

The crucial role of meshing in computational fluid dynamics simulations for organic geometries in paleobiology: Describing fluid dynamics performance through best practices

Matheo López-Pachón  | Jordi Marcé-Nogué 

Department d'Enginyeria Mecànica,
 Universitat Rovira i Virgili, Tarragona,
 Spain

Correspondence
 Jordi Marcé-Nogué
 Email: jordi.marce@urv.cat

Funding information
 European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No., Grant/Award Number: 945413; Universitat Rovira i Virgili, Grant/Award Number: 2023PFR-URV-1239; Agency for Management of University and Research, Grant/Award Number: SGR2021-01239

Handling Editor: Russell Dinnage

Abstract

1. Computational fluid dynamics (CFD) has become an essential tool for studying fluid interactions in biological systems. While widely used in engineering, its application in the natural sciences, particularly in paleobiology, remains limited due to the challenges of meshing complex organic geometries. This study aimed to bridge that gap by serving both as an accessible tutorial and as an empirical comparison of meshing strategies and their effects on simulation accuracy and efficiency.
2. We analyse hybrid meshing techniques that integrate structured and unstructured grids to improve boundary layer resolution while optimizing computational costs. As a case study, we focus on the Japanese giant salamander (*Andrias japonicus*), assessing the impact of meshing choices on the accuracy of CFD simulations. Our approach involves comparing mesh configurations and evaluating their performance in reconstructing fluid dynamics around biological structures.
3. Our findings demonstrate that well-optimized meshing strategies significantly enhance the reliability of CFD simulations in reconstructing biomechanics and ecological interactions of both extinct and extant species. The integration of structured and unstructured grids improves flow resolution, providing insights into locomotion, functional adaptations and environmental interactions. These results highlight the importance of meshing choices in advancing CFD applications in evolutionary biology.
4. This study underscores the potential of CFD as a powerful tool in paleobiology and functional morphology. By offering both practical guidance and structured comparison of meshing techniques, we provide a framework that enhances the accessibility and accuracy of CFD simulations in evolutionary studies. Our work facilitates the integration of CFD into paleontological research, improving reconstructions of extinct species' biomechanics and ecological roles.

KEY WORDS

computational fluid dynamics, CT scan, drag coefficient, hybrid meshes, lift coefficient, photogrammetry, turbulence models

1 | INTRODUCTION

Organismal biologists often propose hypotheses about the relationship between form and function, but it remains uncommon to support these ideas using numerical methods. Computational fluid dynamics (CFD) provides a robust framework to simulate and analyse fluid-structure interactions, offering unique insights into biomechanical and ecological processes. The manuscript introduces key CFD concepts with a particular focus on mesh generation, aiming to make these tools more accessible to researchers in organismal biology. A simplified technical overview is presented to help those working in functional morphology and biomechanics critically interpret CFD-based studies and apply best practices in mesh design. The manuscript also explores how turbulence models relate to meshing decisions, particularly regarding boundary layer resolution, without delving into complex modelling theories.

In recent years, mechanical engineering has become highly interdisciplinary, encompassing fields beyond its traditional boundaries. This has been accomplished through the use of computational tools employing approximate numerical methods, such as Multibody Dynamics Analysis (MDA) (Huston, 1991), Finite Element Analysis (FEA) (Unnikrishnan & Somanath, 2024) and Computational Fluid Dynamics (CFD) (Versteeg & Malalasekera, 2007), to predict various scenarios in which physics plays a fundamental role. MDA focusses on the kinematic analysis of rigid bodies, namely the motion of objects subject to inertial conditions and constraints (Haug, 1989; Shabana, 2020). FEA enables the analysis of deformable solids under various loading conditions and thermal problems (Erhunmwun & Ikponmwosa, 2017; Rao, 2005).

The physics of fluids is inherently complex, governed by the Navier-Stokes (N-S) equations, which currently lack a general analytical solution for most practical cases (Khatib-Shahidi & Smith, 2010). This fundamental challenge has classified the N-S equations as one of the millennium problems (Pérezagua López, 2023) making numerical methods the only option to achieve an approximate solution, with the Finite Volume Method (FVM) being the most widely used among the many available options (Gerck, 2022). These methods allow for the discretization of the fluid domain, breaking it into smaller control cells where conservation laws are applied. As a result, CFD enables the simulation and analysis of fluid behaviour across various applications, from aerodynamics to biomechanics, despite the inherent limitations in accurately capturing all turbulence scales (Versteeg & Malalasekera, 2007; Young & Hildebrand, 1957).

Those three numerical methods can be as complex as desired. However, CFD stands out for its unique combination of complexity

and versatility, as it involves solving nonlinear partial differential equations (PDE) specifically the Navier-Stokes equations, which are further complicated by fluid turbulence. This versatility allows for the analysis of aerodynamics, hydrodynamics, combustion, heat transfer, mass transport, rheology, electromagnetism, acoustics and many other fields (Bahary, 1994; Cedrón, 2018). Additionally, CFD allows for two-phase simulations involving air and water, and three-phase simulations with the addition of solid particles. Moreover, CFD enables the incorporation of solid mechanics from FEA, coupling both methods and developing Fluid-Structure Interaction (FSI) analysis (Shahril et al., 2024). The addition of dynamic meshes and the time variable leads to the integration of CFD with MDA (Li et al., 2015; Liu et al., 2017). However, the CFD numerical method is highly sensitive to various phenomena, which can occur simultaneously. This sensitivity categorizes it as a multi-physics method, making its use, understanding, and computational cost challenging.

Both FEA and CFD solve the PDE corresponding to their physics by dividing the volume, whether fluid or solid, into small cells to simplify the calculation. This process is known as discretization, which results in a mesh (Marcé Nogué et al., 2015). Afterwards, each cell is assembled to form a system that can be computationally solved approximately. Meshes are essential for obtaining accurate predictions in numerical methods. While a well-designed mesh does not always ensure the best solution, an incorrect mesh will inevitably produce poor results (Cadence, 2016).

In recent years, CFD tools are expanding beyond conventional engineering applications to fields such as biology, describing the behaviour of living organisms and even aiding in the interpretation of extinct animal behaviour (Rahman, 2017).

In biology, there are records of two-dimensional CFD analysis in tadpoles from the 1990s (Liu et al., 1996). Gradually, with better computational technologies, the use of 3D models became more common and helped determine swimming hydrodynamics in trilobites (Esteve et al., 2021; Esteve & López-Pachón, 2023; Wang et al., 2024) and other invertebrates (Hebdon et al., 2020; Lei et al., 2023; Rahman et al., 2020). In vertebrates, more and more experiments are being carried out to validate the results of CFD simulations, such as the study of suction-feeding in the living *Andrias davidianus* (Heiss et al., 2013), as well as in other extinct and living vertebrates (Bourke et al., 2014; Brown & Kirk, 2023; Grohganz et al., 2023; He et al., 2024; Skews, 2016; Troelsen et al., 2019). In addition, CFD can offer evolutionary insights: for example, CFD was used to assess the energy expenditure of *Ichthyosaurus* in water, comparing it with present-day aquatic species (Gutarría Diaz et al., 2019). All these applications underline the value of computational methods for testing ecological hypotheses from fossil

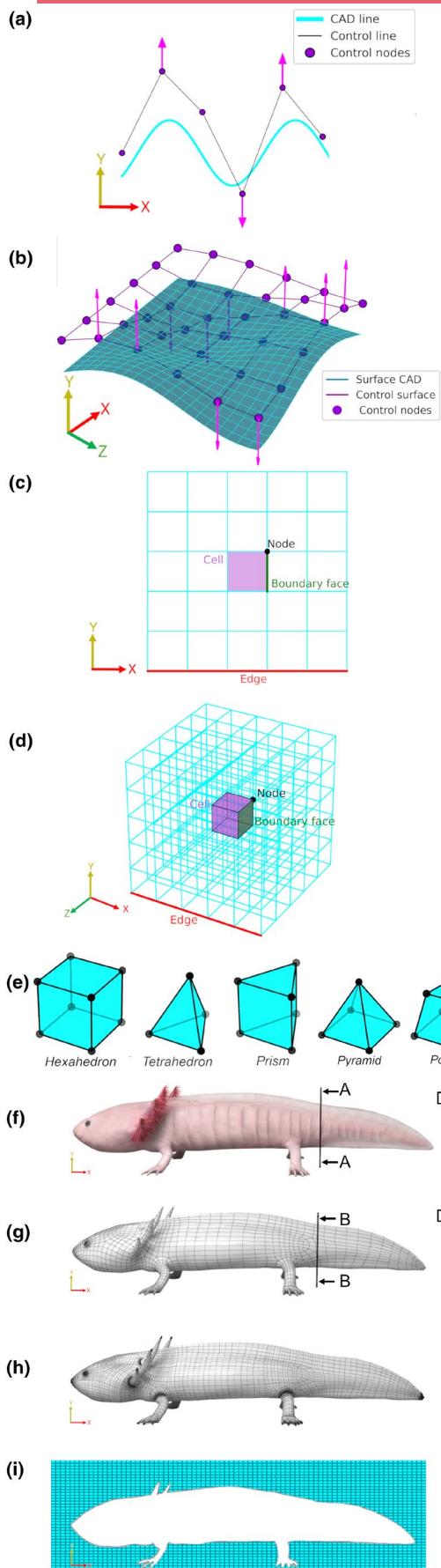


FIGURE 1 (a) Control nodes in NURBS curves in the 2D model, (b) control nodes in NURBS surfaces in the 3D model, (c) parts included in the CFD mesh for a simple 2D model, (d) parts included in the CFD mesh for the 3D model, adapted from (ManchesterCFD, 2024), and (e) type of cells. Different types of meshes (f) real specimen, (g) CAD geometric, (h) FEA, and (i) CFD.

specimens (Rahman, 2017). This advance in palaeontology is driving a transition from a narrative-based science to a more precise and quantitative one, thanks to new digital technologies.

