STAG: Smart Tools and Applications in Graphics (2025)
M. Comino Trinidad, C. Mancinelli, D. Cabiddu, and D. Giorgi (Editors)

A lightweight open-source tool for Meshing within Geosciences

M. Miola[1] and D. Cabiddu[1] and S. Pittaluga[1] and M. Raviola[2] and M. Vetuschi Zuccolini[2,1]

[1]Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche (CNR-IMATI), 16149 Genova, Italy
[2]Dipartimento di Scienze della Terra, dell'Ambiente e della Vita, University of Genova (DISTAV-UNIGE), 16132 Genova, Italy

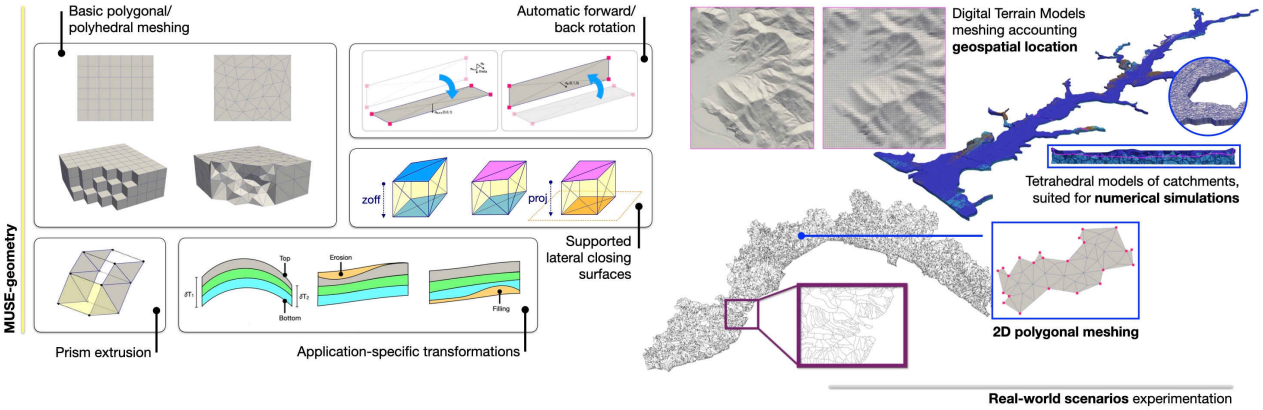

Figure 1: *Summary of* `MUSE-geometry` *main functionalities (left), including basic polygonal/polyhedra meshing, automatic forward/back data rotation according to their position in the space, types of lateral closing surfaces, triangle-to-prism extrusion, and application-specific transformations; some real-world scenarios in which the tool has been involved (right).*

Abstract

Understanding and modeling the complex geometries of the natural world is a key challenge in the Geosciences. Representing terrains, subsurfaces, catchments, and environmental domains requires not only accurate data acquisition but also robust and flexible geometric modeling tools. While computer graphics and geometry processing have made significant advances in representing and manipulating complex 2D and 3D shapes, their uptake in environmental modeling remains limited due to the lack of geoscientific constraints and interoperability with geospatial standards.

We present `MUSE-geometry`*, a lightweight computational geometry tool designed to bridge this gap. Our tool integrates geometric primitives, structured and unstructured meshing, and editing operations with explicit support for geospatial data formats and coordinate reference systems. By combining computational geometry with geospatial awareness, the tool enables the creation of simulation-ready 2D/3D models that preserve input integrity and topological consistency. Implemented as a self-contained command-line application in C++, our tool has been tested on real-world scenarios, including modeling topographic surfaces or coastal water volumes, demonstrating its ability to unify data structures and geometric processing within a single pipeline. The tool offers flexible and interoperable operations that enhance the integration of advanced geometric processing into geoscientific workflows in compliance with geospatial standards.*

CCS Concepts

• ***Computing methodologies*** → *Mesh geometry models;* • ***Applied computing*** → *Environmental sciences;*

1. Introduction

Modeling geometries of the natural world poses a challenge in the field of geosciences. Natural domains are inherently irregular, anisotropic, and multi-scale; they evolve under physical–chemical processes and are affected by morphological changes over time. Capturing these properties through geometric modeling is essential for improving our understanding of reality and generating sup-

DOI: 10.2312/stag.20251318

port for numerical simulations of complex phenomena whose spatial distribution and evolution over time are unknown.

Over the last few decades, the focus on advanced digital representation has gradually shifted towards more realistic targets, such as modeling terrain topography, subsoils, and catchments. Applied mathematics and computational geometry have provided powerful techniques for shape modeling that can be suitable for environmental contexts. Scientific studies increasingly highlight the demand for advanced, geoscientific 3D modeling technologies, demonstrating significant progress in landscape representation and surface-based modeling through computer graphics [AGSG25, GGP*19, CCLCdV*09]. These technologies are essential to analyze spatial structures and track the temporal evolution of landscapes shaped by environmental processes [CGEY25, CC24, HMPP19]. Combining these needs and methods has promoted a growing collaboration between two scientific communities, i.e., Computer Graphics and Earth Sciences, encouraging interdisciplinary approaches.

Geometric tools in Computer Graphics have surely enabled the detailed representation of complex surface morphologies [TG07, NLP*13, ZBGS15]. They also support fine-scale modeling using segmentation techniques, coupled with refinement and adaptive meshing, which enhances pattern detection and improves computational efficiency for subsequent numerical simulations [SMP*25]. However, integrating these methodologies into geoscientific computational workflows is challenging: in some cases, they lack physical constraints, typically ignore geospatial standards, and are not designed to manage large-scale, multivariate datasets efficiently.

Geoscientific modeling, on the other hand, is anchored to consolidated Geographic Information Systems (GIS) standards. GIS offer well-established functionality for storing, visualizing, and analyzing geographically located data by producing maps, digital images, and tables [BC94]. Their strengths lie in 2D operations such as cartography, spatial querying, and basic geoprocessing (e.g., buffering/offsetting, clipping, intersections), which are widely used and highly interoperable. However, most geoscientific modeling still relies on regular grids or block models (i.e., pixel-based), which are computationally efficient but poorly capture faults and discontinuities. This simplification produces artifacts such as stair-stepped surfaces, disrupted connectivity, and inadequate representation of subsurface heterogeneity. 3D support in GIS is basic and is usually limited to simple triangulations. Advanced mesh processing, topological consistency, or local geometry editing are generally unsupported. Both open-source and proprietary GIS can generate surface models from point clouds, commonly via Delaunay triangulation. However, they have limited direct mesh editing functionality and are affected by interoperability issues and weak standardization, due to varying implementation approaches.

While exporting triangulated models to standard computer graphics formats is partially supported, importing pre-constructed meshes is possible but constrained by format incompatibilities and limited support for topological editing. As a result, GIS platforms mostly provide pseudo-3D representations, which often lead to distortions and loss of structural detail. Although some functionalities to manage 3D spatial data are supported by third-party plugins, GIS are not designed for advanced modeling and customization of meshes. Therefore, employing external software is required.

To bridge this gap, we present `MUSE-geometry`, a lightweight and self-contained computational geometry module designed to support the needs of geoscientific modeling. It offers a set of geometric primitives and operations suited to irregular, multi-scale, and often incomplete data that characterize geoscientific applications. Unlike general-purpose geometry libraries, it prioritizes topological consistency and interoperability with common geodata formats and reference systems, enabling seamless integration into geospatial data-driven workflows. While `MUSE-geometry` builds on well-established meshing and geometry processing approaches (e.g., Delaunay triangulation, tetrahedralization, and simple extrusion), its novelty does not lie in the design of fundamentally new algorithms. Instead, the contribution of this work is the integration and adaptation of existing methods into a lightweight, easy-to-use, geoscientifically oriented software framework. `MUSE-geometry` unifies geometric processing, explicit coordinate reference systems handling, and interoperability with geospatial formats in a single tool, as illustrated in Figure 1. This integration fulfills the research demand, providing domain experts with a coherent tool tailored to their workflows.

`MUSE-geometry` is developed as a module of the `MUSE` (Modeling Uncertainty as a Support for Environment) [Mio25, MCPVZ22], a stochastic framework designed to model spatial uncertainty in environmental monitoring and simulation processes, from which it also derives its name.

The paper is organized as follows. Section 2 provides an overview of state-of-the-art tools and frameworks in both Computer Graphics and Geosciences. Section 3 introduces the definitions and terminology adopted throughout the paper. Section 4 presents `MUSE-geometry`, describing its features and functionalities. Section 5 discusses the experimental results obtained by applying `MUSE-geometry` to real environmental scenarios of varying complexity and requirements. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Works

In the following sections, we present an overview of the most common geometry processing tools and their implementations in geoscientific modeling. We categorize and discuss existing tools and frameworks into three macro-groups: *general-purpose geometry libraries* (Section 2.1), *GIS-oriented frameworks* (Section 2.2), and *domain-specific geoscience tools* (Section 2.3).

