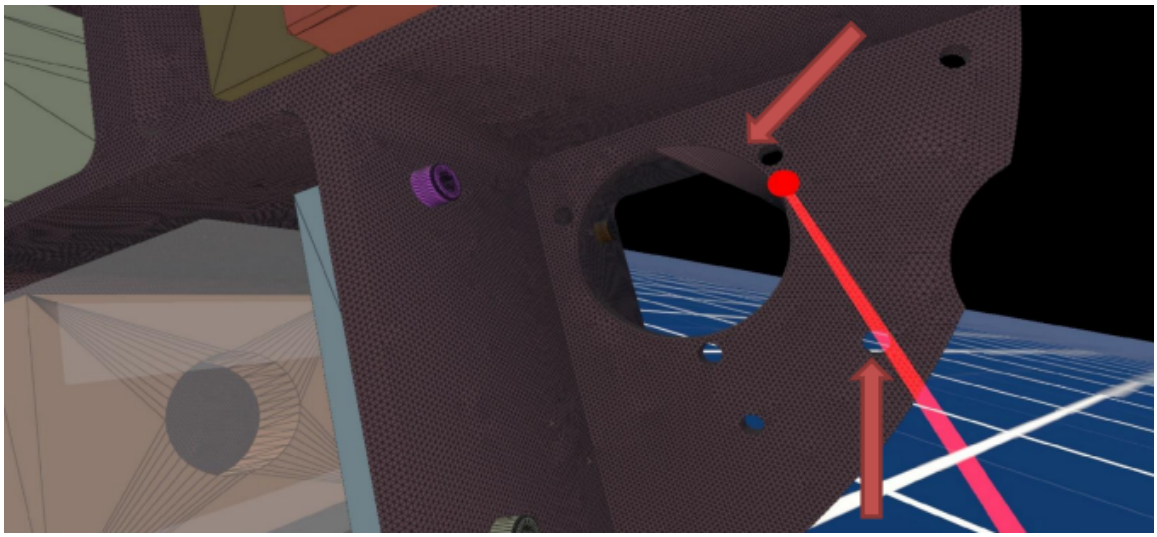


Figure 7.10: The mesh features visualization in a Unity VR scene, with collaborative tools between the non-expert users and a controller on the workstation.

7.5.3 Use Case : Visualization of Simulation Data

Our third use case is the exploration of simulated data with a scientific visualization pipeline to extract isosurfaces or perform volume rendering. To take advantage of existing solutions, we have used the VR plugin of ParaView[Ahrens et al., 2005a] named *OpenVR* (recently renamed to *XRInterface* in ParaView 5.11). The data is loaded from the ParaView GUI on the workstation and the visualization pipeline is set up. The rendering view is then sent to the HMD display with several interactions adapted to scientific visualization, such as translation, rotation, and scaling but also cropping and picking.

7.5.4 VR Open-source Software Setup

The use cases described above show that VR could help scientists to understand different types of data. Even new VR users can benefit from this technology with visualizations adapted to the application domain (CAD, meshing, simulation). The various experiments also show that VR workstations should be as close as possible to the working environment in order to be easily used by the expert. A dedicated room for VR far from the user's workstation or an additional

transfer between networks should be avoided as much as possible. Therefore, we are trying to reproduce such VR experiences, but on the network of the supercomputer where the data are explored.

We used a restricted network from CEA under Linux with high software security requirements, and without internet access. SteamVR and Unity cannot be deployed for every user due to the security requirements. That's why we are investigating the use of open-source software running on Linux to control the HMD. In order to simplify the deployment of all required VR software on the restricted network, we prepare a container using the Flatpak [Flatpak Team] tool. From an open network, we first configure the proxy to download the required packages. Then the home directory and download repository are set up. After the configuration is done into this container, an archive is created for transfer and deployment on the restricted network.

The HMD needs middleware to communicate with the VR workstation drivers. Our first experiment was made on the open network, with a support of the HTC Vive Pro in Godot using SteamVR. Since we want to simplify software installation on the secure side, we are now investigating how to use an open-source alternative like *Monado* [Monado]. This open-source XR runtime brings immersive XR experiences to workstations, mobile phones, and other devices. Monado aims to be a complete implementation of the OpenXR API created by Khronos and runs on Linux.

7.5.5 Build VR Scene in Godot

As mentioned earlier, there is a lot of software available to build a 3D scene ready for VR. In this project, we focus on lightweight open-source solutions. Therefore, we will use the Godot 4 engine, which is a free open-source software to build desktop and mobile applications in 2D and 3D. We package the engine as described in the previous section by adding the VR capabilities. This is done using the Godot XR tool [XR Tools] plugin and special shaders to enable the stereoscopic views in the engine.

First, we import data models for the VR scene tool with a CAD file exported from CATIA and converted with the open-source FreeCAD [FreeCAD] software to a glTF [Khronos] file. The hierarchy of the complex CAD model is preserved and ready to be imported into VR software. Then the 3D nodes are created into a Godot scene with a `Node3D` named `Main`, an `XROrigin3D` node as a child of `Main`, an `XRCamera3D` node as a child of `XROrigin3D`, and two `XRController3D` nodes as children of `XROrigin3D` for the left and right hand. In the Inspector panel, the `Tracker` is configured for the left and right hand. Then a mesh is added as a child of the left and right hand to visualize them and different 3D objects like a plane mesh for the ground. Finally, the OpenXR tools is activated for the scene with the add-on *godot-xr-tools*. The use of a `VRCommonShaderCache` node to the `XRCamera3D` tells Godot to use dedicated shaders from the XR Tools library. With all the previous steps, we successfully created a simple VR scene including the CAD model with appropriate material properties, object hierarchy, and simple interaction tools as shown in Figure 7.11.