For the reasons mentioned above, here we focus on meshes as the discretization of the computational domain of fluids in the CFD method in palaeobiological research and to discuss how the proper use of meshes can significantly impact the accuracy of CFD simulations (Blocken et al., 2011; Tominaga et al., 2008).

This work aimed to help biologists understand and use CFD more critically and with greater confidence, especially when dealing with simulations in studies of form and function. Through examples, practical tips and clarifications of common mistakes, the text serves as a useful introductory guide, useful both for those applying these methods and for those interpreting them in collaborative or review contexts. In doing so, we hope to support dialogue between biology and numerical simulation, and to encourage broader and more rigorous use of these tools in the life sciences.

2 | WHAT IS A MESH?

A mesh is a computational discretization that divides a continuous domain into finite elements, enabling numerical analysis in various scientific and engineering fields. In computational fluid dynamics (CFD), finite element analysis (FEA) and structural mechanics, a mesh represents the spatial domain by breaking it into small, manageable regions where governing equations can be approximated numerically. The primary purpose of meshing is to convert complex geometries into a network of nodes and elements, allowing for the efficient application of numerical methods such as the Finite Volume Method (FVM) or Finite Element Method (FEM). The quality of a mesh significantly influences the accuracy, stability and computational cost of simulations. A well-structured mesh ensures proper resolution of critical regions, such as boundary layers in CFD or stress concentration zones in FEA, while minimizing unnecessary computational costs. Therefore, selecting the appropriate mesh type and resolution is fundamental to obtaining reliable and precise simulation results.

In the context of this document, a mesh is a structure comprised of lines or points that delineate a series of polygons, polyhedrons or other geometric cells. These components serve to segment and organize a two- or three-dimensional space into smaller, more manageable regions (Ganovelli et al., 2014). The principal cell types for mesh generation in computational simulations vary by dimensionality and physical purpose (Figure 1e). In 2D, triangles

and quadrilaterals (quads) are essential for surface discretization, adapting to planar and curved boundaries (Baker, 2005). In 3D, tetrahedra, hexahedra, prisms, pyramids and polyhedral elements serve different computational needs. Tetrahedra are flexible but computationally expensive due to high cell counts, while hexahedra offer efficiency in structured domains but are harder to fit into complex geometries. Prisms improve boundary layer modeling, pyramids aid in mesh transitions and polyhedra optimize 3D discretization by reducing element count and computational cost (Marcon et al., 2019).

2.1 | Mesh types

There are many ways to classify meshes. In the present work, the classification is based on the numerical method, except for the MDA, which is a method that does not involve meshing.

In engineering, square geometries are typically used, involving exact measurements, straight lines, right angles and symmetrical topologies. However, nature is made of irregular and curved geometries since the form of living beings is purely organic. For this reason, animal morphology represents a major challenge in digital modelling. With that in mind, geometries can be managed and analysed using a mesh.

Computer-Aided Design (CAD) models—created using specialized software packages allow—for high-precision and non-destructive testing, preserving fossil integrity while enabling fluid behaviour analysis (Rahman et al., 2017). Geometric meshes in CAD often utilize Non-Uniform Rational B-Splines (NURBS) curves (Figure 1a,b) to create smooth and organic shapes and can be deformed using control nodes, which is how geometries are shaped when a real scan is not available (Rogers, 2001). CAD models can be classified as shell-type, which only define the surface topology (Figure 1g), or solid

models, which represent the complete volumetric structure of an object (Figure 1f). These CAD models serve as the foundation for numerical simulations in various sciences, from engineering to palaeontology, aiding in the generation of high-quality meshes that ensure accurate computational results (Piegl & Tiller, 1997).

FEA and CFD meshes differ significantly in structure and purpose. FEA meshes encompass the entire model, including both its surface and its interior, to analyse mechanical behaviour. They require refined areas around stress concentrations, such as notches and welds (Figure 1h), to improve simulation accuracy (Fortuny et al., 2015; Marcé Nogué et al., 2015). In contrast, CFD meshes focus on the fluid domain, surrounding the geometry of interest (Figure 1i) and typically requiring a more extensive and intricate mesh. The complexity of CFD meshes arises from handling irregular geometries and the vast number of cells needed to capture flow dynamics accurately. Figure 1c,d illustrate 2D and 3D representations of simplified computational domains and the important components within a mesh (Baker, 2005).

In CAD design, various file formats are used to represent digital geometries. These formats differ in the type of information they store: some preserve parametric precision and model structure, while others contain only surface meshes. The choice of format depends on both the source of the model and the requirements of the analysis. Table 1 summarizes the most common options.

Digital models obtained from 3D scanning or computed tomography (CT) initially produce point clouds or image volumes. It is good practice to process point clouds with software such as Autodesk Recap, Fusion or Geomagic; MeshLab is a free alternative. For CT data, Avizo or Drangonfly are widely used, with open-source alternatives such as 3D Slicer, ITK-SNAP or Biomedisa. Once the surface mesh is generated, editors such as ZBrush or Blender (open-source) are suitable for organic modifications, while precise, symmetric editing or engineering-grade CAD models are

TABLE 1 Summary of common geometry file formats and their typical use in numerical methods.

Format	Data type	Best practices and typical applications in numerical methods
.IGES (.igs)	Precise curves and surfaces (NURBS)	Perhaps the most versatile format; widely supported across software. Widely used to transfer CAD to CFD/FEA mesh generation
.STEP (.stp, .step)	Parametric geometry and full topology (NURBS, solids, assemblies)	Widely used, more modern than IGES. Common in CFD, and heavily used in FEA for solid representation
.STL	Surface mesh (triangles only)	Very common for CFD with complex organic shapes, fossils, or biological specimens (from 3D scanning or photogrammetry)
.OBJ	Surface mesh (polygons)+optional textures/materials	Occasionally used in CFD when models come from photogrammetry or 3D modelling software
.BREP	Full precision Boundary Representation (CAD kernel)	Used in advanced pre-processors (OpenFOAM, Ansys, Comsol) for exact meshing of models

best prepared with tools such as Inventor, Solid Edge or the free option Onshape.

3 | APPROACH TO SALAMANDER *Andrias japonicus* MESHING

In this study, two-dimensional and three-dimensional meshes were generated to perform CFD simulations of the hydrodynamic performance of an *Andrias japonicus* salamander (Figure 10). The salamander model was based on a real specimen identified as STL820, from the Kanagawa Prefectural Museum of Natural History (Japan), with a snout-tail length (STL) of 820 mm and snout-vent length (SVL) of 564 mm. The individual died of natural causes and was preserved in ethanol. It was scanned using a Revopoint MIRACO Pro 3D structured light scanner, achieving a precision of up to 0.02 mm.

The CAD model was post-processed using the Oneshape and Meshlab software solutions. Using the meshing code snappyHex-Mesh from OpenFOAM, an unstructured hybrid mesh was developed within a polyhedral domain. The meshing process consisted of several stages: (1) the initial generation of a background hexahedral mesh, (2) refinement boxes in regions of interest and (3) the addition of prismatic layers to resolve boundary layer effects. Subsequently, steady-state simulations were carried out using Reynolds-Averaged Navier-Stokes (RANS) equations with the $k-\omega$ SST turbulence model, imposing an inlet velocity of 1 m/s and assuming incompressible flow. The results (post-processed with the Paraview software) provided valuable insights into the effectiveness of the meshing strategy and its impact on simulation accuracy, which is the main focus of this study.

The simulations were run on a system equipped with an AMD EPYC 7313 CPU featuring 16 cores (32 threads with SMT) at 3.0 GHz, 512 KiB of L1d and L1i cache per core, 8 MiB of L2 cache and 128 MiB of L3 cache. The system includes 4 NUMA nodes and 32 GB of RAM distributed across 4 memory nodes. MPI parallelization was employed, and all 32 available threads were used. All calculations were performed in double precision (64 bits) on a 64-bit Linux operating system (Ubuntu 24.04 LTS). The hardware is part of the cluster hosted at the Department of Mechanical Engineering at the Universitat Rovira i Virgili, specifically on the 'Arquimedes' node. The total computation time for the complete simulation, using a mesh of 9 million cells, was 10 h of wall-clock time.

4 | MESH GENERATION FOR CFD

4.1 | Discretization

Fluids are modelled as continuous because their molecules are densely packed and in constant motion. At macroscopic scales, any small region of fluid contains enough particles for properties

such as density, temperature, and velocity to be well-defined and to vary smoothly in space and time. However, to apply numerical methods, we must transition from this continuum perspective to a discrete representation. In other words, it is necessary to transform the PDEs into discrete algebraic equations because only in this form can numerical methods be implemented (Baker, 2005; Thompson et al., 1999). This transformation is carried out using mapping techniques (Figure 2), which enable the discretization of the computational domain and allow differential methods. These mesh types are visually summarized in Figure 2 and described in Table 2 for clarity and comparison.

These mesh types are visually summarized in Figure 2, described in Table 2 for clarity and further synthesized in Figure 3.

4.2 | Mesh classification by method

By examining the workflow shown in Figure 4, we can select the mesh type according to the method used. In the following sections, each method will be explained to clarify the criteria for making this selection.

4.2.1 | Structured method

Structured mesh methods include the Single Block and Contiguous Multiblock approaches, each suited to different types of geometries. The Single Block method uses a single block of ordered cells, making it ideal for simple and uniform geometries, offering high accuracy with low computational cost.

However, for more complex geometries, the Contiguous Multiblock method is a better choice. This approach divides the domain into multiple connected blocks, allowing each block to better fit different areas of the geometry while keeping cell alignment and smooth transitions between blocks. This continuity at the block interfaces ensures that the mesh remains structured and optimized for simulations that require greater flexibility without sacrificing precision (Baker, 2005).