2.1. General-purpose geometry libraries

General-purpose geometry libraries provide robust functionalities for mesh generation and manipulation and are widely used in computer graphics, CAD, and scientific computing. Their main focus is on simplicial structures (i.e., triangles in 2D and tetrahedra in 3D) which form the standard primitives for most algorithms in computational geometry and graphics. While this representation is well suited for visualization and numerical simulation, it overlooks grid-based discretizations (regular, rectilinear, or curvilinear), which are predominant in geoscientific applications for representing fields such as topography, climate variables, or subsurface properties. In

recent years, however, software tools have increasingly incorporated support for alternative discretizations such as hexahedral and polyhedral meshes, which are better suited for certain simulation scenarios due to their improved numerical properties and flexibility in representing complex domains.

Representative examples include *CGAL* [CGA24], *cinolib* [Liv19], *geogram* [Lé25], *ImatiSTL* [Att17], *libigl* [J*23], and *VCGlib* [VCG24], which offer rich toolsets for surface and volumetric mesh processing and remeshing. In parallel, mesh generators such as *Triangle* [She96], *TetGen* [Si24], *TetWild* [HZG*18], and their extensions (*triWild* [HSG*19], *fTetWild* [HSW*20]) demonstrate the-state-of-the-art for simplicial meshes. Beyond simplicial structures, tools such as *Gmsh* [GR09] and *Cubit* [Cor] provide dedicated support for hexahedral and hex-dominant meshing, while frameworks like *PolyMesher* [TPMP12], *OpenFOAM* [Ope25], and the *SALOME Platform* [EDF25] offer capabilities for polyhedral meshing. However, both categories are often developed for domain-specific purposes (Section 2.3).

Despite their maturity, the application of these libraries in geosciences is not straightforward. They are designed to operate on clean, manifold geometries and to treat models in abstract Euclidean space, while geoscientific data are often noisy, incomplete, and characterized by strong multi-scale variability. Moreover, these tools rarely incorporate concepts of geospatial information such as coordinate reference systems, datum transformations, or metadata management, which are essential for integrating geometries within real-world datasets. Only a few libraries, such as *Boost.Geometry* [Dev24] and the *Point Cloud Library (PCL)* [Poi24], provide some degree of geospatial awareness, but their scope is limited and insufficient to cover the full complexity of geoscientific workflows.

2.2. GIS-oriented frameworks

GIS-oriented frameworks provide extensive geospatial functionalities that general-purpose geometry libraries often lack. Widely used GIS libraries and engines include desktop applications, full GIS systems, and programming libraries, each with varying levels of support for data management, geoprocessing, and visualization. In the following, we mention the most popular GIS-oriented libraries and tools, highlighting the main features and compatibility with computer graphics.

GDAL/OGR [GDA22] is the de facto standard library for geospatial data I/O and raster/vector management. It provides robust coordinate reference system (CRS) support and integrates with *PROJ* [PRO22] for coordinate transformations, but it offers no native visualization or advanced geometric processing functionalities. Lightweight library solutions such as *shapelib* [OSG25] are specified for managing I/O vector data and lack both geoprocessing and visualization capabilities. More comprehensive frameworks like *TerraLib* [CVF*08] provide core geospatial processing for raster and vector data, including coordinate transformations, but similarly fall short in terms of advanced visualization or general-purpose geometric operations. Integrating *GDAL* and *PROJ* is essential for GIS engines. Open source examples include *GRASS GIS* [GRA22] and *QGIS* [QGI24], while commercial platforms such as *ArcGIS* [All11] support many GDAL-compatible formats and coordinate systems through their internal implementations. They offer comprehensive geoprocessing tools, advanced management of georeferenced coordinate systems, and basic visualization features.

While these tools excel in managing complex geospatial workflows, their geometric processing capabilities are limited to simple triangulations or basic 2D/3D rendering, and they often rely on external libraries for mesh generation. The *MDAL* [MDA20] library is the clearest example of an attempt to integrate meshing within GIS. Just as *GDAL* offers a layer of abstraction for geospatial data, *MDAL* provides a data model for representing unstructured meshes, regardless of their storage format, but lacks geometry processing capabilities. Geospatial file support is only possible indirectly via *GDAL*, making it insufficient for computer graphics requirements.

In summary, the long tradition in using GIS-oriented frameworks demonstrates strong capabilities in geospatial data management and geoprocessing, but rarely integrates advanced geometric operations, mesh generation, and visualization within a lightweight and unified environment.

2.3. Domain-specific frameworks

Several domain-specific frameworks have been developed to support applications in hydrology, geology, geophysics, and environmental modeling. Unlike generic libraries, they are designed around the requirements of specific scientific and industrial communities, and therefore provide tools tailored for representing and simulating natural phenomena.

A notable example is *Gmsh* [GR09], which is widely used for finite element simulations and appreciated for its flexible meshing capabilities and powerful scripting interface. Other frameworks explicitly developed for geoscientific purposes include *LaGriT* [GMHH25], targeting hydrological and geological simulations, and *Cubit* (now commercialized as Coreform Cubit) [Cor], which provides advanced support for complex 3D meshes in geophysics and subsurface modeling. Recent toolkits such as *SimPEG* [CKH*15], with its *discretize* meshing component [dev25], and the *DMPlex* module in PETSc [Tea25], extend mesh handling to large-scale geophysical simulations. More specialized research efforts include the *RINGMesh* library [PCJ*17] and *PZero* software [MBB*24], designed for 3D geological modeling workflows, as well as 3D meshing generator–based frameworks such as [XL14] for reservoir modeling, [MCP*22], which integrates heterogeneous geoscientific data into consistent 3D models, and *MeshIt* [CB15], a robust open-source tool for meshing complex and fractured geological models, although it is limited to tetrahedral meshing of convex domains.

Belonging to the same category, but aiming for broader coverage, are comprehensive frameworks that provide end-to-end environments, from data import and interpretation to the generation of models ready for numerical simulations. Notable examples include the commercial software *Leapfrog* [lea], widely used in geological and mining contexts for implicit surface and volume modeling via interpolation of survey data and field observations. Similarly, *GOCAD/SKUA* [Mal92] has become a standard for structural geology and sedimentary basin modeling, while *GeoModeller* [BG25] allows the construction of volumetric models including lithologies and faults, integrating heterogeneous data such as

cross-sections, boreholes, geophysical, and hydrogeological surveys. *Petrel* [Sch25], developed for the oil and gas industry, offers integrated functionalities for seismic interpretation, reservoir characterization, and simulation.

Overall, domain-specific frameworks span a continuum from research-oriented tools, offering algorithmic flexibility and open experimentation, to industrial solutions focused on robustness and complete workflow integration. While highly effective in their respective domains, they reveal the lack of a unifying framework that can combine advanced geometry processing, geospatial awareness, and data heterogeneity.

3. Background and Terminology

Geoscientific data differs in many ways from the geometric data commonly encountered in computer graphics. This section is intended to provide some background knowledge and terminology necessary to better understand the methodology proposed in this paper. We recognize the following key aspects, i.e., (1) data allocation in the space (georeferencing), (2) data types, (3) meshing, and (4) specialized transformations.

A first distinction between the two worlds lies in how positions are represented. Instead of local Cartesian coordinates, geospatial datasets are usually expressed in *geographic coordinates*, such as latitude, longitude, and elevation, or in projected *coordinate reference systems* (CRS) that map the curved surface of the Earth onto a plane. Overlaying different geographic datasets requires specifying the *EPSG* (European Petroleum Survey Group) code, which uniquely identifies a standard CRS. Transformations between these systems are frequent and essential, since working directly with geographic coordinates can lead to numerical instabilities in geometry processing due to the very large or highly precise values involved.

Another characteristic feature of geoscientific data is the variety of formats used to represent the environment. *Raster data*, for instance, encodes information on a regular grid, where each cell contains a value, e.g., elevation, as in the case of Digital Elevation Models (DEM). *Vector data* instead represents discrete geometries, such as points (e.g., measurement stations), lines (e.g., rivers), and polygons (e.g., terrain boundaries). Beyond raster and vector, point clouds acquired through laser scanning technologies or photogrammetry are increasingly important for capturing detailed 3D information about surfaces and vegetation, but they lack connectivity and must often be transformed into meshes before they can be used in simulations.

Meshing plays a fundamental role in geosciences. Following the terminology commonly adopted by geoscientists, *structured meshes*, consisting of regular grids of cells, are widely used for numerical models where uniform discretization is sufficient. However, natural domains are rarely regular: terrains, subsurfaces, and geological formations exhibit complex geometries that require *unstructured meshes*, such as triangular or tetrahedral discretizations, to capture their variability [MCM*24]. Constructing such meshes while respecting structural constraints is often a key bottleneck [CCLCdV*09].