These several VR experiments with different software and data illustrate real-world applica-

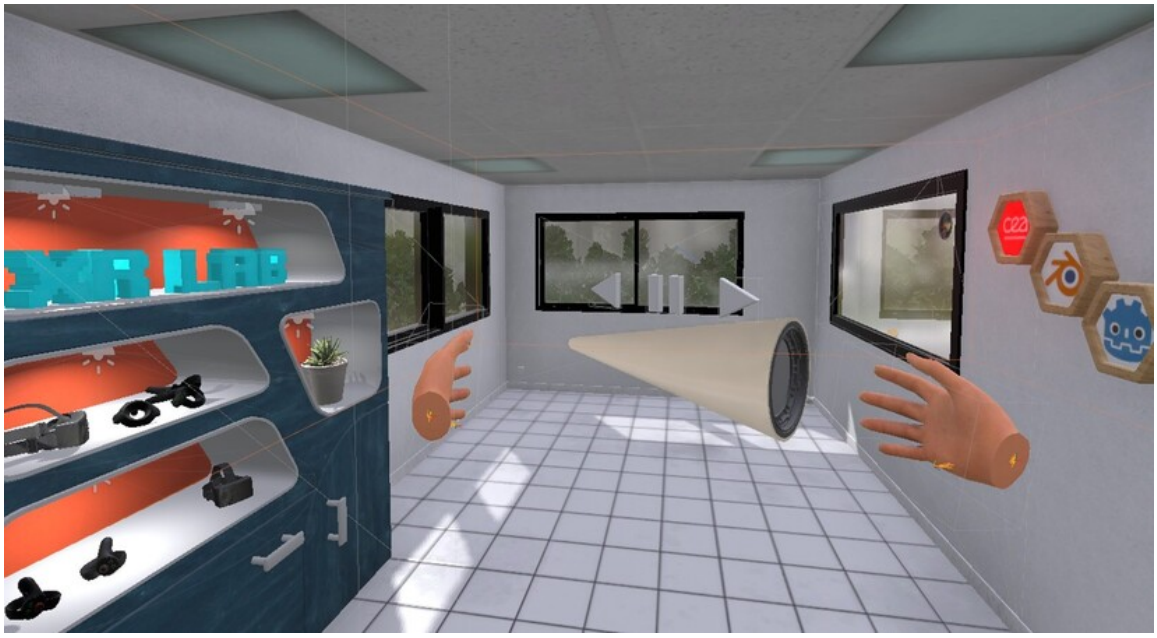


Figure 7.11: VR environment displays in a HMD from a Godot scene with interaction and animation of a CAD model.

tions of VR for CAD and scientific data exploration used by scientists. The experience shows that to enhance VR capabilities, users need to have a running VR environment as close as possible to their data. Therefore, we showed how to set up a lightweight and portable VR environment on a restricted network. However, software configuration for VR technologies can be cumbersome on a network disconnected from the Internet. Thus further work needs to be done on the driver side to fully replace SteamVR with Monado. Also, new technologies for HMD arise each year like the Vision Pro from Apple in 2024, making the software move constantly to be adapted to scientific data.

CHAPTER 8

Conclusion and Future Work

To conclude this thesis, I summarize in this chapter the highlights of research projects carried out over the last twenty years and the lessons learned for future work. I present my vision of the data analysis challenges on which my research will focus in the coming years. Indeed, the heterogeneity of needs in terms of data analysis has led me to invest in many theoretical and practical aspects of scientific visualization. The interdisciplinarity required to address these challenges and the necessarily collaborative approach have been sources of inspiration in proposing methods adapted to scientific needs.

8.1 Research Project Overview

Topological data analysis is a particularly effective method of simplification when data complexity impedes understanding. By identifying abstractions from the application domain into a simplified representation, topological signatures can be generated. The field of computational topology focuses on these structures as presented in the chapter 3. Much of my research has focused on adapting abstraction methods for many different scientific applications.

My early work in topology contributed to mesh simplification by edge contraction, preserving the topology of sub-structures within unstructured tetrahedral meshes. The innovative idea behind this work is to extend the simplicial complex by a cone of simplexes, building on the simplexes to be preserved, in order to represent all multidimensional features in the same extended complex. Using the link condition on the extended complex, we were able to identify and guarantee edge contractions that preserve the topology of the mesh and that of all its sub-structures. This simplification method has been proposed for multi-resolution data visualization

in electromagnetism and human cortex comparison in neuroscience, as detailed in chapter 4.

Subsequently, to analyze the results of computational codes, I became interested in the study of scalar fields carried by these simplifying complexes. In this respect, persistent homology studies the evolution of the topology of sets of sublevels, extracted from scalar fields, during filtering in the vicinity of particular points called critical points. The classification of these points is also performed using a combinatorial approach by examining neighboring simplexes. The notion of topological persistence can then be used to describe their importance, with representations in the form of diagrams or persistence curves widely used in my data analysis projects when simplification is required. In addition, the abstraction of domain-specific data into topological structures such as contour trees or the Morse-Smale complex makes it possible to identify topological signatures relevant to analysis. Over the last ten years, we have proposed a number of topological analysis pipelines to help understand physical simulation results. These projects are described in chapter 5 and chapter 6 for the comparison of large ensembles in fluid dynamics, the segmentation of hydrodynamic instabilities, or the identification of turbulent vortices in hypersonic flow.

Finally, the use of topological analysis methods in a high-performance computing context requires a great deal of work on data and visualization software to enable scientists to exploit their results. In parallel with this research work on TDA, we have been conducting research projects on scientific visualization tools, as detailed in chapter 7. Several approaches have been investigated, depending on the year and the project. Remote visualization tools transposing only images to the user's workstation have been implemented, with tools such as VisIt and Paraview. To integrate the analysis stages as closely as possible with the calculation code, data model visualizations were proposed within a simulation platform developed in the form of a Java client and views in VTK, requiring the extension of this library's functions. Work on high-performance data analysis (HPDA) was also carried out, with particular emphasis on the data formats generated by CEA's calculation codes.

The research I was able to carry out at the CEA, in a multidisciplinary environment, with a variety of research engineer profiles, showed me the importance of communicating and popularizing the data. This is why I also worked on 3D computer graphics and virtual reality to improve understanding and communication of the physical phenomena studied. Data processing has enabled me to achieve a high level of rendering of 3D computer graphics scenes from simulation data such as CAD, meshes, or calculation results. Automation methods facilitated the modeling, rendering, and animation of 3D computer graphics models in Blender. We have thus produced over thirty scientific computer graphics films to help popularize concepts proposed by the CEA on a wide range of subjects, such as innovative materials or the phases of ballistic flight. These films combine computer graphics renderings and scientific visualizations.

They are often used for training in poorly documented, often classified fields, or to present simulation activities at CEA. The success of these projects required a great deal of work in terms of computational data preparation, interviews with experts in the field, and scripting. To enhance understanding of the scientific data presented in the form of 3D computer graphics, I also designed a series of immersive experiences using virtual reality as close as possible to the workstations of research engineers. Due to security requirements, the deployment of such an environment on a secure Linux-based network not connected to the Internet necessitated

software implementation work. We set up scientific data scenes in a virtual reality headset using the Godot open-source video game engine.