Octree Meshes and Tetrahedral Meshes are two common types of unstructured meshes, each offering specific advantages for complex geometries. Octree Meshes use a hierarchical division of space, where each cell is subdivided into smaller cells, adapting naturally to irregular or intricate shapes by refining areas of interest while keeping other areas coarser (Figure 5a). This approach is particularly efficient for managing large, complex domains as it reduces the number of cells where high resolution is unnecessary. In contrast, Tetrahedral Meshes consist of tetrahedron-shaped cells and are widely used for their flexibility in filling irregular spaces, especially in three-dimensional domains (Samet, 1984). Tetrahedral meshes are commonly employed in applications where precise boundary representation is critical, though they may require additional refinement near complex boundaries to improve accuracy. Together, these unstructured mesh types enhance the

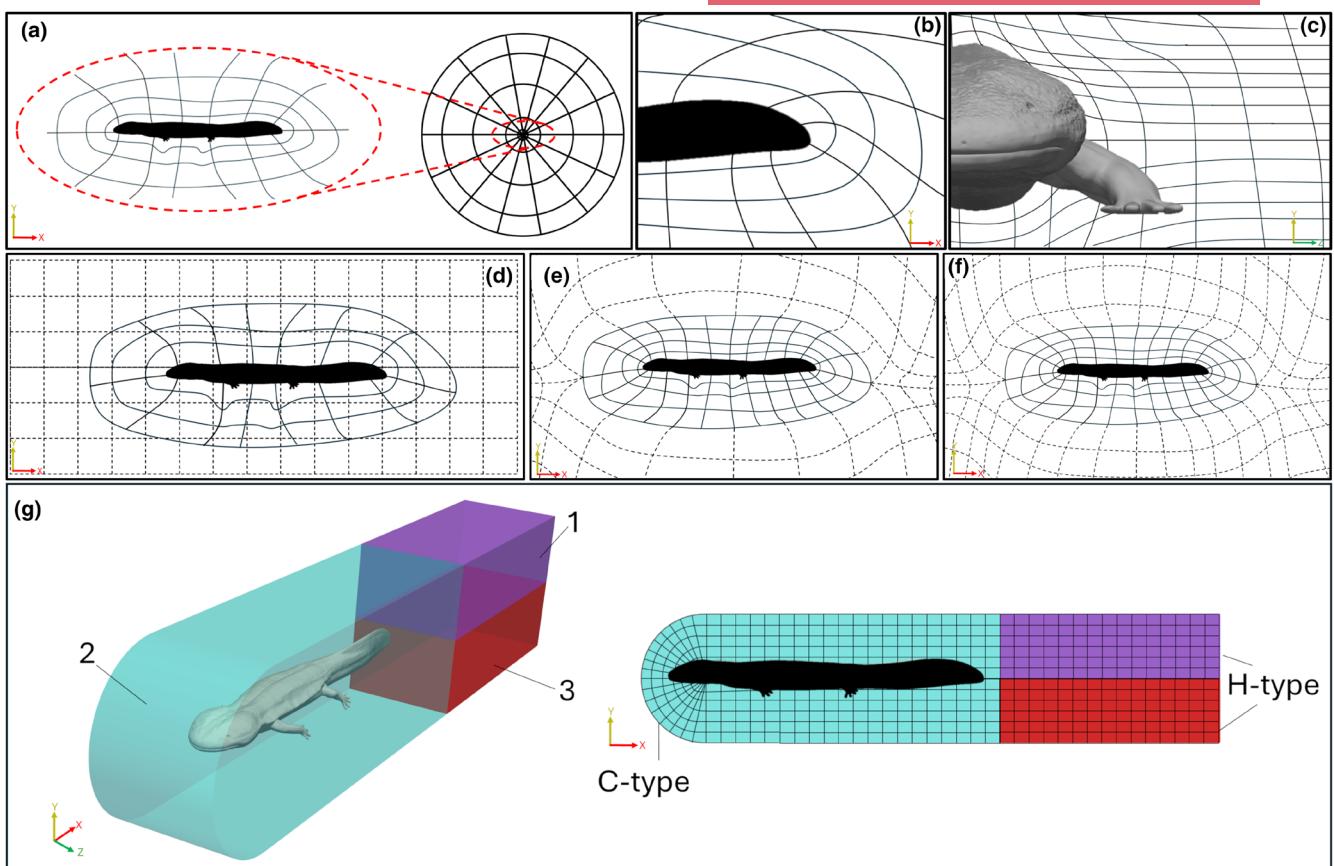


FIGURE 2 Different mapping meshes in the salamander *Andrias japonicus*. (a) O-mesh: Mapping into the unit circle (Theodorsen & Garrick, 1979), (b) C-mesh: Coordinate mapping for the tail of the salamander (Arlinger, 1975), (c) H-mesh: Section through the frontal symmetry of the salamander, (d) Overset method, (e) Patched method, (f) Composite method, and (g) a multiblock C-H mesh (Baker, 1986, 2005; Eiseman, 1979).

TABLE 2 Definitions of common meshing terms used in CFD.

Mesh type	Definition	Typical use/advantage
C-mesh	Structured mesh wrapping around the geometry in a C-like shape	Ideal for capturing wakes in 2D profiles (e.g. airfoils), good boundary layer control
H-mesh	Structured mesh aligned parallel to flow, with symmetry before and after the object	Useful for symmetric flows and rectangular domains
Patched mesh	Multi-block mesh where each block is generated independently and connected without node-to-node continuity	Allows flexible mesh generation in complex geometries with minimal constraint
Composite mesh	Multi-block mesh with full continuity between adjacent blocks (matching nodes across interfaces)	Ensures better flow solution continuity across domains, especially in structured grids
Multiblock method	Division of domain into smaller blocks to enable local control and structured meshing in each subdomain	Combines the advantages of structured meshing with adaptability to complex geometries

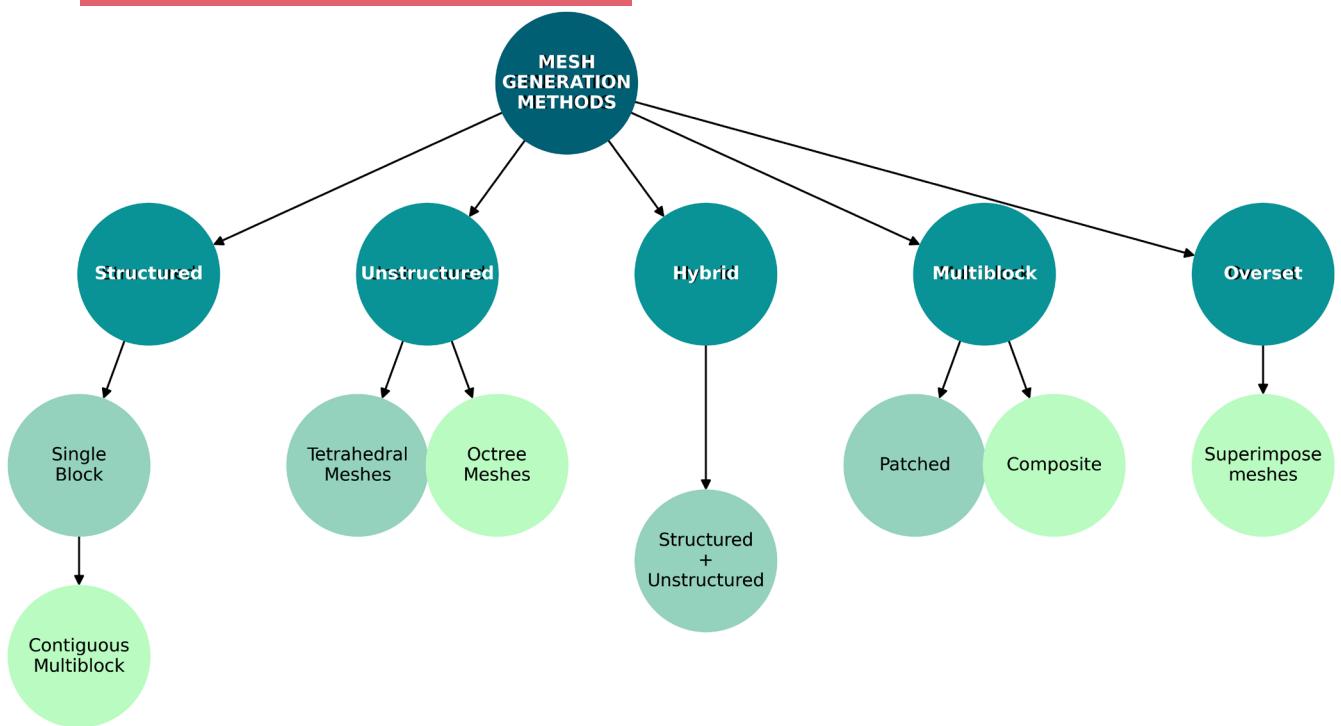


FIGURE 3 Summary of various mesh generation methods (Baker, 2005).

adaptability of CFD simulations across a range of geometries and flow conditions (Baker, 2005).

4.2.2 | Multiblock method

The multiblock meshing method is a strategy used in mesh generation to divide a complex domain into simpler blocks, which facilitates the creation of meshes in each block individually. This technique is based on the idea of decomposing the domain into smaller blocks, where each block can be mapped and meshed more simply than the entire domain.

Blockmesh is widely used in mesh combinations. In the context of a C-H mesh, 'C' generally refers to a structured mesh, where the grid lines are aligned with the geometry, providing good resolution and accuracy in regions of interest. On the other hand, 'H' refers to a block-structured mesh approach, where the domain is divided into multiple structured blocks or grids, allowing local refinement and adaptation. In (Figure 2g), a hybrid meshing approach would be employed, utilizing an H-H mesh combination within Blocks 1 and 3, while adopting a C-H mesh combination within Block 2. Each block represents a subdomain that is considerably less complex geometrically compared to the entire configuration. Consequently, these blocks can be efficiently meshed either by solving PDE or by utilizing algebraic methods (Lee et al., 1980).