Finally, certain operations are specific to geoscientific applications and have no direct counterpart in computer graphics. One example is the *stratigraphic transformation* [LB22, RKCC15, McL04], used in geostatistics modeling for faulted and folded domains. The points representing the physical model, which includes structural and geological characteristics of the actual deposit, are re-parameterized in a coordinate system aligned with geological layers rather than with the global Cartesian frame. A simple unfolding operation of the model, acting only on points' z-coordinate, allows spatial continuity modeling to be performed while preserving stratigraphic continuity, which would be intractable in the original coordinates.

4. `MUSE-geometry`

Where general-purpose geometry libraries lack geospatial awareness, and GIS frameworks provide limited geometric flexibility, `MUSE-geometry` combines the strengths of both domains in a single lightweight computational tool. Its core objective is to enable the generation and processing of 2D and 3D meshes directly from geospatial datasets while preserving their reference systems and geometric quality. Rather than introducing new meshing algorithms, the contribution of `MUSE-geometry` lies in the integration and adaptation of established methods within a geoscientifically oriented, geospatially aware framework. This integration provides domain experts with a coherent and accessible tool that bridges the gap between computational geometry and environmental modeling workflows. The tool is not limited to mesh generation. It also provides a suite of functionalities for geometric editing and manipulation, enabling users to perform common operations such as point projection, surface interpolation, and application-specific transformations (i.e., stratigraphic coordinates transformations for geostatistics computation).

The complete list of functionalities, ranging from data processing and meshing, is described in Section 4.1. `MUSE-geometry` is implemented as a command-line tool to facilitate integration into automated pipelines and large-scale data processing workflows. It supports a variety of input formats commonly used in GIS and environmental modeling, and is designed to be modular, extensible, and interoperable with existing open-source tools and GIS (Section 4.2).

4.1. Main Functionalities

`MUSE-geometry` consists of a suite of geometric functionalities for geospatial data processing and mesh generation. It enables streamlined I/O operations with explicit CRS handling, ensuring accurate format conversion while preserving geometry integrity and attribute consistency (Section 4.1.1). The tool supports both structured and unstructured meshing for surface and volume modeling, including algorithms for triangulation and, more generally, polygonal meshing, local and adaptive refinement, tetrahedral meshing, as well as prism-based representation (Section 4.1.2). Lightweight operations, such as splitting, offsets, translation, scaling, and rotation, are integrated to allow geometric editing and transformation of meshes, to generate watertight volumetric domains (Section 4.1.3). Finally, application-specific operations, such as stratigraphic transformation, enable real-time use for spatial prediction in geoscientific contexts (Section 4.1.4).

4.1.1. Input/Output and CRS management

Built on *GDAL* [GDA22] and *PROJ* [PRO22], the tool supports I/O of common vector (e.g., Shapefile, Geopackage, GeoJSON) and raster formats (e.g., GeoTIFF, ASCII Grid), as well as point clouds.

Georeferencing is preserved through explicit CRS management, allowing conversion between coordinate systems by specifying the appropriate EPSG code. When reading geospatial data, the tool reports the georeferencing information, indicating whether the coordinates are geographic (latitude/longitude) or in a projected metric system. The latter is suitable for subsequent geometric processing. In the former case, special attention must be paid to the units of measurement: the tool returns an error, prompting the user to convert the data from degrees to metric coordinates before meshing. Conversion can be performed directly by `MUSE-geometry` itself by specifying the target EPSG code, enabling automatic transformation of geographic coordinates into a metric system suitable for meshing.

The content of vector files is extracted according to the geometry type, such as (multi-) points, (multi-) lines, (multi-) polygons, supported by *GDAL*. Associated attributes can be summarized in metadata or preserved in separate tabular files, such as CSV, if they are available. Raster files, on the other hand, are approached as a point cloud, where georeferenced points correspond to the centroids of each pixel that builds the map.

Collecting environmental data from diverse sources requires careful verification to ensure compatibility, validate georeferencing, and confirm that the data belong to the correct computational domain. When `MUSE-geometry` checks the reference system, the integration of heterogeneous datasets is facilitated through the implementation of *point-in-polygon* functions [Fra70, J*23], ensuring proper alignment within the computational domain.

4.1.2. Structured and unstructured meshing

`MUSE-geometry` supports both structured grid generation (regular subdivision of a bounding box possibly constrained by arbitrary boundaries) and unstructured meshing based on Delaunay triangulation and tetrahedralization [She96, Si24]. Figure 2 shows examples of 2D/2.5D meshing strategies implemented in `MUSE-geometry`.

The code is inherently aware that geospatial and computer graphics representations differ. For example, *geospatial* polygons are explicitly closed by duplicating the first vertex, whereas in formats typically used in computational geometry, vertices are usually not duplicated. To avoid compromising mesh computation due to duplicates, the tool handles such situations to ensure the success of the meshing algorithm.

Structured grids are generated by a simple approach, dividing the bounding box of a specific domain into a discrete number of 2D equal-dimension cells (or volumetric cells in 3D) and then adding iterative cells and vertices to compose the grid.

Unstructured meshing relies on advanced triangulation algorithms that support various boundary conditions (e.g., convex hull, concave hull [Ada19], or user-defined boundaries) and allow polygonal mesh generation. `MUSE-geometry` enables triangulation of any surface that can be approximated as a plane. In practice, vertical or oblique sections (such as a geological cross-section) are handled by first rotating the section into a horizontal plane, performing the triangulation with standard algorithms, and then back-rotating the resulting mesh to restore the original orientation. This process is fully transparent to the user, as it is executed automatically through the rotation functionality implemented in `MUSE-geometry` (Section 4.1.3).

For volumetric modeling, structured meshing naturally extends from the 2D case. Unstructured meshing, in contrast, follows the computational pipeline introduced in [MCP*22], with extensions to handle heterogeneous input data, enhance algorithmic robustness, and enforce mesh quality constraints (e.g., adaptive refinement or application-specific geometric criteria such as minimum angles or maximum cell areas). If a triangulated surface is provided as input, `MUSE-geometry` generates the volumetric domain following the methodology described in Section 4.1.3. If the input consists of vector or point cloud data, the procedure is preceded by triangulation via *Triangle* [She96]. The final tetrahedralization is then performed using *TetGen* [Si24], possibly setting quality parameters, producing a high-quality volumetric mesh suitable for subsequent simulations.

4.1.3. Geometric editing and transformations

Lightweight geometric editing tools include:

- local operations on surface/volume, such as poly/edge splitting, surface offsets (i.e., shifting vertices along surface normal) of a constant quantity or to an absolute elevation, translations, scaling, rotations, z-values filtering, extrusion to prism-based representation (Figure 5);
- coupling surfaces, such as merging in a single model, edge matching/boundary alignment (implicit in merging operation).

Rotation is a key feature in this context and serves a dual purpose. It can be applied explicitly to align data with a specific plane in the space, or it can be handled implicitly as a preprocessing step before triangulation. The latter is essential in geological modeling, where data often come from cross-sections, oriented on an arbitrary plane (i.e., not parallel to the X-Y plane). If loaded in their original reference system, such geometries are not compatible with standard triangulation algorithms, which typically operate in the X-Y plane. The tool addresses this automatically by checking the orientation of the input data and applying a rotation when necessary to ensure proper alignment with *triangulation* space (Figure 3).

The rotation angle θ is computed between the normal to the best-plane fitting point ($\boldsymbol{n}_p$) and the normal to the X–Y plane ($\boldsymbol{n}_{xy} = (0,0,1)$); the rotation axis is detected by computing the cross product between the two normal vectors; the rotation center is set to the center of the bounding box of points. The rotation parameters are defined as

$$\theta = \arccos\left(\frac{\boldsymbol{n}_p \cdot \boldsymbol{n}_{xy}}{\|\boldsymbol{n}_p\| \, \|\boldsymbol{n}_{xy}\|}\right), \tag{1}$$

$$\boldsymbol{u} = \frac{\boldsymbol{n}_p \times \boldsymbol{n}_{xy}}{\|\boldsymbol{n}_p \times \boldsymbol{n}_{xy}\|}, \tag{2}$$

where θ is the rotation angle and $\boldsymbol{u}$ is the unit rotation axis.