8.2 Lessons Learned for Future Work

Combinatorial approach : Hebert Edlsbrunner's work showed me early on the power of combinatorial methods for data analysis. Topology preservation tests based on link condition extension are just one example. Computational topology is a discipline that perfectly exploits the theoretical foundations of continuous domain topology with an expression in the discrete domain and obtained by combinatorial descriptions of the elements of interest (vertices, cells, critical points, etc.). In the search for new topological structures, it is therefore essential to rely on robust combinatorial properties derived from topology.

Data management first : In an HPC context and with increasingly complex data from simulation or experimental acquisition systems, analysis pipelines need to integrate more and more steps at all levels of the data lifecycle. In an industrial or research laboratory environment, it is crucial to maintain an in-depth understanding of data formats, codes, and acquisitions. This work on HPDA is sometimes more engineering than research, but will always be a major time-saver in the overall analysis pipeline. Data preparation is crucial for the rest of the analysis including TDA.

Multidisciplinary approach : Successful data analysis projects in a scientific context require a multidisciplinary approach. The list of disciplines varies from project to project. By way of example, and based on the experience of my research, we can cite the involvement of physicists to identify quantities and models, mathematicians to identify optimization methods, numerical engineers to optimize numerical schemas, and computer scientists to manage the parallelism of I/O and visualizations. Experience shows that projects combining these different skills are the most successful to reach data understanding.

Hybrid rendering : The most relevant visualizations need not only to be expressed in the visual language of the expert domain but also to offer a context that can be understood by any audience. Over the years, I've produced visualizations for a wide range of audiences, from experts to the general public, and I've been interested in different types of rendering, from computer graphics to virtual reality. In all cases, the objective is a better understanding of simulation data, but rendering methods and tools can be very different and need continuous improvement as much as data analysis.

Collaboration, supervision and teaching : The field of visualization is collaborative by nature, since its aim is to convey information to others. While this is not dedicated only to this domain, it is important to emphasize that collaboration, student mentoring, and teaching scientific visualization are fundamental aspects of the field. The hours invested in these tasks are necessary to sustain a scientific visualization community.

8.3 Perspectives

8.3.1 Data and Software Challenges

High-performance computing has enabled significant progress in the domain of scientific discovery and engineering over the past few decades. As demonstrated in the TOP500 list [TOP500.org, 2025] of the most powerful supercomputers, many of these machines are utilized by institutions such as CEA for the execution of large-scale simulations.

Data preparation : It is therefore essential that the data and software used for analysis are enhanced. The most recent simulations now involve high-resolution discretization, which results in the creation of extensive meshes and variable data. The preparation of data for an HPC environment is contingent upon the origin and structure of the input sources. A number of challenges must be addressed, including I/O bottlenecks, resource scheduling, and fault tolerance. As outlined in this document and my research, which spans more than 20 years, this multidisciplinary approach enhances the analysis of simulation data. Therefore, it is imperative to consider the integration of in-memory data from HPC simulations within the visualization strategy. This should be achieved through the use of APIs such as Catalyst, Conduit, Adios, and parallel HDF5. In the course of our research, conjointly with code developers we will enhance the management of data between the simulation code and the visualization pipeline, as outlined in [Vivodtzev and Stauffert, 2024] for a various projects within an HPC environment.

AI visualization paradigm : Machine learning approaches can automate the identification of known patterns in large datasets. Data preparation augmented with AI features can ease data management and accelerate visualization pipeline generation. Toward augmented analytic or multimodal large language models (MLLM) for visualization such as [Liu et al., 2025], we will investigate domain-specific analytics. In the context of restricted networks away from the internet such as at CEA, AI systems need to be tuned with domain-specific analytics. We will consider exploration pipelines, user training, or meaningful visualizations to build retrieval augmented generation models (RAG) and thus benefit from AI advantages on HPDA. This agent-driven visualization paradigm will need to be investigated to rethink the interaction with data and visualization tools.

8.3.2 Domain Specific Exploration Challenges

The topological data analysis methods proposed in my research have made it possible to extract features that traditional methods would not have been able to achieve, or at the cost of unacceptable computation. On the basis of these convincing results in several application fields, we have identified other areas where specialization is required.

TDA for material design : The design of new materials is a key area of research for many applications, particularly in aerospace. Modeling innovative materials composed of complex structures involves finding compromises between numerous characteristics, such as resistance to high temperatures, abrasion, lightness, and transparency. In the case of protective

materials, numerous compounds can be used such as ceramics whose rigid atomic structure confers exceptional hardness and thermal resistance. Ceramic matrix composites (CMC) use a ceramic matrix and a fibrous reinforcement (alumina or carbon). I will explore visualization strategy with topological data analysis methods to improve the understanding of 3D protective materials subjected to high stress like mechanical shock, laser energy deposition, or projectile impact. This analysis will be used to enhance the numerical models developed to characterize porous microstructure and microcracks.

TDA for experimental data : Numerical simulation data is frequently characterized as massive, particularly when considering the utilization of high-performance computing capabilities, which often results in data sets of a high resolution. Nonetheless, experimental data are frequently characterized by a high degree of variability and a lack of uniformity with regard to temporal and spatial resolutions. By experience it can be tedious to implement supervised tracking methods on these data and an even more arduous task to build up a learning database. In future work on material resistance we will investigate TDA methods on experimental videos showing the impacts of projectiles launched at speeds in excess of $10,000 km/h$. The shock wave within the material fragments it into a large number of splinters of varying sizes, which are ejected out of the material at varying speeds (from 3 to $4,000 km/h$). Statistical understanding of the population of these fragments in terms of size and velocity is a crucial element in assessing the strength of the material. But only a few videos exist with limited resolution and time frames because of their high production costs. Moreover, each frame of these videos contains several hundred fragments, making reliable manual labeling impractical. We obtained preliminary results based on the optimal matching of persistence diagrams of critical points built on debris in videos. Thus I will propose methods to improve feature tracking in such data sparse in a sense and sometimes heterogeneous depending on the sensor to acquire it.