Each block in the multiblock method can be meshed independently using algebraic or numerical methods. For example, PDE or algebraic methods such as transfinite interpolation can be applied to generate the mesh in each block. Subsequently, the meshes of the

individual blocks are combined to form a global mesh that covers the entire domain accurately.

This multiblock strategy allows greater flexibility and control in mesh generation by dividing the problem into more manageable parts. It also makes it easier to tailor the mesh resolution in specific areas of the domain, which can result in higher accuracy in numerical simulations. It even allows refinements to be made in the areas of interest that require better resolution, for example in regions near the geometry surface or within the boundary layer zone, where higher mesh resolution is required.

The two multiblock mesh methods are as follows (Baker, 2005):

Patched Methods: In this method, a block structure with defined boundaries is created, but it is not necessary to maintain the continuity of mesh lines between neighbouring blocks (Figure 2e). This allows for refined mesh in specific regions without the need to refine unnecessarily in other areas.

Composite Methods: This approach is a variant of the patched method where mesh lines are required to be continuous across block interfaces (Figure 2f). Although more difficult to create than a standard patched method, it offers significant advantages in terms of mesh line continuity and smoothness of the mesh.

4.2.3 | Triangulation methods

This is a mesh generation technique that divides a domain into triangular cells in 2D or tetrahedral cells in 3D (Figure 2b). These methods offer greater flexibility in discretizing complex geometries, as they are not constrained by a predefined mesh structure such as

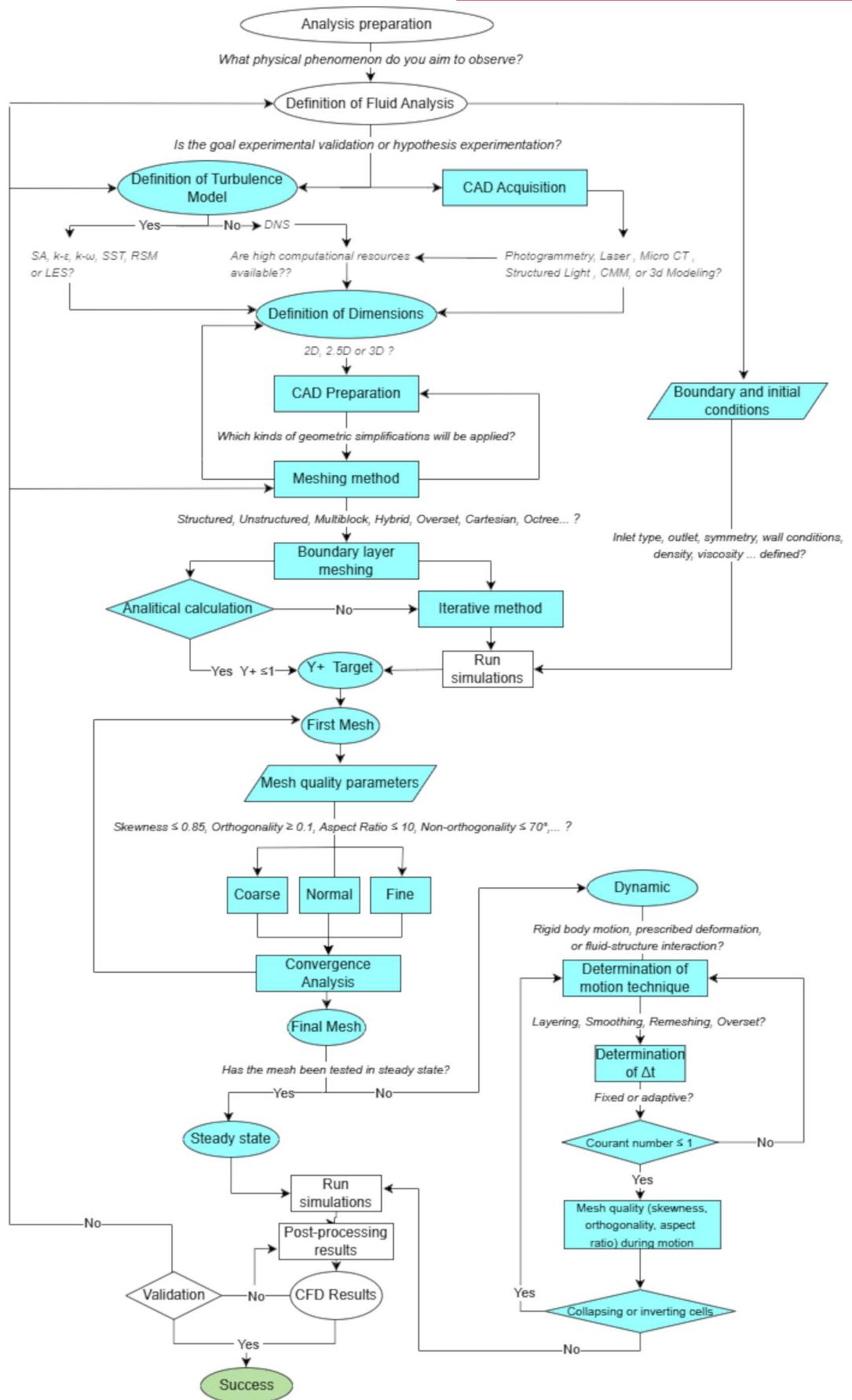


FIGURE 4 Conceptual workflow summarizing the meshing pipeline, including steady and dynamic mesh processes, quality criteria, and decision points. This diagram is intended to serve as a practical guide for researchers unfamiliar with CFD-based meshing strategies applied to biological models. The cyan colour represent work related with meshing.

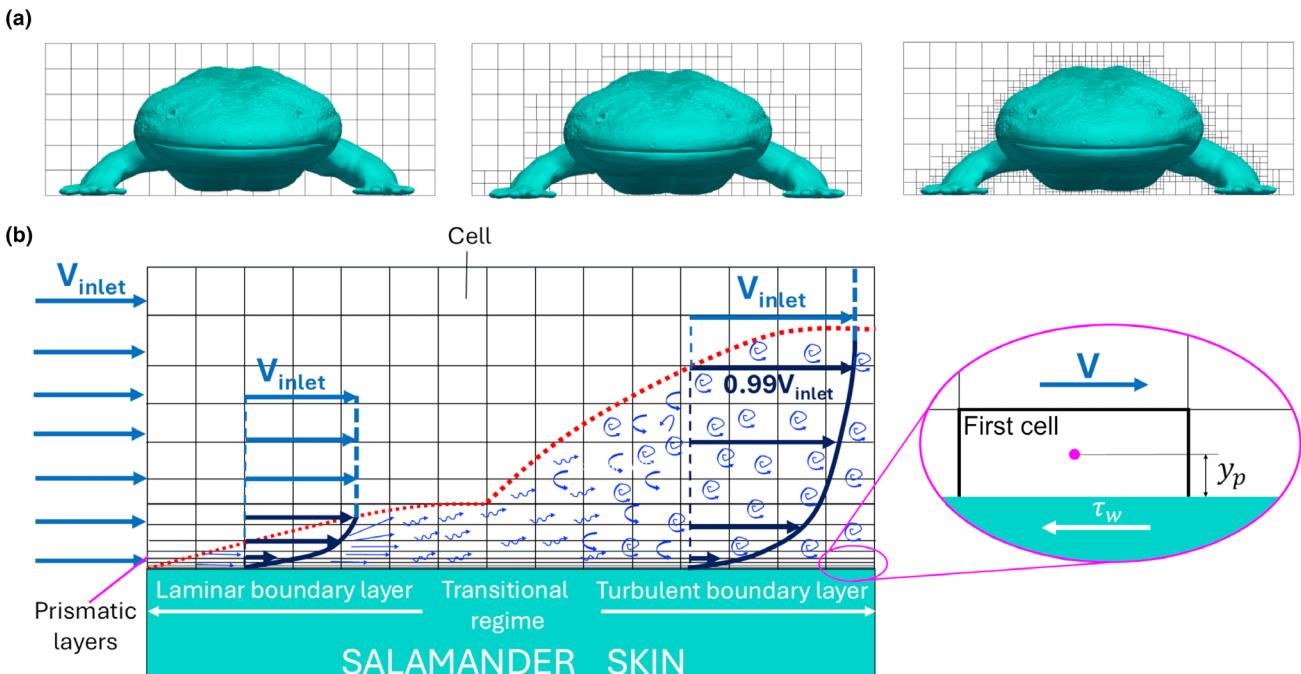


FIGURE 5 (a) Basic steps for block meshing used in software codes, illustrated here with an Octree decomposition of the region around the salamander. (b) Behaviour of the boundary layer over the skin of the salamander, highlighting the layered prismatic meshing, derived from (White, 2009).

block methods. Triangulation can be done incrementally by adding points and locally adjusting the mesh to maintain cell quality.

One common approach in triangulation methods is the use of algorithms based on Delaunay triangulation, which automatically generate meshes from an initial set of points. These methods offer the advantage of being able to insert points at specific locations to improve mesh quality while maintaining surface integrity. Although triangulation may require reconstruction operations to ensure mesh consistency, its level of automation makes it easier to generate meshes for a wide range of applications in engineering and computational sciences (Shewchuk, 1996).

The Advancing Front Method generates a mesh by starting from a boundary and advancing inward, allowing precise control over mesh refinement and quality in specific regions. This approach is particularly useful for ensuring high-resolution cells around critical areas while keeping the rest of the domain less refined, optimizing computational resources and accuracy (Baker, 2005).

5 | HOW TO BUILD A MESH IN CFD?