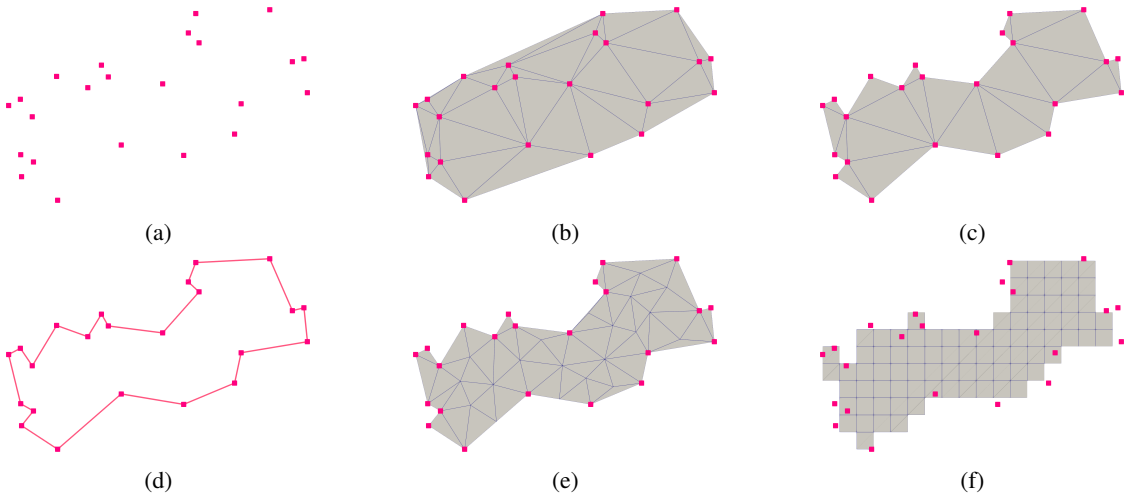

Figure 2: *From vector geospatial data (GeoPackage) to 2.5D surface models under different boundary conditions: (a) input points, with derived models (b) triangulation with convex hull and (c) triangulation with concave hull; (d) input polygon, with derived models (e) boundary-constrained triangulation with refinement and (f) regular grid mesh. Vertex coordinates of the 2.5D models (x, y, z) are consistent with point coordinates in the input geospatial data.*

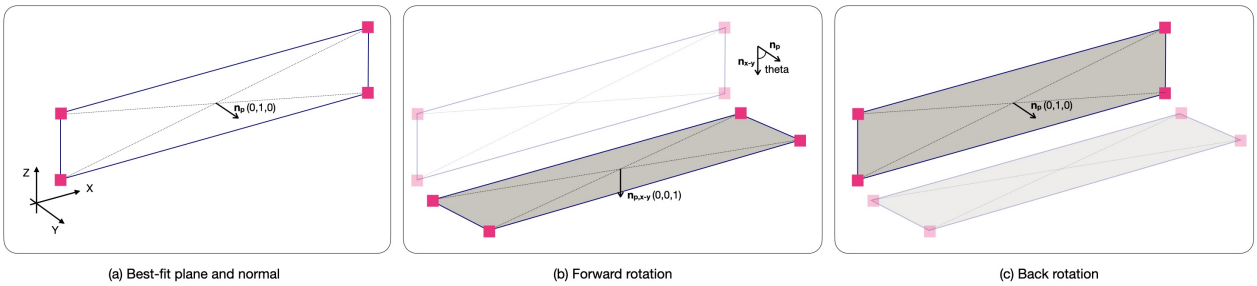

Figure 3: *Stepped procedure to check the orientation of points allocation before starting triangulation: (a) best-plane fitting points and related normal, (b) checking normal respect to the horizontal plane, computing the angle between the two normal vectors for the points' forward rotation, (c) back rotation to restore the original reference system.*

Another key feature of `MUSE-geometry` is the ability to generate volumetric domains from 2D surfaces using a variety of methodologies. One approach involves creating surfaces that are exact copies of each other but vertically offset, which is illustrated in Figure 4a and is particularly useful for representing layered structures. A second methodology addresses irregular surfaces at arbitrary elevations, allowing the modeling of complex, non-uniform geometries, as shown in Figure 4b. A third approach combines an irregular surface with a plane defined by a set of points, enabling the closure of open or incomplete surfaces, as depicted in Figure 4c.

In addition, `MUSE-geometry` supports extensions for prism-based meshes, allowing a 2D triangulation (such as a geological cross-section) to be extruded into a 3D prism mesh by offsetting the triangular elements along a specified direction (Figure 5).

Overall, these functionalities demonstrate how `MUSE-geometry` provides a comprehensive set of lightweight geometric editing and mesh generation tools. From local operations on surfaces and volumes, such as splitting, offsets, transformations, and extrusion, to the coupling and alignment of multiple surfaces, the software ensures robust handling of complex geometries. Rotation and alignment procedures guarantee compatibility with triangulation algorithms, while volumetric domain construction and prism-based mesh extrusion support the creation of 3D models suitable for subsequent numerical simulations.

4.1.4. Application-specific transformations

A distinctive feature of `MUSE-geometry` is its support for stratigraphic transformations used in geostatistics. These are crucial when dealing with spatial modeling of non-planar stratified geological bodies for accurate estimation [McL04, RKCC15, Boi10, LB22]. This approach is an integral part of geostatistics workflows, aiming at predicting the spatial distribution of environmental variables, collected in discrete points in the space, by a stochastic-based algorithm [GHS21].

Firstly, we model the physical deposit by meshing, proposing

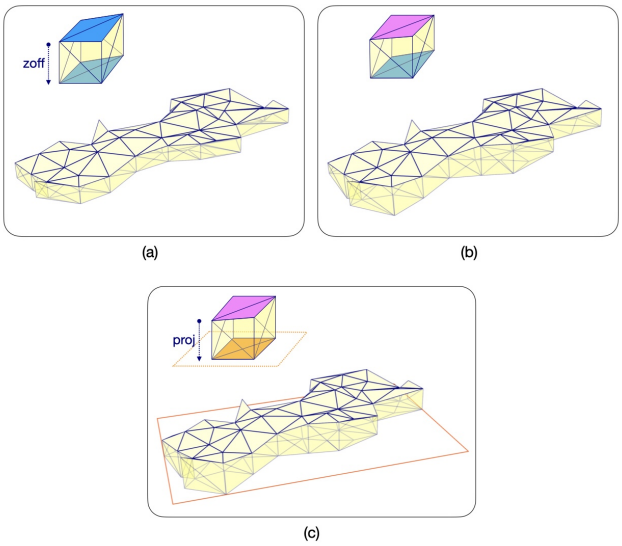

Figure 4: *Different scenarios supported for lateral closing surface meshes into a watertight object, useful for subsequent tetrahedral processing. The (a) scheme indicates the closure between a triangle mesh and its shifted duplicate at an arbitrary vertical elevation; the (b) scheme represents the coupling of multiple irregular triangular surfaces; the (c) scheme illustrates the closure between an irregular triangle surface and the triangulation of mesh vertices on an arbitrary plane.*

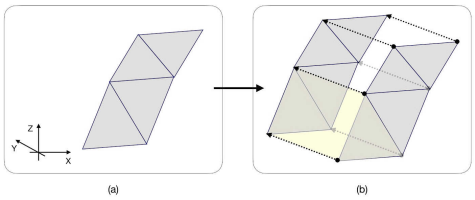

Figure 5: *From (a) 2D to (b) 3D model through prism generation.*

the employment of triangular/tetrahedral grids. Then, we apply the coordinate transformation on mesh vertices, which consists of removing the models' distortive effect and consistently re-allocating sampled data in the stratigraphic coordinate system. They are projected onto reference surfaces and remapped into stratigraphic coordinates, enabling coherent variogram modeling and stochastic simulations computation.

Despite developments in the use of meshes in reservoir modelling, existing approaches remain anchored to the use of traditional block grids [BCBC08, PD14, LB22] . Although existing software, such as GOCAD [Mal92], supports stratigraphic grid generation or deformation handling, it does not provide an open-source and ready-to-use implementation that performs explicit stratigraphic coordinate transformations on both 2D/3D structured and unstructured meshes. `MUSE-geometry` addresses this gap by offering a unified, open-source framework for such transformations, to enable direct geostatistics modeling.

The tool currently supports three transformation types, corresponding to typical reservoir scenarios: *proportional*, *truncation*, and *onlap* [PD14]. In the proportional case (Figure 6-1a), the coordinate z_p of a generic point P is scaled relative to the bounding surfaces and mapped to a stratigraphic coordinate $z_s = (z_p - z_b)/(z_t - z_b) \cdot \delta T$, where z_t and z_b denote the top and bottom of the stratigraphic interval, and δT is the average thickness of the interval, computed across all points or along a reference profile. This parameter normalizes the stratigraphic coordinate so that variations in interval thickness across the domain do not distort the transformed coordinates.

For *truncation*, the transformation references the lower bounding surface, which remains intact, with respect to the eroded top layer (i.e., the orange top layer in Figure 6-1b) as $z_s = z_p - z_b$.