Other physics : In a multi-physics environment, the implementation of topological analysis techniques is an important task in the process of continuous improvement of simulation workflows. When a physicist expresses an advanced need for analysis, then the evolution of TDA methods and tools is necessary, as demonstrated in the projects of this thesis. Conversely, for many other physics, it is also interesting to go the other way and propose topological abstractions to experts. The aim is to explore the unexpected. For my future research, I'll be evaluating persistence-based segmentation, tracking, and simplification methods on a wide variety of simulation and experimental data. The aim is to make proposals to physicists to improve exploration. In the coming years, I'll be looking at the application of TDA to a variety of application domains, such as tomography data, scanning electron microscope acquisitions, electromagnetic vulnerability simulations, pyrolysis phenomena, neural network optimizers.

8.3.3 Display Challenges

To communicate the complex results of simulations, I have developed data processing methods to showcase the results of calculation codes in 3D computer graphics films. These methods remain time-consuming in terms of manipulation. However, communicating meaningful visualizations is an important part of my research. My main objective is to improve the storytelling of complex physical phenomena.

Fill the gap between design space and computer graphics : Research needs to be done to fill the gap between the universe of design and engineering with the illustrative field of computer graphics and visualization. We will work to optimize the time to prepare CAD models and meshes and be able to use them for visualization. Integration in game engines such as Unity or Godot will be part of my research work. Scene construction with structures such as the Universal Scene Descriptor (USD) and a link to a software environment like Omniverse from nVidia will be investigated. Collaboration with CEA-List, Inria, or the Center for Computing Research at Sandia National Laboratory, for CAD preparation, scene description, or interaction will be engaged.

Adapt displays to science : As explained in chapter 7, virtual reality technologies evolve quickly driven by the entertainment companies. For scientific use and especially in a secure environment, we will collaborate with laboratories such as Los Alamos National Laboratory to evaluate Head Mounted Displays and their software environment adapted to scientific restricted networks. A key aspect of my research will be to bring the scientific working environment closer to the immersive world.

The enhancement of understanding of complex simulation data and collaboration with many people will be the primary motivations for my future research. I will provide training and support to students and direct my research to contribute to these challenging domains of scientific visualization and data analysis.

Bibliography

H. Abu-Shawareb, R. Acree, P. Adams, J. Adams, B. Addis, R. Aden, P. Adrian, B. Afeyan, M. Aggleton, L. Aghaian, et al. Lawson criterion for ignition exceeded in an inertial fusion experiment. In *Physical review letters*, volume 129, page 075001. APS, 2022.

A. Acharya and V. Natarajan. A parallel and memory efficient algorithm for constructing the contour tree. In *IEEE PV*, 2015.

J. Ahrens, B. Geveci, and C. Law. ParaView: An end-user tool for large data visualization. In *Visualization Handbook*. Elesvier, 2005a. ISBN 978-0123875822.

J. Ahrens, B. Geveci, and C. Law. ParaView: An End-User Tool for Large-Data Visualization. In *The Visualization Handbook*, pages 717–731. Academic Press, Inc., 2005b.

K. Anderson, J. Anderson, S. Palande, and B. Wang. Topological data analysis of functional MRI connectivity in time and space domains. In *MICCAI Workshop on Connectomics in NeuroImaging*, 2018.

Ansys. Ansys EnSight, https://www.ansys.com/products/fluids/ansys-ensight.

T. M. Athawale, D. Maljovec, C. R. Johnson, V. Pascucci, and B. Wang. Uncertainty visualization of 2d morse complex ensembles using statistical summary maps. volume abs/1912.06341, 2019. URL http://arxiv.org/abs/1912.06341.

Autodesk. CAD-to-VR, https://www.autodesk.com/developer-network/certified-apps/cad-to-vr.

U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. Mauldin. Paraview catalyst: Enabling in situ data analysis and visualization. In *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV2015)*, pages 25–29, Austin, TX, USA, 2015. Association for Computing Machinery. doi: 10.1145/2828612.2828624. URL https://doi.org/10.1145/2828612.2828624.

J. C. Baez. Open Questions in Physics. Technical report, UC Riverside, Department of Mathematics, 2006.

K. Bai, C. Wang, M. Desbrun, and X. Liu. Predicting high-resolution turbulence details in space and time. *ACM Trans. Graph.*, 2021.

T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 77(5):475–485, 1970.

U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *Algorithm Engineering and Experiments*, 2014.

J. Bennett, F. Vivodtzev, and V. Pascucci. *Topological and Statistical Methods for Complex Data: Tackling Large-Scale, High-Dimensional, and Multivariate Data Spaces*. Springer, 01 2015. ISBN 978-3-662-44899-1. doi: 10.1007/978-3-662-44900-4.

D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 1981.

H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer. Topoms: Comprehensive topological exploration for molecular and condensed-matter systems. 2018.

S. Biasotti, D. Giorgio, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. 2008.

T. Bin Masood, J. Budin, M. Falk, G. Favelier, C. Garth, C. Gueunet, P. Guillou, L. Hofmann, P. Hristov, A. Kamakshidasan, C. Kappe, P. Klacansky, P. Laurin, J. Levine, J. Lukasczyk, D. Sakurai, M. Soler, P. Steneteg, J. Tierny, W. Usher, J. Vidal, and M. Wozniak. An Overview of the Topology ToolKit. In *TopoInVis*, 2019.

R. D. Blevins. Flow-induced vibration. *New York, Van Nostrand Reinhold Co., 1977. 377 p.*, 1977.

A. Bock, H. Doraiswamy, A. Summers, and C. T. Silva. TopoAngler: Interactive Topology-Based Extraction of Fishes. 2018.

G. Boffetta and R. E. Ecke. Two-dimensional turbulence. *Annual Review of Fluid Mechanics*, 2011.

R. Borges, M. Carmona, B. Costa, and W. S. Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 2008.

P. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A Multi-Resolution Data Structure for 2-Dimensional Morse Functions. In *Proc. of IEEE VIS*, 2003.

P. 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. 2011.

T. Bridel Bertomeu. *Investigation of unsteady phenomena in rotor/stator cavities using Large Eddy Simulation*. PhD thesis, 2016.

T. Bridel-Bertomeu. Immersed boundary conditions for hypersonic flows using eno-like least-square reconstruction. *Comp. & Flu.*, 2021.

T. Bridel-Bertomeu, B. Fovet, J. Tierny, and F. Vivodtzev. Topological analysis of high velocity turbulent flow. In *2019 IEEE 9th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 87–88, 2019. doi: 10.1109/LDAV48142.2019.8944351.

R. Bujack, L. Yan, I. Hotz, C. Garth, and B. Wang. State of the art in time-dependent flow topology: Interpreting physical meaningfulness through mathematical properties. 2020.

B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. Technical report, LLNL, 1993.