The meshing process in CFD simulations, particularly in biological and paleobiological contexts, can be systematically approached by following a methodological workflow such as the one summarized in Figure 4. This diagram guides the reader from the initial analysis preparation, where the physical phenomenon of interest is defined and the turbulence model is selected, through to the final stages of

validation and result interpretation. At each step, key decisions are integrated, including the meshing method, CAD model preparation, boundary layer resolution, mesh quality criteria, and in more complex cases, the use of dynamic meshes or fluid–structure interaction.

The first aspect to consider in this process is the scale of the phenomenon being simulated and its range of influence. This can vary from the microscopic, such as biomolecules or cells, to the macroscopic, including large organisms or even atmospheric phenomena. In CFD simulations, the choice of mesh resolution is directly influenced by these scales, as they affect the accuracy of the results, especially when applying models such as RANS, which have known limitations in certain paleobiological contexts (Blocken, 2015; Schlünzen et al., 2011).

5.1 | Choosing between 2D and 3D models

In life sciences, 2D models are sometimes viewed as outdated compared to more modern techniques. However, models are simplifications that allow for the study and prediction of physical phenomena efficiently (Anderson et al., 2011). In engineering, simplifications are common, and 2D models play a fundamental role in numerical methods, especially in fluid mechanics, offering valuable and precise approximations. The decision between 2D and 3D simulations depends on the nature of the problem, available computational resources, and study objectives. In line with the decision-making framework illustrated in Figure 4 2D simulations are especially suitable when the primary objective is to test a specific

hypothesis or evaluate a physical principle prior to engaging in more computationally expensive 3D setups.

For 2D scenarios, such as flow in long channels, 2D models are highly effective. Additionally, their lower computational cost makes them useful for preliminary studies, validation and verification, particularly in analysing ground effect, hydrodynamic behaviour and flow coefficients.

When combining meshing strategies such as overset grids, transient simulation, dynamic deformation and hybrid meshes with accurate boundary layer resolution presents significant challenges, particularly in maintaining mesh quality metrics and a stable Courant number throughout the simulation; the definition of this dimensionless number is provided in Section 6 of the **Supporting Information**. In our case study of *Andrias japonicus* during suction feeding (Figure 6), biologically relevant movements such as jaw opening, hyoid expansion and fluid–prey interaction are explicitly captured through overset mesh dynamics and large-scale mesh deformation.

Starting with a 2D model can be the best initial strategy, as it allows for a highly accurate representation, potentially more realistic than a simplified 3D model that only animates the jaws. By simplifying the geometry to a sagittal plane (Figure 6), we were able to isolate the essential flow dynamics and boundary layer behaviour, while ensuring mesh quality and simulation stability over time. This approach results in sufficiently precise predictions to study the suction feeding phenomenon characteristic of these animals. Therefore,

a 3D CFD model is not necessarily superior to a 2D one; its effectiveness depends more on the level of detail and quality of the mesh than on dimensionality alone. Such 2D approaches can serve as powerful tools for validating theoretical expectations, optimizing mesh configurations and guiding experimental or 3D CFD efforts in later stages.

As a good practice, the overset mesh should have smaller elements than the background mesh (Figure 6e). If overset elements are too large, interpolation errors increase and numerical stability decreases, especially in the overlapping regions. A finer overset mesh ensures more accurate and stable coupling during transient simulations. Additionally, it is recommended to include layers on the surfaces of the geometry covered by the overset mesh, as these regions also exhibit relevant viscous effects (Figure 6d).

Meshes are also categorized by topology and purpose, including 2.5D meshes, which balance surface-based and volumetric discretization (Kong et al., 2019; Powalla et al., 2022). These are particularly useful in surface-bound applications where complete volume discretization is unnecessary but depth information remains relevant (Openfoam, 2024). Surface meshes serve as a precursor to 3D meshing, ensuring boundary accuracy before volume discretization. Meanwhile, hybrid meshes, integrating 2D, 2.5D and 3D elements, are widely applied in complex geometries such as anatomical structures, where capturing flow gradients and surface intricacies is vital. These advancements in meshing techniques improve computational efficiency while maintaining precision.

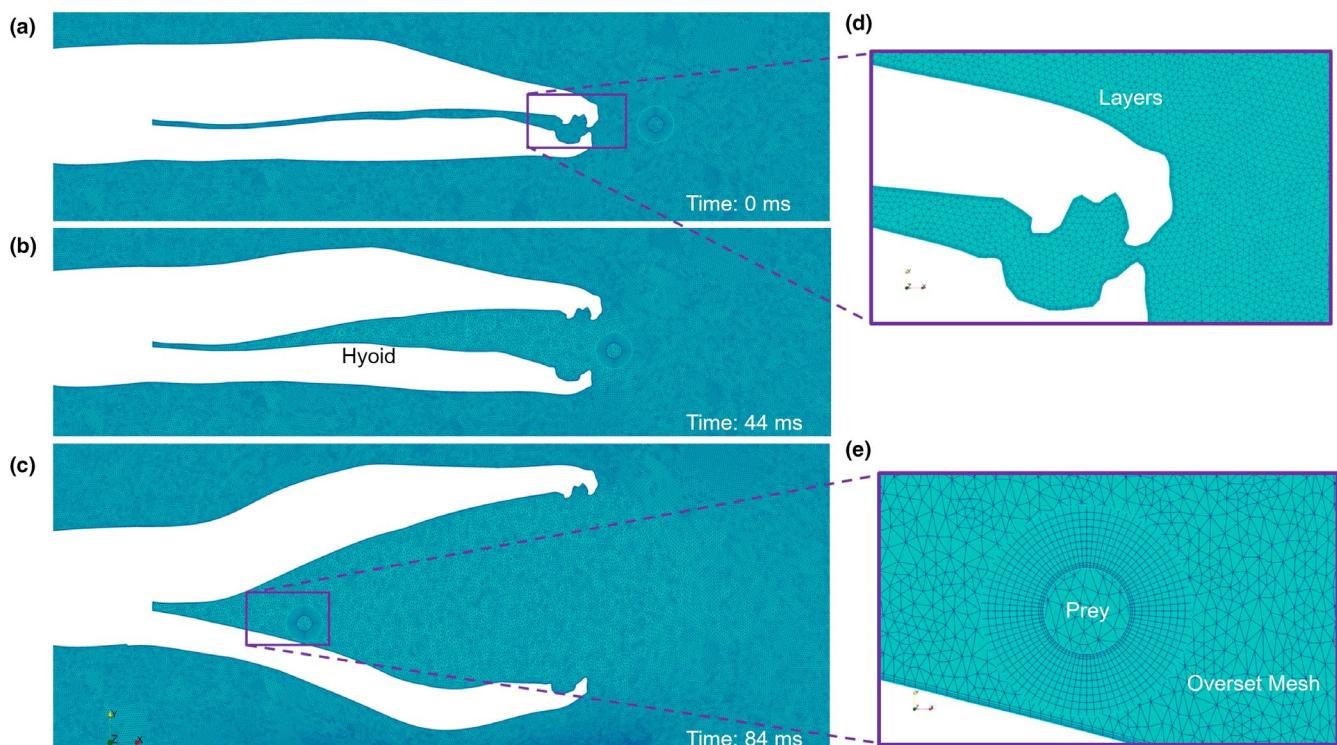


FIGURE 6 2D mesh of the central section of the *Andrias japonicus* cranium, obtained through micro-computed tomography, for simulating the suction feeding phenomenon. (a) Initial mesh with prey; (b) Intermediate jaw opening; (c) Continued jaw opening and hyoid expansion; (d) Mesh with layered refinement to capture boundary layer effects; (e) Prey mesh implemented using the overset technique.

5.2 | When to use 3D models in CFD?

Three-dimensional modelling in CFD becomes indispensable in situations where the nature of the flow, interactions and geometry demand greater precision and representation. For instance, flows that are inherently three-dimensional, such as those involving complex turbulence, chemical mixing in reactors or interactions around intricate bodies, require 3D models to fully capture the details of flow behaviour. Similarly, problems involving the interaction of multiple physical phenomena, such as the coupling of flow with heat transfer or chemical reactions in three dimensions, necessitate the use of 3D models to ensure these complex interactions are accurately represented.

Additionally, three-dimensional modelling is necessary for flows with inherently 3D effects and geometries lacking a predominant symmetry axis. This includes cases where complex flow behaviour cannot be accurately represented in 2D, such as vehicles, wind turbines, buildings and biological organisms such as snails, fish and asymmetrical species such as *Diplocaulus minimus* (Figure 7c). A 3D representation ensures that geometric influence on the flow is precisely captured, preserving key interactions that would be lost in a 2D simplification.

Moreover, when simulations demand high accuracy and detail, especially in critical areas of the domain, a 3D model provides a more detailed and reliable prediction of the physical phenomena under study.

5.3 | Dynamic meshes

One of the greatest challenges in meshing for numerical methods is enabling domain motion, which presents a significant computational difficulty. The main dynamic mesh techniques include mesh morphing, sliding interfaces and remeshing.

In the case of *Andrias japonicus* (Figure 8a), the remeshing technique was employed to accurately represent the fluid dynamics between the movements of the jaws, as it is well-suited to generate new elements in regions vacated by moving geometries, such as the jaws in this study. The remeshing is particularly recommended for complex or organic geometries.

In each of the dynamic mesh techniques, it is essential to use a sufficiently small time step size to prevent boundary movements from deforming or collapsing the mesh, as both mesh quality metrics and the Courant number must be monitored, as shown in the dynamic mesh workflow in Figure 4. In simulations involving rapid motions, an inadequate time step can cause abrupt deformations, negative cell volumes, and simulation divergence. However, an excessively small time step will substantially increase computational cost. Therefore, it is recommended to test different time step size values to achieve a balance between geometrical stability and computational efficiency. A common best practice is to initially estimate the time step size based on the displacement of the moving boundary, ensuring that it remains smaller than the length of the adjacent

mesh elements. In the case of *Andrias japonicus*, during the mouth opening motion, the moving boundary corresponds to the mouth surface, including the associated prism layers. Therefore, at each time step, the displacement of this boundary should not exceed the total length of the mesh cells adjacent to these layers (Figure 8b), in order to maintain mesh quality and geometrical stability.