For *onlap*, the upper bounding surface serves as a reference, having a filling at the bottom (i.e., the orange bottom layer in Figure 6-1c) as $z_s = z_t - z_p$.

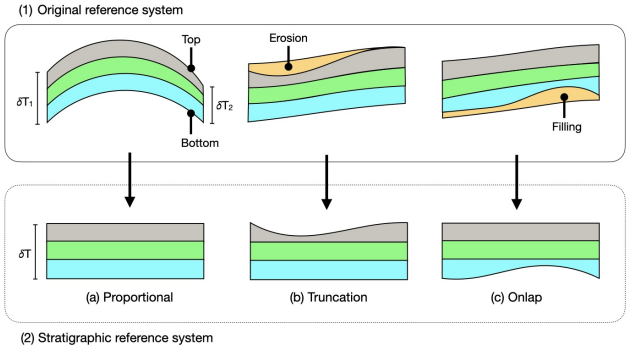

Figure 6: *Examples of (1) different stratigraphic scenarios derived from reservoir modeling and the (2) relative transformation into the stratigraphic coordinate system: (a) proportional, (b) truncation, and (c) onlap.*

4.2. Principles in Practice

`MUSE-geometry` has been designed from the ground up to embody its three guiding principles in every aspect of its functionalities.

Numerical robustness. Environmental and geospatial data often involve coordinates with values near machine precision, which can easily destabilize standard meshing algorithms. To ensure stability, all meshing operations in `MUSE-geometry` are performed in local Cartesian frames, and coordinates are restored to their original georeferenced system after processing. This approach is consistently applied across both structured and unstructured meshing routines, enabling robust handling of large and complex datasets without requiring user intervention.

Flexibility. The framework supports multiple mesh types, adaptive refinement, prism-based meshes, and a variety of geometric editing operations, such as splitting, offsetting, translations, scaling, rotations, and enabling watertight volumetric domains. Similarly,

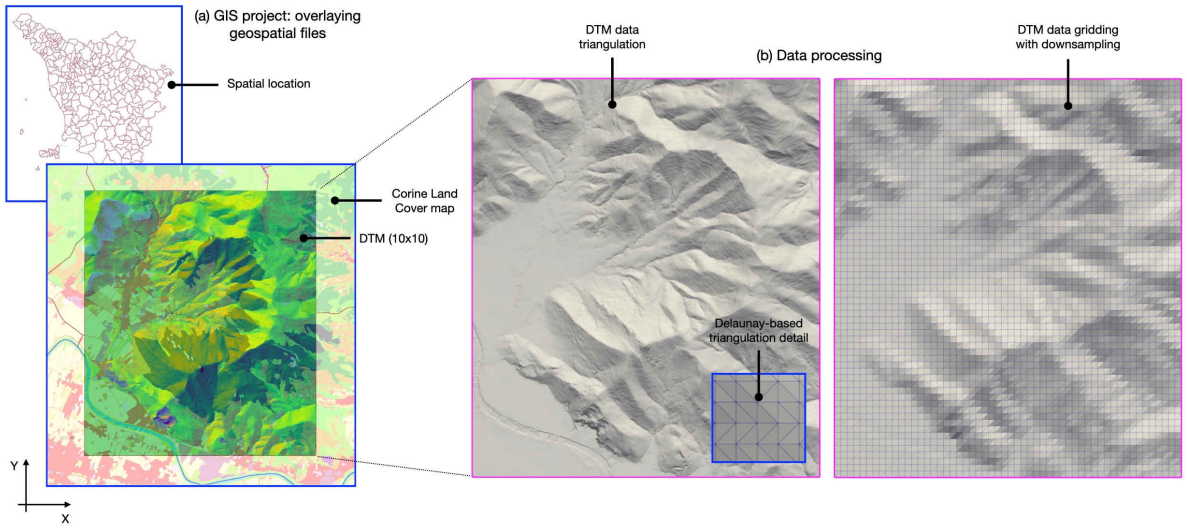

Figure 7: *From a raster DTM with an original resolution of* 10×10 *m (GeoTiff standard format) in (a) GIS project to (b) data processing by* `MUSE-geometry`*: Delaunay-based triangulation of points and gridding with down-sampling (i.e., final resolution of* 100×100 *m).*

application-specific transformations such as stratigraphic mappings are supported. By providing multiple options for mesh generation, editing, and transformation, `MUSE-geometry` can be adapted to a wide range of geoscientific modeling scenarios. Additionally, its remarkable flexibility makes it suitable for generic, non-geological domains, extending its applicability beyond geoscientific use cases.

Interoperability. `MUSE-geometry` integrates seamlessly with widely adopted geospatial formats through *GDAL* and *PROJ*, preserving both geometric and attribute information. Exports can be directed to geoscientific workflows, computer graphics applications, or simplified tabular formats for attribute-only use. Moreover, application-specific transformations and meshing operations are designed to work directly with existing mesh structures and geospatial datasets, enabling the framework to fit naturally into broader workflows without additional preprocessing or conversion.

By consistently applying these principles across all functionalities, `MUSE-geometry` ensures that geospatial meshing and transformations are robust, adaptable, and compatible with real-world applications.

5. Results & Discussion

`MUSE-geometry` is released as an open-source standalone software, designed for a large community of users, from those with low programming expertise to expert developers. It is developed in C++ and it is available in a GitHub public repository (`https://github.com/mariannamiola/MUSEgeom.git`) under the GPLv3 licence. It has been tested on Windows, Linux, and macOS. Specific instructions for cloning, external dependencies, building the source code, and using the tool are provided in the repository.

In the following Section 5.1, we illustrate the application of `MUSE-geometry` to real case studies, selected to cover a range of complexities. Each example is identified by specific keywords that represent the geometry processing operations involved, while the limitations of the current version of `MUSE-geometry` are discussed in Section 5.2.

5.1. Real Case Studies

To evaluate the effectiveness of `MUSE-geometry`, we conducted a series of experiments on real-world geoscientific datasets. The selected use cases span a variety of domains, including terrain modeling and subsurface or water-body representation, each posing different requirements on the resulting geometric models. In some cases, the focus is on accurately capturing surface morphology and boundaries, while in others the objective is to generate volumetric meshes suitable for numerical simulation. By applying `MUSE-geometry` across these heterogeneous scenarios, we demonstrate both the versatility of our approach and its ability to integrate seamlessly into geoscientific workflows. In the following paragraphs, we present examples of geoscientific workflows in which `MUSE-geometry` has been employed to generate 2D and 3D meshes. The results were validated by experts from different geoscientific domains, who confirmed that `MUSE-geometry` simplifies mesh generation and simulation activities effectively benefit from the meshes produced by `MUSE-geometry`.

Raster to mesh This application involves the pure application of geometry processing techniques on standard raster data, preserving georeferencing and its native features. We process a publicly available real-world Digital Terrain Model (DTM) from the Tuscany Region portal [ope], covering the province of Pisa. The dataset, originally in ASCII-Grid format in Gauss-Boaga coordinate reference system (EPSG:3003), was preliminarily processed in QGIS to extract a smaller, more easily manageable area. The selected portion covers the Monte Pisano site and maintains the original pixel

size (resolution) of 10×10 meters. This dataset is used to evaluate raster loading (Geotiff format) and geometric processing, including Delaunay-based triangulation (Figure 7a) and gridding based on original pixel size and varying resolution with down-sampling (Figure 7b-c). The DTM is remapped by choosing a lower resolution (larger pixel size). A new elevation value will be associated with the resized pixels, equal to the average elevation of the original cells that fall within the new cell.

Vector to mesh This application involves the pure application of geometry processing techniques on standard vector data, preserving georeferencing and its native features, and automatically solving the inconsistency between GIS-based formats and Computer Graphics. As an example, Figure 8 shows the result obtained by processing a vector file covering the Liguria region and dividing it into slope units, i.e., polygons that delimit areas with homogeneous geomorphological properties (i.e slope). These slope units are publicly available from the IRPI-CNR Geomorphology Research Group [AGM20], which provides a dataset and tools for generating slope units across Italy. In this case, the vector file is converted into a polygon mesh, where each cell corresponds to a detected slope unit and retains its original, potentially irregular, shape.

Figure 8: *Example of a polygonal mesh obtained by processing a vector standard file, composed of a series of polygons representing units with homogeneous geological properties.*

Geological section modeling This application represents a traditional geological workflow, based on the reconstruction of a heterogeneous vertical geological section from a sampled dataset (e.g., lithology, porosity, permeability). First, the example tests the automatic detection of the spatial arrangement of points and their rotation onto the X–Y plane to satisfy the requirements of the triangulation algorithm. The resulting 2D model is then used for stochastic simulation, with an example output shown in Figure 9a. When combined with heterogeneity information, this model can be used to perform flow and transport simulations, such as tracer dispersion in a heterogeneous porous medium. Extensions for prism-based meshes is applied to satisfy constraints on mesh structure, such as cell type or orientation, required by external tools. In particular, the 2D triangular input mesh is extruded into a 3D prism mesh by offsetting the triangular elements along a specified direction (Figure 9b), enabling geochemical modeling in `PFLOTRAN` [LHL*20] (Figure 9c).