H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Symp. on Dis. Alg.*, 2000.

H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003. ISSN 0925-7721. doi: https://doi.org/10.1016/S0925-7721(02)00093-7. URL https://www.sciencedirect.com/science/article/pii/S0925772102000937. Special Issue on the Fourth CGC Workshop on Computational Geometry.

H. A. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE VIS*, 2004.

H. A. Carr, G. H. Weber, C. M. Sewell, and J. P. Ahrens. Parallel peak pruning for scalable SMP contour tree computation. In *LDAV*, 2016.

A. Casner. Recent progress in quantifying hydrodynamics instabilities and turbulence in inertial confinement fusion and high-energy-density experiments. In *Philosophical Transactions of the Royal Society A*, volume 379. The Royal Society Publishing, 2021.

A. Casner, V. Smalyuk, L. Masse, I. Igumenshchev, S. Liberatore, L. Jacquet, C. Chicanne, P. Loiseau, O. Poujade, D. Bradley, et al. Designs for highly nonlinear ablative rayleigh-taylor experiments on the national ignition facility. In *Physics of Plasmas*, volume 19. AIP Publishing, 2012.

A. Casner, L. Masse, S. Liberatore, P. Loiseau, P. Masson-Laborde, L. Jacquet, D. Martinez, A. Moore, R. Seugling, S. Felker, et al. Probing the deep nonlinear stage of the ablative rayleigh-taylor instability in indirect drive experiments on the national ignition facility. In *Physics of Plasmas*, volume 22. AIP Publishing, 2015.

CEA and Kitware. Provide swt based vtk rendering component for the java api. https://github.com/Kitware/VTK/blob/master/Wrapping/Java/vtk/rendering/swt/vtkSwtComponent.java, 2012.

CEA EXA, 2025. CEA Computing centers, https://www-hpc.cea.fr/en/EXA.html.

CEA TERA, 2001. CEA Computing centers, https://www-hpc.cea.fr/en/TERA.html.

M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.*, 40(1):200–210, Jan. 2013. ISSN 0957-4174. doi: 10.1016/j.eswa.2012.07.021. URL https://doi.org/10.1016/j.eswa.2012.07.021.

L. Ceurvorst, R. Betti, A. Casner, V. Gopalaswamy, A. Bose, S. Hu, E. Campbell, S. Regan, C. McCoy, M. Karasik, et al. Hybrid target design for imprint mitigation in direct-drive inertial confinement fusion. In *Physical Review E*, volume 101, page 063207. APS, 2020.

S. Chandrasekhar. *Hydrodynamic and hydromagnetic stability*. Courier Corporation, 2013.

H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, K. Bonnell, M. Miller, G. Weber, C. Harrison, D. Pugmire, T. Fogal, C. Garth, A. Sanderson, E. Bethel, M. Durant, D. Camp, J. Favre, O. Rubel, P. Navratil, M. Wheelera, P. Selbya, and F. Vivodtzev. Visit: An enduser tool for visualizing and analyzing very large data. In *Proceedings of SciDac 2011*, 2011.

D. Cohen-Steiner and J.-M. Morvan. Restricted Delaunay triangulations and normal cycle. In *Proceedings of the Annual Symposium on Computational Geometry (SCG-03)*, pages 312–321, New York, NY, USA, 2003. ACM Press.

D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. In *SoCG*, 2005.

Dassault Systèmes, a. 3DX, https://www.3ds.com/3dexperience.

Dassault Systèmes, b. Catia, https://www.3ds.com/fr/produits-et-services/catia/.

L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo. Morse complexes for shape segmentation and homological analysis: discrete models and algorithms. 2015.

T. Dey, H. Edelsbrunner, S. Guha, and D. Nekhayev. Topology preserving edge contraction. 1998.

T. Dey, H. Edelsbrunner, S. Guha, and D. Nekhayev. Topology preserving edge contraction. *Publications de l'Institut, Mathematique (Beograd)*, 60, 11 1999.

J. Dongarra. Performance of Various Computers Using Standard Linear Equations Software. Technical Report CS-89-85, University of Tennessee, Knoxville, TN, 1989. URL http://www.netlib.org/benchmark/performance.ps. The date provided in the query is a placeholder for the date accessed, not the publication date of the report itself. The actual publication year is 1989.

H. Doraiswamy and V. Natarajan. Computing reeb graphs as a union of contour trees. 2013.

H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. AMS, 2009.

H. Edelsbrunner and E. P. Mucke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. 1990.

H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. In *SoCG*, 2003a.

H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manfiolds. 2003b.

C. Elkan. Using the triangle inequality to accelerate k-means. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 147–153. AAAI Press, 2003. ISBN 1577351894.

Epic Games. Unreal Engine, https://www.unrealengine.com/.

G. Favelier, C. Gueunet, and J. Tierny. Visualizing ensembles of viscous fingers. In *IEEE SciVis Contest*, 2016.

G. Favelier, N. Faraj, B. Summa, and J. Tierny. Persistence Atlas for Critical Point Variability in Ensembles. *IEEE TVCG*, 2018.

C. L. Fefferman. Existence and Smoothness of the Navier-Stokes Equation. Technical report, Clay Mathematics Institute, 2000.

F. Ferstl, K. Bürger, and R. Westermann. Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE TVCG*, 2016.

Flatpak Team. Flatpak, https://flatpak.org/.

L. D. Floriani, E. Puppo, and P. Magillo. *A formal approach to hypersurface multiresolution modeling*. Springer, 1997.

L. D. Floriani, P. Magillo, and E. Puppo. The mt (multi-tesselation) package, 2000. URL http://www.disi.unige.it/person/MagilloP/MT. DISI, University of Genova, Italy.

R. Forman. A User's Guide to Discrete Morse Theory. 1998.

FreeCAD. FreeCAD, https://www.freecad.org/.

P. J. Frey and P.-L. George. *Mesh Generation: Application to Finite Elements*. ISTE, 2007. ISBN 1903398002.

L. Fu, X. Y. Hu, and N. A. Adams. A family of high-order targeted eno schemes for compressible-fluid simulations. In *Journal of Computational Physics*, volume 305, pages 333–359. Elsevier, 2016.

A. R. Fuller, B. Hamann, K. I. Joy, E. G. Jones, L. Linsen, B. A. Olshausen, T. W. Slankard, J. Stone, F. Vivodtzev, G. H. Weber, D. F. Wiley, and P. C. Yau. Brain atlas mapping. Presented at the 34th Annual Meeting of the Society for Neuroscience, 2003. Poster presentation.