5.4 | CAD models for paleobiology and CFD applications

These models can be in 2D or 3D and are used in paleobiology to represent both living and extinct specimens. In the context of CFD, they can also model the behaviour of fluids around these animals. The advantage of using CAD models in paleobiology is that they eliminate the risk of wear and damage to fossils, as they allow for non-destructive testing. The precision and detail of the digital models are nearly identical to the real objects. Therefore, this discussion will cover the geometric meshes that make up CAD models, their acquisition, and the types of analyses that can be performed.

The CAD model must balance geometric detail and computational efficiency. Simplifying features such as the pronounced scales of *Andrias japonicus* (Figure 9d,e) reduces mesh density while preserving accuracy. Proper spatial positioning on the symmetry plane (Figure 10a,b) minimizes computational costs. Additionally, adopting a hydrodynamic posture with raised limbs and aligned tails (Figure 9a–c) refines simulation realism, while defining the distance from the ground (Figure 9b), important for an accurate hydrodynamic analysis. To see the full recommendations, refer to Section 3 of the Supporting Information.

The CAD model of the animals used in CFD simulations does not necessarily need to be an exact replica of the real specimen. Simplifications are common in these models, as capturing every morphological detail would result in excessively complex meshes with high computational cost. Furthermore, biological variability, such as intraspecific variation, ontogenetic changes and sexual dimorphism, means that no two individuals within a species are morphologically identical. In this context, creating a perfectly faithful CAD model is not only impractical but also often unnecessary. The required level of morphological detail depends on the scientific question being addressed and the desired degree of precision. For details on how the geometries were obtained from fossils, refer to Section 3.1 of the Supporting Information. Well-justified geometric simplifications, supported by comparative analyses and sensitivity studies, can yield biologically meaningful results while optimizing computational efficiency.

As a general guideline, the overall proportions of the anatomical region of interest should be preserved, while unnecessary surface complexity should be avoided. Features such as hair, sharp angles, small holes and irregularities should be removed or smoothed to improve mesh quality. In the case of *Andrias japonicus*, we removed details such as scales and claws, and in the 2D model, we also excluded the fangs. Small morphological elements

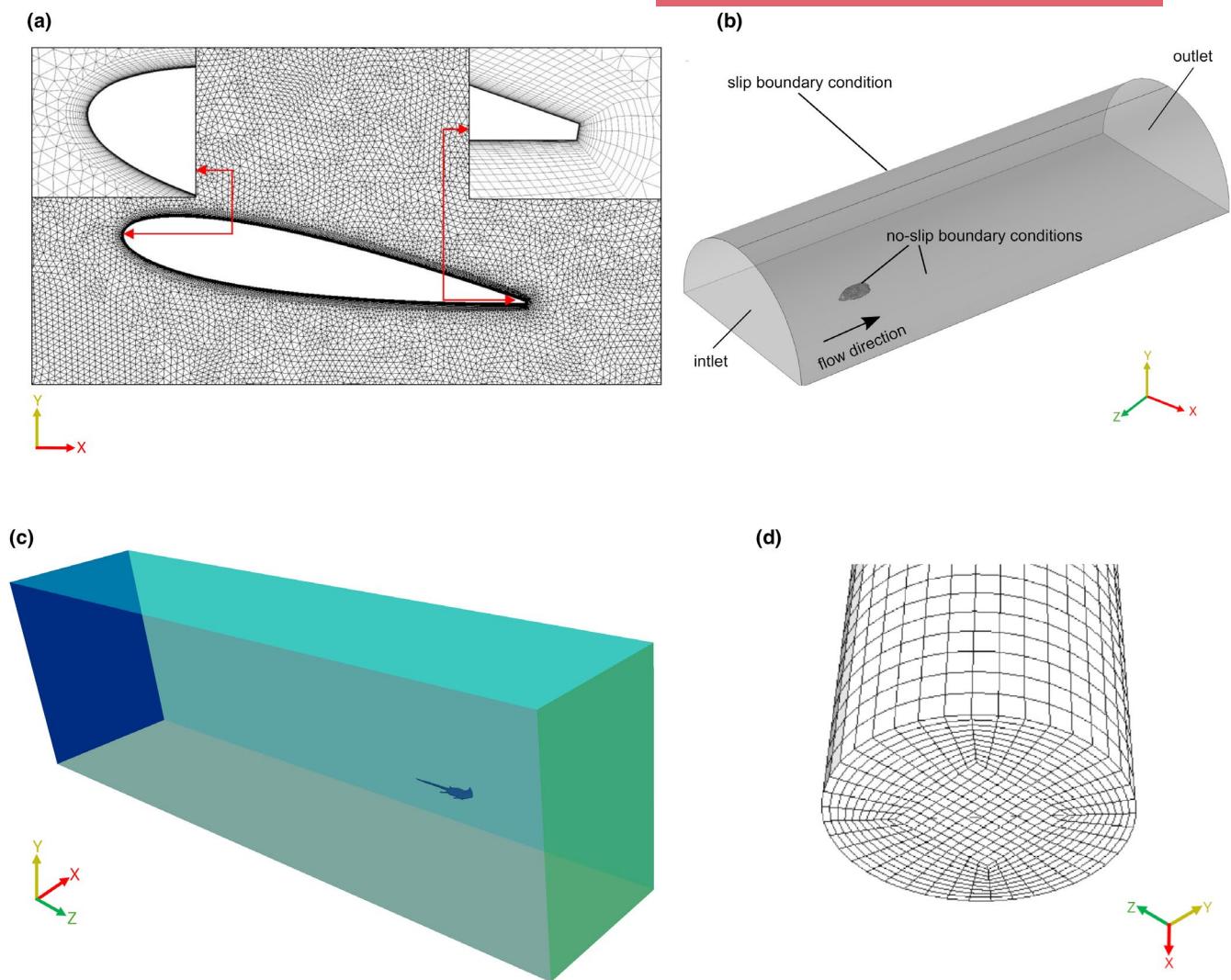


FIGURE 7 External shapes of the computational domain. (a) Shape of the computational domain for a 2D airfoil (Satyam et al., 2024), (b) atypical computational domain shape for *Protocinctus mansillaensis* taken from (Rahman, 2017), (c) *Diplocaulus minimus* computational domain representation. Typical hexahedral computational domain shape; and (d) cylindrical geometry of the computational domain of a 3D tube taken from (Hernandez-Perez et al., 2011).

tend to increase mesh density without contributing meaningfully to the fluid simulation; thus, it is advisable to simplify or eliminate them. Only planar or smoothed surfaces should be retained when feasible.

5.5 | Computational domain

5.5.1 | Shape

To understand the dimensions of the computational domain, it is useful to compare it to the test section of a wind tunnel. In CFD, similar principles apply, such as blockage ratio and domain length (Figure 11c). The computational domain acts as a digital equivalent of a wind tunnel test section, ensuring that the simulated airflow is not artificially constrained. Additionally, since CFD simulations are often validated through wind tunnel

experiments, maintaining a cubic or appropriately scaled domain shape helps to improve consistency between numerical and experimental results.

For some two-dimensional simulations, such as those involving airfoils, circular domains are used (Figure 7a). In some palaeontology studies, various domain shapes are used to reduce the number of elements required. This is because their cross-sectional area is smaller than that of a cube, which optimizes computational efficiency while maintaining accuracy (Figure 7b). However, the most widely used and recommended shape for its benefits is the hexahedron. Its widespread use is due to its geometric simplicity, ease of boundary condition implementation, numerical stability, computational efficiency and compatibility with structured meshes. Additionally, it offers scalability in HPC environments and adaptability for localized refinements (Figure 7c).

Although perfect symmetry does not exist in the animal kingdom, following best practice and within the scope of this example