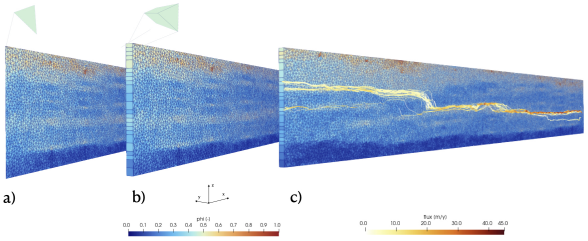

Figure 9: *Stochastic modeling of a heterogeneous porous medium on a high-resolution unstructured mesh support, varying from (a) triangulation to (b) prism-based representation for PFLOTRAN software compatibility, aiming at (c) flow and transport numerical modeling of a tracer dispersion.*

Water volume modeling This application compares geometry processing techniques (structured vs. unstructured meshes) for modeling coastal surface water volumes, with the final aim to provide a 3D mesh to support real-time monitoring activities. Two datasets are employed, representing the harbors of Genoa and Portofino (Liguria, Italy). Each dataset includes (1) high-density bathymetric data and (2) the boundary delineating a portion of the corresponding water volume. The 3D volume meshing workflow involves several computational steps, including the 2.5D triangulation of the bathymetry, offsetting to represent the water surface, lateral closure (see Figure 4b), and final tetrahedralization in the case of unstructured meshing. Figure 10 illustrates the results obtained for the Portofino harbor, comparing structured meshing (regular gridding with a $20 \times 20 \times 20$ resolution) and unstructured meshing (tetrahedralization), both constrained by the harbor boundary defining the modeled water volume.

Geostatistics application This application demonstrates how geometry processing tools are integrated into computational workflows to support geostatistical and stochastic modeling. Figure 11 illustrates the volume tetrahedral model of the Polcevera river basin (Liguria, Italy), starting from the triangulation of a Digital Elevation Model constrained to the basin boundary, which is highly anisotropic. The model is built by combining DEM-constrained triangulation with the projection of mesh vertices onto an arbitrary plane, following the scheme presented in Figure 4c. The resulting mesh serves as a geometrical framework for stochastic simulations to estimate the spatial distribution of sparsely sampled lithologies (e.g., filling, rock, and sediment), while accounting for spatial covariance laws. In this context, the correctness of spatial covariance is ensured by applying a stratigraphic coordinates transformation. Figure 11b shows the tetrahedral mesh before (gray) and after (yellow, "flat") the stratigraphic transformation. Figure 11c presents the results of stochastic simulations, highlighting the utility of this approach for geological analyses, such as identifying interfaces between rock and sediment.

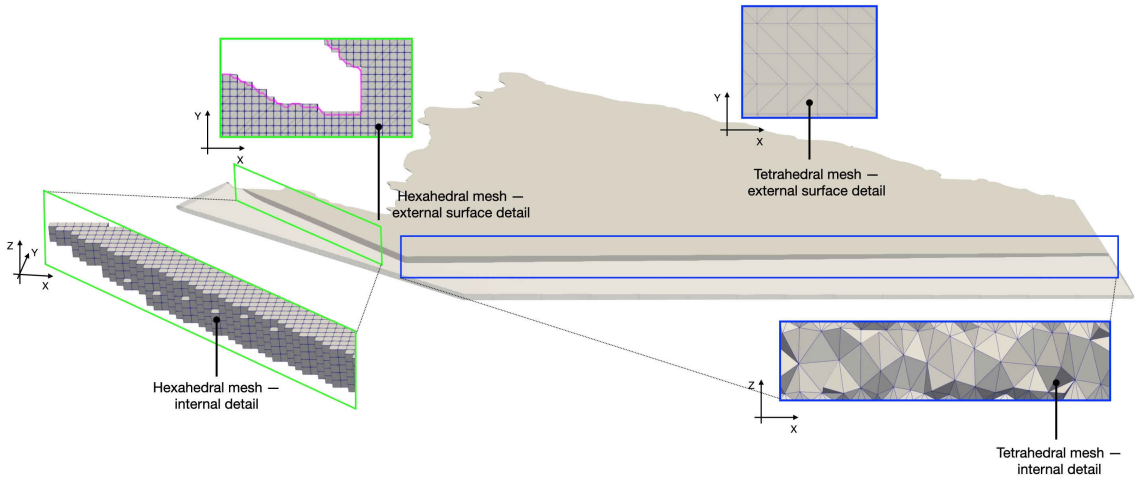

Figure 10: *An illustrative example of the results obtained modeling coastal surface water volume coupling bathymetry and harbor domain boundary. Two different meshing techniques are tested and compared, including tetrahedral (i.e., details marked by blue boxes) and hexahedral meshing with a* $(20 \times 20 \times 20)$ *grid resolution (i.e., details marked by green boxes).*

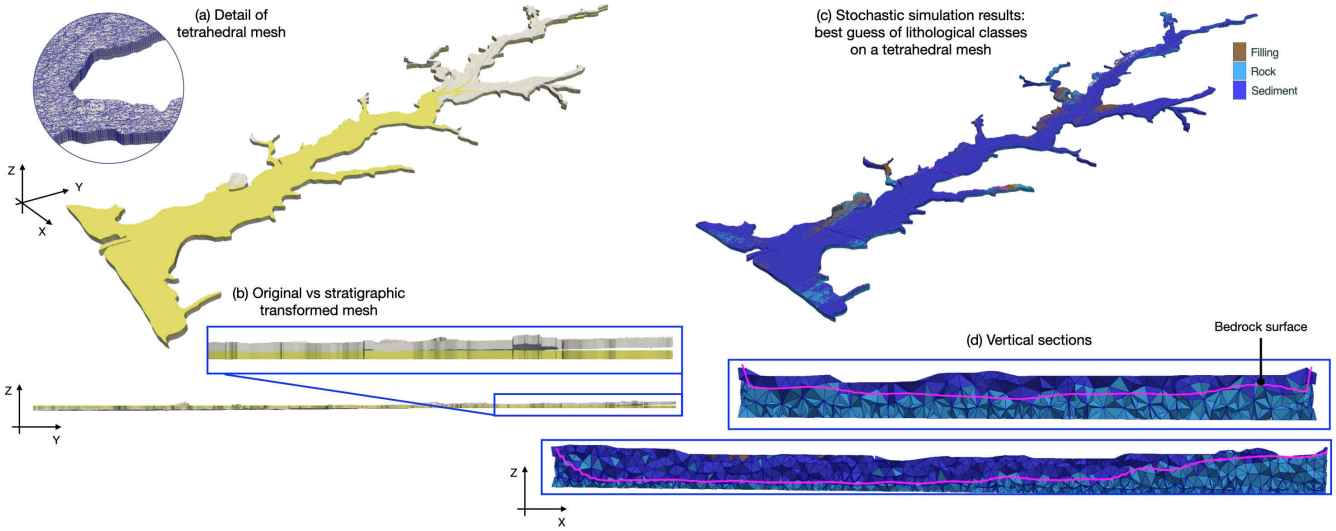

Figure 11: *Injecting geometry processing in a geostatistic modeling workflow: the case study of Polcevera basin (Genoa, Italy). The (a) volume tetrahedral mesh becomes the geometric support for (c) spatial prediction and (d) detecting the interface between sediment and rock (i.e., the line marked in magenta), including (b) stratigraphic coordinate transformation for coherent estimation of spatial covariance laws.*

5.2. Limitations

While MUSE-geometry successfully bridges a significant gap between computational geometry and geoscientific applications, the current implementation has several limitations that present opportunities for future development.

First, as a command-line tool, it requires a certain level of technical proficiency, which may present a barrier to adoption for geoscientists and GIS analysts accustomed to graphical user interfaces. The lack of real-time visual feedback during mesh generation and editing operations can make the process of parameter tuning and debugging iterative and less intuitive. Second, the tool's performance is constrained when processing exceptionally large datasets, such as continent-scale digital elevation models or high-resolution LiDAR point clouds. While the underlying libraries are efficient, the current architecture of MUSE-geometry does not yet support out-of-core processing or distributed computing, limiting the scale of models that can be handled on standard workstations. Third, while MUSE-geometry supports a wide range of standard geospatial and mesh formats, its internal data model is primarily geometry-focused. The handling of rich, heterogeneous attribute data linked to geometric elements is currently basic. Attributes are preserved through operations but lack dedicated, optimized storage or advanced querying capabilities within the tool itself. Fourth, currently MUSE-geometry is not able to manage geographical coordinates and thus cannot be used to work directly on the Earth's spheroid. Finally, the current meshing algorithms in MUSE-geometry, while robust, are primarily focused on Delaunay-based methods and simple extrusions. The tool does not yet include more advanced techniques such as anisotropic meshing tailored to directional physical phenomena, boundary layer generation for computational fluid dynamics, or hex-dominant meshing for improved numerical performance in certain simulations.