M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of the Conference on Visualization '98*, VIS '98, page 263–269, Washington, DC, USA, 1998. IEEE Computer Society Press. ISBN 1581131062.

M. Garland and Y. Zhou. Quadric-based simplification in any dimension. *ACM Trans. Graph.*, 24(2):209–239, Apr. 2005. ISSN 0730-0301.

C. Garth and X. Tricoche. Topology- and feature-based flow visualization: Methods and applications. In *VLDUDS*, 2006.

Godot. Godot Game Engine https://godotengine.org/.

D. Guenther, R. Alvarez-Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny. Characterizing molecular interactions in chemical systems. 2014.

C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-based Augmented Reeb Graphs with Dynamic ST-Trees. In *EGPGV*, 2019a.

C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-based Augmented Contour Trees with Fibonacci heaps. *IEEE TPDS*, 2019b.

H. Guo, W. He, T. Peterka, H. Shen, S. M. Collis, and J. J. Helmus. Finite-time lyapunov exponents and lagrangian coherent structures in uncertain unsteady flows. *IEEE Trans. Vis. Comput. Graph.*, 2016.

A. Gyulassy, M. A. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. 2007.

A. Gyulassy, P.-T. Bremer, and V. Pascucci. Computing morse-smale complexes with accurate geometry. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2014–2022, 2012. doi: 10.1109/TVCG.2012.209.

A. Gyulassy, P. Bremer, R. Grout, H. Kolla, J. Chen, and V. Pascucci. Stability of dissipation elements: A case study in combustion. In *CGF*, 2014.

A. Gyulassy, A. Knoll, K. Lau, B. Wang, P. Bremer, M. Papka, L. A. Curtiss, and V. Pascucci. Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials. 2015.

A. Gyulassy, P. Bremer, and V. Pascucci. Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy. 2018.

T. Günther and I. Baeza Rojo. Introduction to Vector Field Topology. In *Topological Methods in Data Analysis and Visualization VI*. Springer, 2021.

B. Hamann. Curvature approximation for triangulated surfaces. In G. Farin et al., editors, *Geometric Modelling*, Computing Supplementum, pages 139–153. Springer-Verlag, Vienna, Austria, 1993.

K. Hanser, O. Klein, B. Rieck, B. Wiebe, T. Selz, M. Piatkowski, A. Sagristà, B. Zheng, M. Lukácová, G. Craig, H. Leitte, and F. Sadlo. Visualization of Parameter Sensitivity of 2D Time-Dependent Flow. In *Proc. of International Symposium on Visual Computing*, 2018.

A. Harten, P. D. Lax, and B. v. Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. In *SIAM review*, volume 25, pages 35–61. SIAM, 1983.

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 *CGF*, 2016.

A. K. Henrick, T. D. Aslam, and J. M. Powers. Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points. *Journal of Computational Physics*, 2005.

H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. 1993.

X. Hu and N. A. Adams. Scale separation for implicit large eddy simulation. *Journal of Computational Physics*, 2011.

X. Hu, Q. Wang, and N. A. Adams. An adaptive central-upwind weighted essentially non-oscillatory scheme. *J. of Comp. Phys.*, 2010.

M. Hummel, H. Obermaier, C. Garth, and K. I. Joy. Comparative visual analysis of lagrangian transport in CFD ensembles. *IEEE TVCG*, 2013.

M. Jarema, J. Kehrer, and R. Westermann. Comparative visual analysis of transport variability in flow ensembles. 2016.

G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.

Y. Kaneda, T. Ishihara, M. Yokokawa, K. Itakura, and A. Uno. Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Physics of Fluids*, 15(2):L21–L24, 2003. doi: 10.1063/1.1539474.

L. Kantorovich. On the translocation of masses. *AS URSS*, 1942.

J. Kasten, J. Reininghaus, I. Hotz, and H. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. In *IEEE TVCG*, 2011.

M. Kerber, D. Morozov, and A. Nigmetov. Geometry helps to compare persistence diagrams. *ACM J. of Exp. Algo.*, 2016.

Khronos. glTF, https://www.khronos.org/gltf/.

T. Kim, N. Thürey, D. L. James, and M. H. Gross. Wavelet turbulence for fluid simulation. *ACM Trans. Graph.*, 2008.

K. Kitamura and E. Shima. Towards shock-stable and accurate hypersonic heating computations: A new pressure flux for ausm-family schemes. In *Journal of Computational Physics*, volume 245, pages 62–83. Elsevier, 2013.

The Visualization Toolkit User's Guide. Kitware, Inc., January 2003. URL http://www.kitware.com/publications/item/view/1269.

R. H. Kraichnan. Inertial ranges in two-dimensional turbulence. *Physics of Fluids*, 10:1417–1423, 1967. ISSN 10706631. doi: 10.1063/1.1762301.

R. H. Kraichnan and D. Montgomery. Two-dimensional turbulence. *Rep. Prog. Phys*, 43:547, 1980.

D. E. Laney, P. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. 2006.

R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology-based flow visualization, the state of the art. In *Topology-based Methods in Visualization*. 2007.

R. J. LeVeque et al. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

D. K. Lilly. Two-dimensional turbulence generated by energy sources at two scales. *Journal of Atmospheric Sciences*, 1989.

M.-S. Liou. A sequel to ausm: Ausm+. *J. Comp. Phys.*, 1996.

M.-S. Liou. A sequel to ausm, part ii: Ausm+-up for all speeds. In *Journal of computational physics*, volume 214, pages 137–170. Elsevier, 2006.

S. Liu, H. Miao, and P.-T. Bremer. Paraview-mcp: Autonomous visualization agents with direct tool use. In *Proc. IEEE VIS 2025 Short Papers*, page 00. IEEE, 2025.

X.-g. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.

A. P. Lohfink and C. Garth. Visitation graphs: Interactive ensemble visualization with visitation maps. In *iPMVM*, 2020.

LS Group. Interact, https://www.ls-group.fr/fr/interact.

S. Maadasamy, H. Doraiswamy, and V. Natarajan. A hybrid parallel algorithm for computing and tracking level set topology. In *Proc. of HiPC*, 2012.

C. Mailliet. *Etude expérimentale et numérique du stade fortement non-linéaire de l'Instabilité de Rayleigh-Taylor au front d'ablation en attaque directe*. Etude expérimentale et numérique du stade fortement non-linéaire de l'Instabilité de Rayleigh-Taylor au front d'ablation en attaque directe, Doctoral dissertation of the Université de Bordeaux, Nov. 2018.