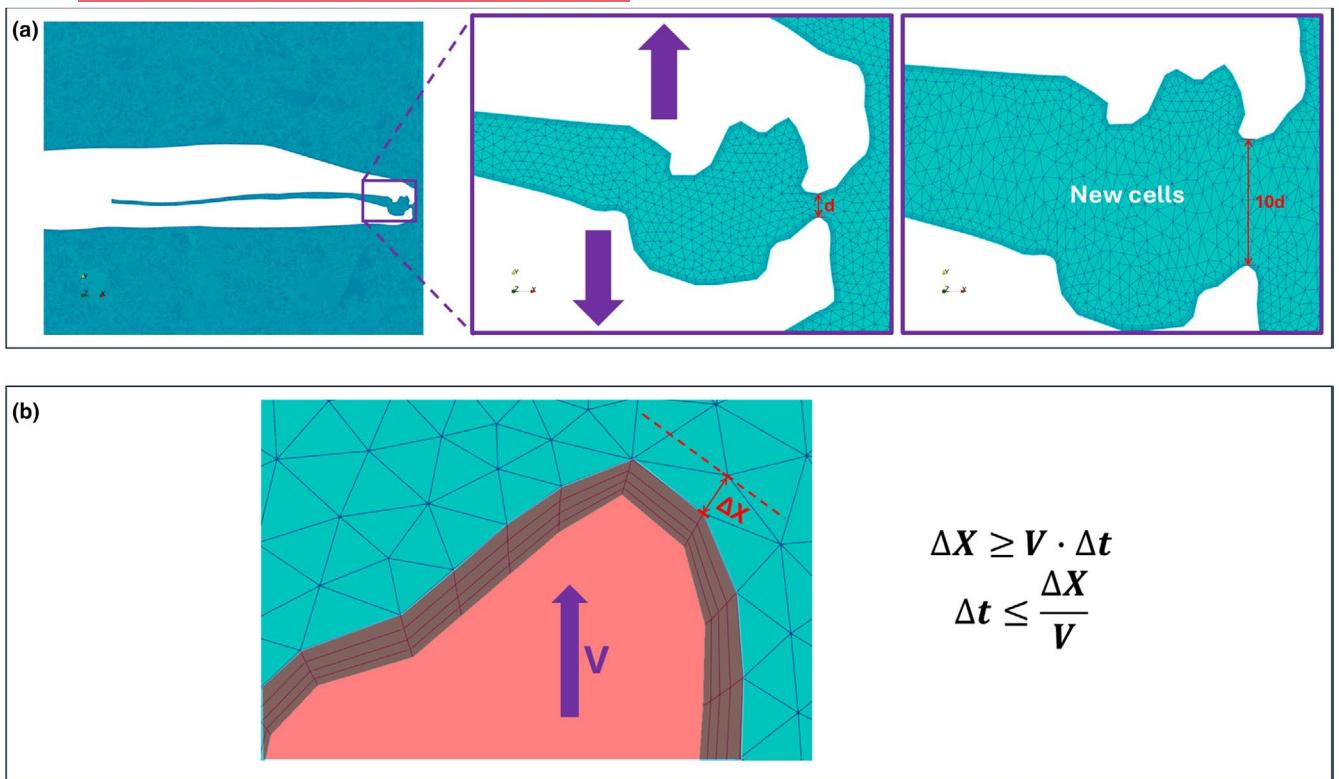


FIGURE 8 (a) Remeshing example in *Andrias japonicus*; (b) Stability condition for mesh morphing, remeshing, and sliding interfaces in dynamic meshes, where ΔX represents the distance between the moving boundary (last prism layer) and the first neighbouring volume.

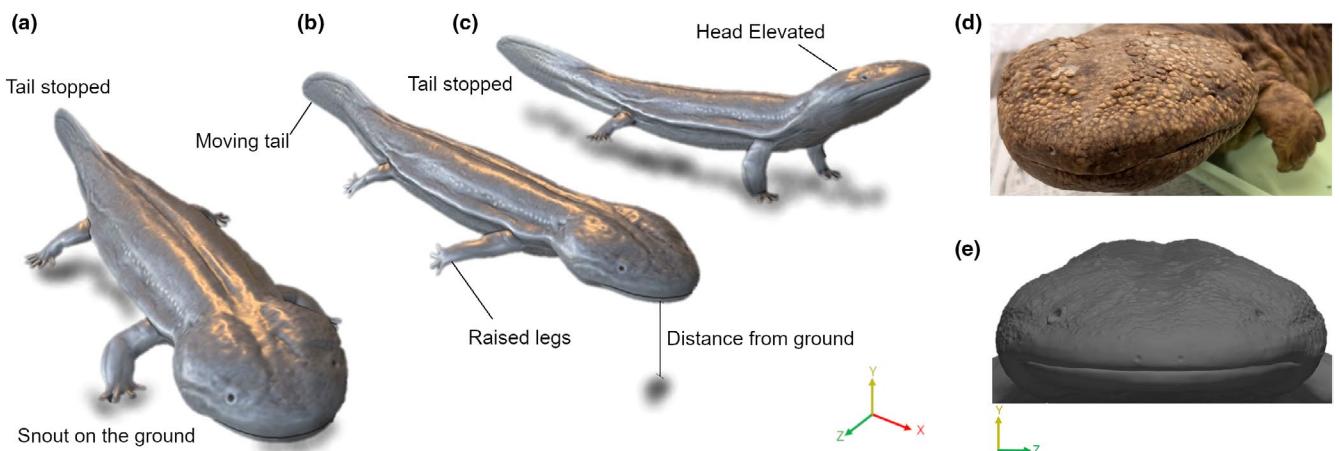


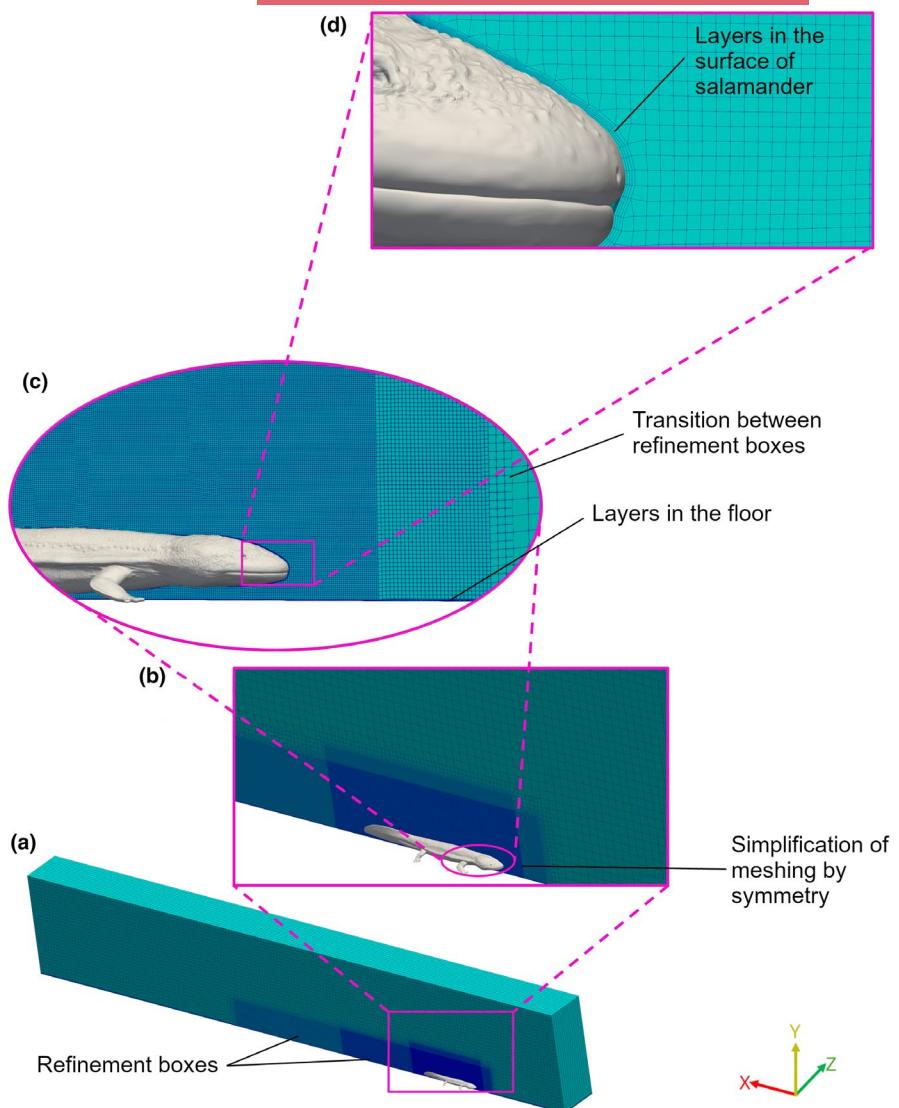
FIGURE 9 *Andrias japonicus* CAD position: (a) relaxed pose, (b) swimming pose, (c) stalking or breathing position, (d) real *Andrias japonicus*, and (e) CAD *Andrias japonicus*. Digitizing Fossil CAD Models.

on *Andrias japonicus*, the organism is assumed to be completely symmetrical. This assumption allows for a computational domain simplification with hexahedral shape, where only half of the mesh is constructed to optimize computational resources. In the case of *Andrias japonicus*, since the entire body is simulated in 3D, the model was positioned along the axis of symmetry and aligned with the computational domain. (Figure 10a,b).

5.5.2 | Dimensions

The general domain dimensions define the space in which the fluid will be present, influencing the number of cells and, consequently, the computational resources required to solve the problem. Beyond mesh convergence analysis and quality parameters, additional challenges are associated with domain size, mesh dimension and the distance

FIGURE 10 (a) Full computational domain of *Andrias japonicus*, general dimensions $1 \times 2.5 \times 14.9$ m, (b) refinement box details surrounding the body, (c) frontal view of the mesh near the salamander's head, and (d) close-up of the surface mesh layers on the salamander's skin.



between the geometry of interest and the domain walls. These factors often become apparent only after conducting simulations, particularly during the post-processing of results. For this reason, it is important to build a first preliminary mesh. When the computational domain is significantly larger than the object of interest, and the geometry occupies less than 5% of the total volume, as is common in certain 3D configurations with its characteristic length oriented vertically and exposed to air velocities exceeding 5 m/s under standard atmospheric conditions, it is essential to analyse potential errors related to domain shape and size (Abu-Zidan et al., 2021; Blocken, 2015; Marcé Nogué et al., 2015). These include the Atmospheric Boundary Layer (ABL), Blockage Ratio (BR), Local Venturi Effect (LVE), and Global Venturi Effect (GVE). Preliminary simulations are necessary to observe how the pressure and velocity of the incoming flow behave as they approach the domain boundaries, and to assess whether significant errors occur in these regions. This behaviour is illustrated in more detail in Section 4 of the *Supporting Information*.

In the animal kingdom, where organisms can alter their shape or posture, these factors must be carefully considered when simulating animals in environments where air acts as a fluid, particularly

at high altitudes, as seen in birds. Similarly, observing these aspects is good practice in simulations of large terrestrial and marine vertebrates, such as colossal dinosaurs, proboscideans (elephants) and cetaceans (whales). While these organisms generally exhibit a predominantly horizontal morphology, vertical movements or posture-induced changes in their characteristic length (see Section 4.2 of the *Supporting Information*) necessitate evaluating the same computational domain errors to maintain simulation accuracy (Abu-Zidan et al., 2021; Blocken, 2015; Blocken et al., 2011). Boundary and inertial conditions in a CFD simulation define the flow behaviour at the domain limits and its initial state. These include parameters such as inlet velocity, outlet pressure, fixed or moving walls and symmetry boundaries. They are crucial for obtaining realistic and reliable approximations and are highly dependent on the mesh geometry, especially when deciding which surfaces are treated as walls or boundary conditions. As a best practice, it is advisable to explore how the solution varies with different setups, depending on the objective of the analysis, as suggested in Figure 4.