6. Conclusions

We have presented MUSE-geometry, a lightweight open-source tool designed to bridge the gap between advanced computational geometry and the specific needs of geoscientific modeling. By integrating geospatial awareness with robust mesh generation and editing capabilities, the tool provides the possibility to create simulation-ready 2D and 3D models from real-world environmental data. Our experiments demonstrate its versatility in handling diverse scenarios, from terrain modeling to volumetric mesh generation for surface and subsurface simulations.

MUSE-geometry is intended to serve as a solid foundation upon which new methodologies for geospatially-aware mesh generation and processing can be developed. While the current command-line implementation ensures interoperability and scripting, the next step is developing a graphical user interface (GUI) to make these tools more accessible to domain experts without a programming background. Furthermore, to address the increasing volume and heterogeneity of geoscientific data, we plan to investigate optimized data storage solutions, including the integration with spatial databases to enhance performance and support complex geometric and attribute queries.

We believe these future developments will greatly ease access to advanced geometric processing in the geosciences and promote stronger collaboration between the computer graphics and earth science communities. The MUSE-geometry source code is publicly available to encourage community involvement and further innovation in this interdisciplinary field.

Acknowledgements

Marianna Miola acknowledges the Italian National Biodiversity Future Center (NBFC) - National Recovery and Resilience Plan (NRRP) funded by the European Union - NextGenerationEU (project code CN 00000033).

Daniela Cabiddu, Simone Pittaluga, and Marino Vetuschi Zuccolini are members of the RAISE Innovation Ecosystem, funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE - Robotics and AI for Socio-economic Empowerment" (ECS00000035).

References

[Ada19] ADASZEWSKI S.: Official repository of concaveman-cpp. https://github.com/sadaszewski/concaveman-cpp, 2019. Accessed: 2025-01-16. 5

[AGM20] ALVIOLI M., GUZZETTI F., MARCHESINI I.: Parameter-free delineation of slope units and terrain subdivision of italy, 2020. Dataset and software available at https://geomorphology.irpi.cnr.it/tools/slope-units. URL: https://geomorphology.irpi.cnr.it/tools/slope-units. 9

[AGSG25] ARGUDO O., GUÉRIN E., SCHOTT H., GALIN E.: Terrain descriptors for landscape synthesis, analysis and simulation. *Computer Graphics Forum 44*, 2 (2025), e70080. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.70080, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.70080, doi:https://doi.org/10.1111/cgf.70080. 2

[All11] ALLEN D. W.: *Getting to Know ArcGIS ModelBuilder*. Esri Press, 2011. 3

[Att17] ATTENE M.: ImatiSTL-fast and reliable mesh processing with a hybrid kernel. In *Transactions on Computational Science XXIX*. Springer, 2017, pp. 86–96. 3

[BC94] BONHAM-CARTER G.: *Geographic information systems for geoscientists: modelling with GIS*. No. 13. Elsevier, 1994. 2

[BCBC08] BERTONCELLO A., CAERS J., BIVER P., CAUMON G.: Geostatistics on stratigraphic grids. In *Proc. eighth geostatistical geostatistics congress* (2008), vol. 2, pp. 677–686. 7

[BG25] BRGM, GEOPHYSICS I.: Geomodeller: 3d geological modelling and geophysical inversion software. https://www.intrepid-geophysics.com/geomodeller, 2025. Accessed: 2025-09-25. 3

[Boi10] BOISVERT J.: *Geostatistics with locally varying anisotropy*. PhD thesis, University of Alberta, 2010. URL: https://doi.org/10.7939/R31X5C, doi:10.7939/R31X5C. 6

[CB15] CACACE M., BLÖCHER G.: MeshIt—a software for three dimensional volumetric meshing of complex faulted reservoirs. *Environmental Earth Sciences 74*, 6 (2015), 5191–5209. 3

[CC24] COATLÉVEN J., CHAUVEAU B.: Large structure simulation for landscape evolution models. *Earth Surface Dynamics 12*, 5 (2024), 995–1026. URL: https://esurf.copernicus.org/articles/12/995/2024/, doi:10.5194/esurf-12-995-2024. 2

[CCLCdV*09] CAUMON G., COLLON P., LE CARLIER DE VESLUD C., VISEUR S., SAUSSE J.: Surface-Based 3D Modeling of Geological Structures. *Mathematical geosciences 41* (11 2009), 927–945. doi:10.1007/s11004-009-9244-2. 2, 4

[CGA24] CGAL DEVELOPMENT TEAM: CGAL, Computational Geometry Algorithms Library, 2024. https://www.cgal.org. 3

[CGEY25] CIAMPI P., GIANNINI L. M., ESPOSITO C., YOUNSI S.: Reconstructing urban landscape evolution with historical maps and geophysical techniques: 3D time-sensitive and multi-source geomodelling. *Environmental Earth Sciences 84*, 9 (2025), 1–16. 2

[CKH*15] COCKETT R., KANG S., HEAGY L. J., PIDLISECKY A., OLDENBURG D. W.: Simpeg: An open source framework for simulation and gradient based parameter estimation in geophysical applications. *Computers & Geosciences* (2015). 3

[Cor] COREFORM CUBIT: Coreform Cubit hex meshing software. Accessed: 2025-10-02. URL: https://coreform.com/coreform-cubit/. 3

[CVF*08] CÂMARA G., VINHAS L., FERREIRA K. R., QUEIROZ G. R. D., SOUZA R. C. M. D., MONTEIRO A. M. V., CARVALHO M. T. D., CASANOVA M. A., FREITAS U. M. D.: *TerraLib: An Open Source GIS Library for Large-Scale Environmental and Socio-Economic Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 247–270. URL: https://doi.org/10.1007/978-3-540-74831-1_12, doi:10.1007/978-3-540-74831-1_12. 3

[Dev24] DEVELOPERS B.: Boost.geometry, 2024. https://www.boost.org/doc/libs/release/libs/geometry/. 3

[dev25] DEVELOPERS S.: discretize: Finite volume discretization toolkit. https://github.com/simpeg/discretize, 2025. Accessed: 2025-09-25. 3

[EDF25] EDF R&D AND CEA: Salome platform, 2025. https://www.salome-platform.org/. 3

[Fra70] FRANKLIN W. R.: Pnpoly, 1970. URL: https://wrfranklin.org/Research/Short_Notes/pnpoly.html. 5

[GDA22] GDAL/OGR CONTRIBUTORS: *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation, 2022. URL: https://gdal.org, doi:10.5281/zenodo.5884351. 3, 5

[GGP*19] GALIN E., GUÉRIN E., PEYTAVIE A., CORDONNIER G., CANI M.-P., BENES B., GAIN J.: A Review of Digital Terrain Modeling. *Computer Graphics Forum 38*, 2 (2019), 553–577. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13657, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13657, doi:https://doi.org/10.1111/cgf.13657. 2

[GHS21] GÓMEZ-HERNÁNDEZ J. J., SRIVASTAVA R. M.: One step at a time: the origins of sequential simulation and beyond. *Mathematical Geosciences 53*, 2 (2021), 193–209. 6

[GMHH25] GABLE C. W., MILLER T. A., HYMAN J. D., HARP D. R.: LaGriT: Los alamos grid toolbox for mesh generation and optimization. https://github.com/lanl/LaGriT, 2025. Accessed: 2025-09-25. 3

[GR09] GEUZAINE C., REMACLE J.-F.: Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering 79*, 11 (2009), 1309–1331. URL: https://doi.org/10.1002/nme.2579, doi:10.1002/nme.2579. 3

[GRA22] GRASS DEVELOPMENT TEAM: *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 8.2*. Open Source Geospatial Foundation, 2022. URL: https://grass.osgeo.org, doi:https://doi.org/10.5281/zenodo.5176030. 3

[HMPP19] HARMON B. A., MITASOVA H., PETRASOVA A., PETRAS V.: r.sim.terrain 1.0: a landscape evolution model with dynamic hydrology. *Geoscientific Model Development 12*, 7 (2019), 2837–2854. URL: https://gmd.copernicus.org/articles/12/2837/2019/, doi:10.5194/gmd-12-2837-2019. 2

[HSG*19] HU Y., SCHNEIDER T., GAO X., ZHOU Q., JACOBSON A., ZORIN D., PANOZZO D.: TriWild: Robust Triangulation with Curve Constraints. *ACM Trans. Graph. 38*, 4 (July 2019), 52:1–52:15. URL: http://doi.acm.org/10.1145/3306346.3323011, doi:10.1145/3306346.3323011. 3