M. Maltrud and G. Vallis. Energy spectra and coherent structures in forced two-dimensional and β-planeturbulence. *J. Flu. Mech.*, 1991.

K. Masatsuka. *I do Like CFD, vol. 1*, volume 1. Lulu. com, 2013.

N. Max. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*, 4(2):1–6, 1999.

G. H. McCall. Cloud and microjet mix: A possible source of yield limitation of the national ignition facility targets. In *Physics of Plasmas*, volume 30. AIP Publishing, 2023.

J. S. Meredith and H. Childs. Visualization and analysis-oriented reconstruction of material interfaces. *Comput. Graph. Forum*, 29(3):1241–1250, 2010.

M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential-geometry operators for triangulated 2-Manifolds. In *VisMath*, Berlin, 2002.

P. Moin and K. Mahesh. Direct numerical simulation: a tool in turbulence research. *Annual review of fluid mechanics*, 1998.

Monado. Monado OpenXR runtime, https://monado.dev/.

G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences, avec les Mémoires de Mathématique et de Physique*, 1781.

J. Munkres. Algorithms for the assignment and transportation problems. *J. of the Society for Industrial and Applied Mathematics*, 1957.

D. Nassiet and F. Vivodtzev. Retour d'expérience utilisateur : comment rendre compatible les problématiques de visualisation distante dans un environnement trés sécurisé. In *Forum TeraTec 2012*, 2012.

D. Nassiet, Y. Livet, M. Palyart, and D. Lugato. Paprika: rapid ui development of scientific dataset editors for high performance computing. In *Proceedings of the 15th International Conference on Integrating System and Software Modeling*, SDL'11, page 69–78, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642252631. doi: 10.1007/978-3-642-25264-8_7. URL https://doi.org/10.1007/978-3-642-25264-8_7.

F. Nauleau, B. Fovet, and F. Vivodtzev. In situ segmentation of turbulent flow with topology data analysis. In *ACM SIGGRAPH 2022 Posters (SIGGRAPH '22)*, SIGGRAPH '22, page Article No.: 61, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450393614. Poster.

F. Nauleau, F. Vivodtzev, T. Bridel-Bertomeu, H. Beaugendre, and J. Tierny. Topological analysis of ensembles of hydrodynamic turbulent flows – an experimental study. In *Proc. of IEEE Symposium on Large Data Analysis and Visualization*, 2022b.

F. Navah, M. de la Llave Plata, and V. Couaillier. A high-order multiscale approach to turbulence for compact nodal schemes. *Computer Methods in Applied Mechanics and Engineering*, 363: 112885, 2020. ISSN 0045-7825.

A. Obukhov. Spectral energy distribution in a turbulent flow. *Dokl. Akad Nauk SSR*, 32:22-24, 1941. ISSN 0045-7825.

M. Olejniczak, A. S. P. Gomes, and J. Tierny. A Topological Data Analysis Perspective on Non-Covalent Interactions in Relativistic Calculations. 2019.

M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector field topology. *CGF*, 2010.

M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3D vector fields. *IEEE PV*, 2011.

S. Parsa. A deterministic *o(m log m)* time algorithm for the reeb graph. In *SoCG*, 2012.

V. Pascucci, G. Scorzelli, P. T. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: simplicity and speed. 2007.

J. Peng, S. Liu, S. Li, K. Zhang, and Y. Shen. An efficient targeted eno scheme with local adaptive dissipation for compressible flow simulation. *Journal of Computational Physics*, 425:109902, 2021.

C. Petz, K. Pöthkow, and H.-C. Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *CGF*, 2012.

A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. The state of the art in topology-based visualization of unsteady flow. 2011.

S. B. Pope. Turbulent flows, 2001.

W. PR and C. P. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:73–115, 1984.

F. Qu, C. Yan, J. Yu, and D. Sun. A study of parameter-free shock capturing upwind schemes on low speeds' issues. *C. T. S.*, 2014.

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 1971.

G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. volume 222, page 76, 1946.

V. Robins, P. J. Wood, and A. P. Sheppard. Theory and Algorithms for Constructing Discrete Morse Complexes from Grayscale Digital Images. 2011.

P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. In *Journal of computational physics*. Elsevier, 1981.

O. Sadot, V. A. Smalyuk, J. A. Delettrez, D. D. Meyerhofer, T. C. Sangster, R. Betti, V. N. Goncharov, and D. Shvarts. Observation of self-similar behavior of the 3d, nonlinear rayleigh-taylor instability. In *Phys. Rev. Lett.* American Physical Society, Dec 2005.

O. San and K. Kara. Evaluation of riemann flux solvers for weno reconstruction schemes: Kelvin–helmholtz instability. *C. & F.*, 2015.

G. Scheuermann and X. Tricoche. Topological methods for flow visualization. In *The Visualization Handbook*, pages 341–356. 2005.

M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M. Hering-Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment Invariants for the Analysis of 2D Flow Fields. *IEEE TVCG*, 2007.

D. Schneider, J. Fuhrmann, W. Reich, and G. Scheuermann. A variance based FTLE-like method for unsteady uncertain vector fields. In *TopoInVis*, 2012.

G. Seisson, D. Hebert, I. Bertron, J.-M. Chevalier, L. Hallo, E. Lescoute, L. Videau, P. Combis, F. Guillet, M. Boustie, and L. Berthe. Dynamic cratering of graphite: Experimental results and simulations. *International Journal of Impact Engineering*, 63:18–28, 2014.

E. Shima and K. Kitamura. On new simple low-dissipation scheme of ausm-family for all speeds. In *AIAA Aerospace Sciences*, 2009.

N. Shivashankar and V. Natarajan. Parallel Computation of 3D Morse-Smale Complexes. 2012.

N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. P. Bos, and S. Rieder. Felix: A topology based framework for visual exploration of cosmic filaments. IEEE, 2016.

T. Sousbie. The persistent cosmic web and its filamentary structure: Theory and implementations. 2011.

M. L. Spaeth, K. Manes, D. Kalantar, P. Miller, J. Heebner, E. Bliss, D. Spec, T. Parham, P. Whitman, P. Wegner, et al. Description of the nif laser. In *Fusion Science and Technology*, volume 69, pages 25–145. Taylor & Francis, 2016.

P. Tabeling. Two-dimensional turbulence: a physicist approach. *Physics Reports*, 362:1–62, 2002.

S. Tarasov and M. Vyali. Construction of contour trees in 3d in o(n log n) steps. In *SoCG*, 1998.

Techviz. Techviz XL, https://www.techviz.net/en/.

J. Tierny. *Topological Data Analysis for Scientific Visualization*. Springer, 2018. ISBN 978-3-319-71507-0.

J. Tierny and V. Pascucci. Generalized topological simplification of scalar fields on surfaces. *IEEE TVCG*, 2012.

J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. 2009.

J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. In *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)*, 2017a. https://topology-tool-kit.github.io.

J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. 2017b. https://topology-tool-kit.github.io/.

J. Tierny, F. Vivodtzev, and C. Gueunet. Analyzing complex features in ensemble data with the topology toolkit in paraview. Kitware Blog, 2023. URL https://www.kitware.com/blog.

TOP500.org. TOP500 List - June 2025. https://www.top500.org/lists/top500/2025/06/, 2025. Accessed: August 11, 2025.

E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the hll-riemann solver. *Shock waves*, 4(1):25–34, 1994.

J. A. Trangenstein. *Numerical Solution of Hyperbolic Partial Differential Equations*. Cambridge University Press, 2007.

K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer. Fréchet Means for Distributions of Persistence Diagrams. *DCG*, 2014.

Unity Asset Transformer. Pixyz, https://www.pixyz-software.com/.

Unity Technologies. Unity, https://unity.com/.

Valve. Steam VR, https://store.steampowered.com/app/250820/SteamVR/.

J. Vidal, J. Budin, and J. Tierny. Progressive wasserstein barycenters of persistence diagrams. In *IEEE TVCG*, 2019.

Vive. HTC Vive Pro, https://www.vive.com/eu/product/vive-pro/.

F. Vivodtzev. *Hiérarchisation et visualisation multirésolution de résultats issus de codes de simulation*. Theses, Université Joseph-Fourier - Grenoble I, Dec. 2005.

F. Vivodtzev. Visualisation multirésolution de résultats de simulations. In *CHOCS avancées 2009, CEA*, 2009. doi: ISSN1961-7399.

F. Vivodtzev. Google Drive containing Benchmark Ensemble Data. https://drive.google.com/drive/folders/1CAxn4SkkU-u5cQD2bdPn4NNxwQMP7oa7?usp=sharing, 2022.

F. Vivodtzev and I. Bertron. Remote Visualization of Large Scale Fast Dynamic Simulations in a HPC Context. In *Proceedings of 2014 IEEE Symposium on Large Data Analysis and Visualization (LDAV 2014)*, Paris, France, November 2014.

F. Vivodtzev and T. Carrard. Open source software to visualize complex data on remote cea's supercomputing facilities. In *Proceedings of the Workshop on Open Source in Visualization at EUROVIS 2014 (short paper)*, Swansea, Wales, 2014.

F. Vivodtzev and M. Stauffert. Visualisation at CEA: HyperTreeGrid and other Physics. Presentation at the ParaView User Day Europe 2024, Focus session, ENS Lyon, France, Sept. 2024.

F. Vivodtzev, L. Linsen, G.-P. Bonneau, B. Hamann, K. Joy, and B. Olshausen. Hierarchical isosurface segmentation based on discrete curvature. In *5th Joint Eurographics-IEEE TCVG Symposium on Visualization (VisSym '03)*, Grenoble, France, May 2003.

F. Vivodtzev, G.-P. Bonneau, and P. Le Texier. Topology Preserving Simplification of 2D Non-Manifold Meshes with Embedded Structures. *The Visual Computer*, 21(8-10):679–688, Sept. 2005a.

F. Vivodtzev, L. Linsen, B. Hamann, K. I. Joy, and B. Olshausen. Brain mapping using topology graphs obtained by surface segmentation. In G.-P. Bonneau, T. Ertl, and G. M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*. Springer-Verlag, Heidelberg, Germany, 2005b. ISBN 3-540-26066-8.

F. Vivodtzev, D. Wiley, L. Linsen, J. Jones, N. Amenta, B. Hamann, and K. I. Joy. Automatic feature-based surface mapping for brain cortices. In R. F. Erbacher, J. C. Roberts, M. T. Groehn, K. Boerner, M. C. Hao, and P. C. Wong, editors, *Visualization and Data*

Analysis 2006, Bellingham, Washington, 2006. SPIE - The International Society for Optical Engineering.

F. Vivodtzev, G.-P. Bonneau, S. Hahmann, and H. Hagen. *Substructure Topology Preserving Simplification of Tetrahedral Meshes*, pages 55–66. Mathematics and Visualization. Springer, 2010. ISBN 978-3-642-15013-5. doi: 10.1007/978-3-642-15014-2_5.

F. Vivodtzev, A. Casner, L. Masse, L. Ceurvorst, S. Khan, and V. Smalyuk. Poster: Topological data analysis of 3d ablative rayleigh-taylor instability dataset for automatic segmentation. In *The 13th IEEE Symposium on Large Data Analysis and Visualization, LDAV 2023*, October 2023.

F. Vivodtzev, F. Nauleau, J. P. Braeunig, and J. Tierny. Identifying locally turbulent vortices within instabilities. In *2024 IEEE 14th Symposium on Large Data Analysis and Visualization (LDAV)*, page Poster, St. Pete Beach, Florida, USA, October 13–18 2024. URL https://arxiv.org/abs/2408.12662. In conjunction with IEEE VIS 2024; arXiv:2408.12662.

VRTK. VRTK SDK, https://www.vrtk.io/.

W. Wang, W. Wang, and S. Li. From numerics to combinatorics: a survey of topological methods for vector field visualization. 2016.

XR Tools. Godot, https://github.com/GodotVR/godot-xr-tools/releases.

L. Yan, T. B. Masood, R. Sridharamurthy, F. Rasheed, V. Natarajan, I. Hotz, and B. Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. 2021.

M. Zhang, S. Liu, H. Sun, W. Si, and Y. Qian. Hybrid vortex model for efficiently simulating turbulent smoke. In *ACM VRCAI*, 2014.

B. Zheng, B. Rieck, H. Leitte, and F. Sadlo. Visualization of Equivalence in 2D Bivariate Fields. *CGF*, 2019.

A. Zomorodian. *Computational topology*, page 3. Chapman & Hall/CRC, 2 edition, 2010. ISBN 9781584888208.