[HSW*20] HU Y., SCHNEIDER T., WANG B., ZORIN D., PANOZZO D.: Fast tetrahedral meshing in the wild. *ACM Trans. Graph. 39*, 4 (July 2020). URL: https://doi.org/10.1145/3386569.3392385, doi:10.1145/3386569.3392385. 3

[HZG*18] HU Y., ZHOU Q., GAO X., JACOBSON A., ZORIN D., PANOZZO D.: Tetrahedral meshing in the wild. *ACM Trans. Graph. 37*, 4 (July 2018), 60:1–60:14. URL: http://doi.acm.org/10.1145/3197517.3201353, doi:10.1145/3197517.3201353. 3

[J*23] JACOBSON A., ET AL.: libigl: A simple c++ geometry processing library, 2023. https://libigl.github.io. 3, 5

[LB22] LATIFI A. M., BOISVERT J.: Stratigraphic coordinate transformation. *GeostatisticsLessons. Retrieved from http://www.geostatisticslessons.com/lessons/stratcoords* (2022). 4, 6, 7

[lea] Leapfrog Geo. Accessed: 2025-09-16. URL: https://www.seequent.com/products-solutions/leapfrog-geo/. 3

[LHL*20] LICHTNER P. C., HAMMOND G. E., LU C., KARRA S., BISHT G., ANDRE B., MILLS R. T., KUMAR J., FREDERICK J. M.: *PFLOTRAN User Manual*. Tech. rep., 2020. http://documentation.pflotran.org. 9

[Liv19] LIVESU M.: cinolib: a generic programming header only C++ library for processing polygonal and polyhedral meshes. *Transactions on Computational Science XXXIV* (2019). https://github.com/mlivesu/cinolib/. doi:10.1007/978-3-662-59958-7_4. 3

[Lé25] LÉVY B.: geogram: a programming library with geometric algorithms. https://github.com/BrunoLevy/geogram, 2025. Accessed: 2025-09-25. 3

[Mal92] MALLET J.: GOCAD: A computer aided design program for geological applications. In *Three-dimensional modeling with geoscientific information systems*. Springer, 1992, pp. 123–141. 3, 7

[MBB*24] MONTI R., BISTACCHI A., BENEDETTI G., HUSSAIN W., ET AL.: 3D Geomodelling of Alpine structures with PZero. 3

[McL04] MCLENNAN J. A.: Using the variogram to establish the stratigraphic correlation structure. *Sixth Annual Report of the Centre for Computational Geostatistics, University of Alberta, Canada* (2004). 4, 6

[MCM*24] MIOLA M., CABIDDU D., MORTARA M., PITTALUGA S., SORGENTE T., ZUCCOLINI M. V.: Advancing Environmental Modeling with Unstructured Meshes: Current Research and Development. In *Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference* (2024), Caputo A., Garro V., Giachetti A., Castellani U., Dulecha T. G., (Eds.), The Eurographics Association. doi:10.2312/stag.20241350. 4

[MCP*22] MIOLA M., CABIDDU D., PITTALUGA S., MORTARA M., VETUSCHI ZUCCOLINI M., IMITAZIONE G.: A computational approach for 3D modeling and integration of heterogeneous geo-data. *Computers & Graphics 105* (2022), 105–118. URL: https://www.sciencedirect.com/science/article/pii/S0097849322000681, doi:https://doi.org/10.1016/j.cag.2022.05.002. 3, 5

[MCPVZ22] MIOLA M., CABIDDU D., PITTALUGA S., VETUSCHI ZUCCOLINI M.: MUSE: Modeling Uncertainty as a Support for Environment. In *Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference* (2022), Cabiddu D., Schneider T., Allegra D., Catalano C. E., Cherchi G., Scateni R., (Eds.), The Eurographics Association. doi:10.2312/stag.20221265. 2

[MDA20] MDAL CONTRIBUTORS: *The MDAL Mesh Data Abstraction software Library*. Lutra Consulting Ltd., 2020. 3

[Mio25] MIOLA M.: *Increase the knowledge of Natural Systems through the evaluation of the uncertainty of environmental data: operational theory and application*. PhD thesis, Università degli Studi di Genova, 2025. 2

[NLP*13] NATALI M., LIDAL E. M., PARULEK J., VIOLA I., PATEL D.: Modeling terrains and subsurface geology. In *Eurographics (State of the Art Reports)* (2013), pp. 155–173. 2

[ope] OPENTOSCANA. https://dati.toscana.it/dataset/dem10mt. Accessed: 2025-10-03. 8

[Ope25] OPENFOAM FOUNDATION: Openfoam. https://www.openfoam.com/, 2025. C++ library and CFD toolbox for continuum mechanics simulations. 3

[OSG25] OSGEO: shapelib, 2025. https://github.com/OSGeo/shapelib.git. 3

[PCJ*17] PELLERIN J., CAUMON G., JULIO C., MEJIA-HERRERA P., BOTELLA A.: Ringmesh: A programming library for developing mesh-based geomodeling applications. *Computers & Geosciences 101* (2017), 175–186. URL: https://www.sciencedirect.com/science/article/pii/S0098300417302637, doi:10.1016/j.cageo.2017.03.005. 3

[PD14] PYRCZ M., DEUTSCH C.: *Geostatistical Reservoir Modeling, 2nd Edition*. 05 2014. 7

[Poi24] POINT CLOUD LIBRARY PROJECT: Pcl: Point cloud library, 2024. https://pointclouds.org. 3

[PRO22] PROJ CONTRIBUTORS: *PROJ coordinate transformation software library*. Open Source Geospatial Foundation, 2022. URL: https://proj.org/, doi:10.5281/zenodo.5884394. 3, 5

[QGI24] QGIS DEVELOPMENT TEAM: *QGIS Geographic Information System*. QGIS Association, 2024. URL: https://www.qgis.org. 3

[RKCC15] RUBIO R. H., KOPPE V. C., COSTA J. F. C. L., CHERCHENEVSKI P. K.: How the use of stratigraphic coordinates improves grade estimation. *Rem: Revista Escola de Minas 68*, 4 (2015), 471–477. 4, 6

[Sch25] SCHLUMBERGER: Petrel subsurface software: Geology and modeling. https://www.slb.com/products-and-services/delivering-digital-at-scale/software/petrel/petrel-subsurface-software/petrel/petrel-geology-and-modeling, 2025. Accessed: 2025-09-25. 4

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, Lin M. C., Manocha D., (Eds.), vol. 1148 of *Lecture Notes in Computer Science*. Springer-Verlag, May 1996, pp. 203–222. From the First ACM Workshop on Applied Computational Geometry. 3, 5

[Si24] SI H.: Tetgen: A quality tetrahedral mesh generator, 2024. https://wias-berlin.de/software/tetgen/. 3, 5

[SMP*25] SORGENTE T., MIOLA M., PITTALUGA S., CABIDDU D., MORTARA M., VETUSCHI ZUCCOLINI M.: Segmentation of stochastic scalar fields in unstructured meshes. *Computers & Geosciences* (2025), 106041. URL: https://www.sciencedirect.com/science/article/pii/S0098300425001918, doi:https://doi.org/10.1016/j.cageo.2025.106041. 2

[Tea25] TEAM P. D.: DMPlex: Unstructured mesh / discretization object in petsc. https://petsc.org, 2025. Accessed: 2025-09-25. 3

[TG07] TURNER A. K., GABLE C. W.: A review of geological modeling. *Three-dimensional geologic mapping for groundwater applications* (2007), 75–79. 2

[TPMP12] TALISCHI C., PEREIRA A., MENEZES I. F. M., PAULINO G. H.: Polymesher: A general-purpose mesh generator for polygonal elements. *Structural and Multidisciplinary Optimization 45* (2012), 309–328. doi:10.1007/s00158-011-0706-z. 3

[VCG24] VCG TEAM: Vcglib: Visualization and computer graphics library, 2024. https://vcg.isti.cnr.it/vcglib/. 3

[XL14] XING H., LIU Y.: Mesh Generation for 3D Geological Reservoirs with Arbitrary Stratigraphic Surface Constraints. *Procedia Computer Science 29* (12 2014), 897–909. doi:10.1016/j.procs.2014.05.081. 3

[ZBGS15] ZEHNER B., BÖRNER J. H., GÖRZ I., SPITZER K.: Workflows for generating tetrahedral meshes for finite element simulations on complex geological structures. *Computers & Geosciences 79* (2015), 105–117. URL: https://www.sciencedirect.com/science/article/pii/S009830041500031X, doi:https://doi.org/10.1016/j.cageo.2015.02.009. 2