In the case of *Andrias japonicus*, whose predominant body shape is horizontal and for which a velocity of 1 m/s was imposed,

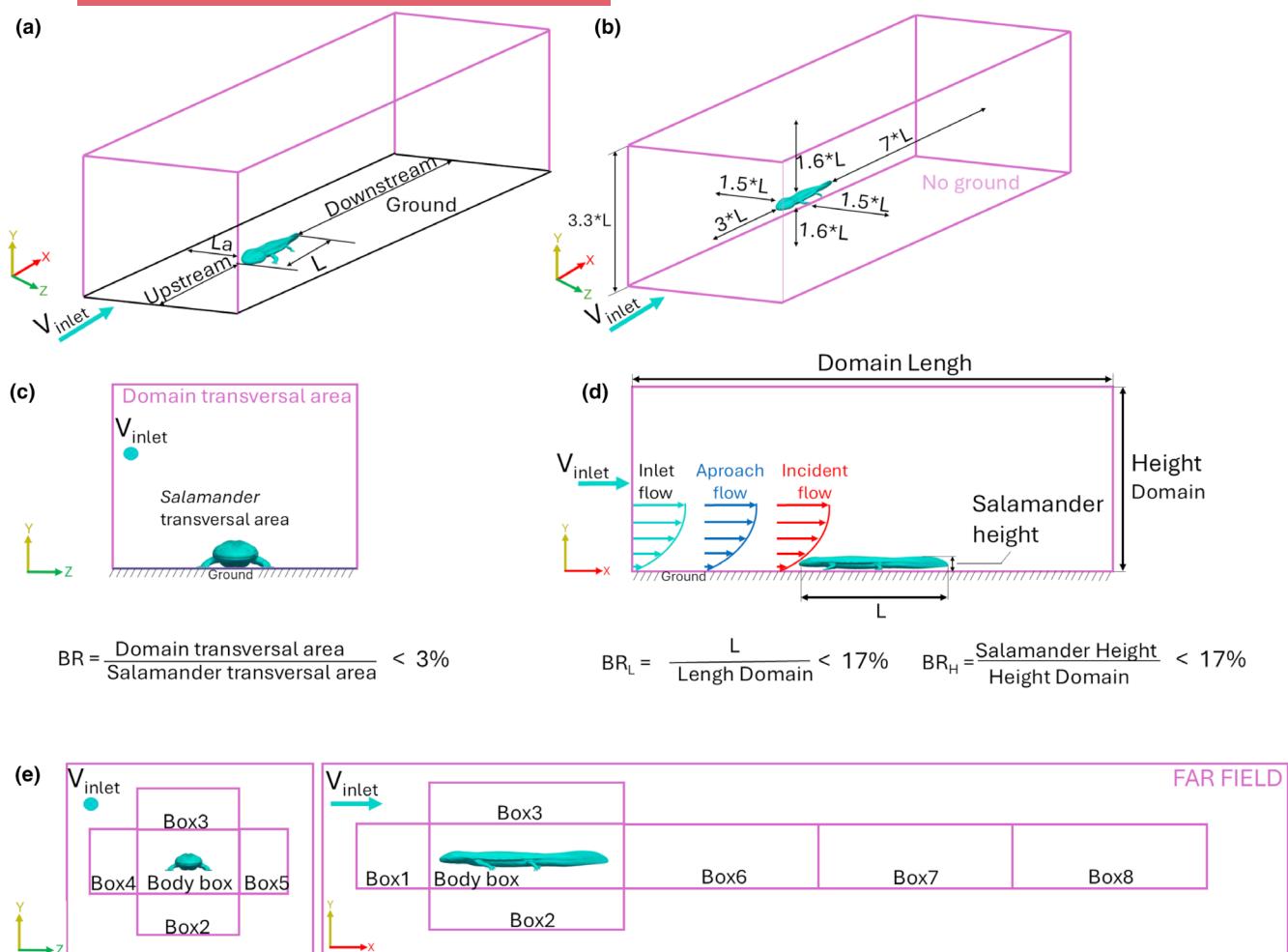


FIGURE 11 Example of domain dimensions in the amphibian *Andrias japonicus*. (a) Dimensions surrounding the geometry of the animal in a resting position on the ground; (b) dimensions surrounding the geometry of the amphibian in a swimming position, maintaining a distance from the ground; (c) blockage ratio; (d) velocity profiles upstream of the animal when positioned on the ground, adapted from (Blocken et al., 2007), and (e) refinement boxes around the body. (L =Salamander length, La =Lateral distance).

the computational domain must be adapted accordingly; for detailed information between boundary and initial conditions, refer to Section 4.3 of the [Supporting Information](#). It is recommended to start with the dimensions specified in Figure 11b to create the initial mesh. Then, you can progressively reduce the domain size based on the pressure and velocity contours to optimize computational efficiency. This will largely depend on parameters such as the inlet velocity and the blockage ratio.

The use of refinement boxes optimizes mesh resolution and reduces computational costs by refining critical areas such as the boundary layer and wake region (Figure 11e). The Multiblock meshing method ensures progressively finer meshing closer to the geometry, with the body box containing the highest resolution to capture flow behaviour accurately. In *Andrias japonicus*, additional refinement is placed upstream to capture disturbances before the fluid reaches the animal, and finer meshing is applied where vortex structures form (Figure 11d).

In the wake region, multiple refinement levels allow for capturing large-scale vortex structures and their gradual dissipation as

the flow moves downstream. This structured refinement maintains numerical stability while balancing accuracy and computational efficiency. For a more detailed explanation of each refinement box, refer to Section 5 of the [Supporting Information](#).

5.6 | Dimensionless parameter y^+

This dimensionless parameter allows for a correct prediction of the boundary layer detachment zones by means of the mesh; these in turn generate the first vortex structures, which in turn end up forming the wake. This approach is applied in external flow analyses to characterize hydrodynamic or aerodynamic coefficients in geometries such as transport vehicles, vehicle accessories and living or extinct organisms (Salim & Cheah, 2009).

The y^+ parameter arises as a numerical calculation that indicates the quality of the distance to the wall (boundary layer) representation from the surface to the far field. Turbulent flows are significantly sensitive to the presence of a wall, and the y^+ parameter exists

to account for this sensitivity. This number is defined by the following expression:

$$y^+ = \frac{y_p u_r}{v} \quad (1)$$

where y_p is the distance between the centroide node at the surface of the wall and the node where the first sub-layer ends, $u_r = \sqrt{\frac{\tau_w}{\rho}}$, τ_w is the shear stress in the wall, ρ is the air density and v is the dynamic viscosity.

It is generally used to define how coarse or fine a mesh is for a particular flow pattern, and it is for this reason that depending on the turbulence model used a different y^+ is recommended (CFD-Online, 2018).

5.6.1 | Importance of the boundary layer

The boundary layer corresponds to the region near the walls, for example, the skin of the geometry of animals or the floor representation, where the flow behaviour undergoes a complete transformation, especially in the direction normal to the surface (Figure 5b). The theory of the boundary layer, introduced by Ludwig Prandtl in 1904 (Prandtl, 1904), states that in a very thin region adjacent to the wall of a body, the effects of viscosity are significant and manifest as a velocity gradient (White, 2009). Within this layer, the fluid velocity increases from zero to approximately 99% of the free stream velocity due to the no-slip condition; for detailed information, refer to Section 2 of the Supporting Information.

5.7 | How to capture boundary layer in CFD?

There are two main approaches to capturing the boundary layer in CFD simulations: fully resolving the boundary layer or using wall functions. The first method consists of generating a fine mesh that allows direct resolution of the turbulent structures all the way to the wall. To achieve this, the mesh must be structured in multiple layers of prismatic cells (Figure 5) with the longest dimension aligned to the flow direction, ensuring an accurate representation of the velocity gradient near the wall (Figure 10d). The centre of the first cell must be positioned inside the viscous sublayer, which is the region where molecular viscosity dominates over turbulence (Figure 5b). This approach requires keeping the y^+ parameter close to 1 in the surface of the wall, meaning the first cell height must be sufficiently small to resolve the near-wall effects without relying on empirical models (Gersten, 2009). As a result, this method provides highly accurate predictions of shear stress and boundary layer detachment but significantly increases computational cost due to the high mesh density required near walls. It is commonly used for low-Reynolds-number flows, such as those found in simulations of terrestrial animals, swimmers, and bio-inspired robotics, where precise drag and lift calculations are essential.

The second method involves using wall functions, which allow for a coarser mesh by avoiding the need to resolve the viscous sublayer. Empirical equations are used to model the flow behaviour in this region, bridging the gap between the wall and the fully developed turbulent region (Liu, 2017). In this approach, the first cell must be positioned in the logarithmic region of the boundary layer (Figure 5b), where velocity follows a well-established log-law profile. This significantly reduces computational costs by allowing larger cell sizes and avoiding excessive refinement near the walls. Wall functions are typically used in high-Reynolds-number flows, such as those found in supersonic and compressible flow simulations (Tominaga et al., 2008). In nature, only a few animals, like the peregrine falcon (*Falco peregrinus*), can reach such speeds, with Mach numbers around 0.31 during a dive (Tucker, 1998). However, in cases where boundary layer detachment plays a determining role, such as in the hydrodynamics of the *Andrias japonicus* salamander, this method may introduce inaccuracies, as it does not fully resolve the separation points along the back and tail where negative pressure gradients occur.

Studies across a range of applications show that for direct boundary-layer resolution using RANS or LES models, maintaining $y^+ \approx 1$ is critical for accuracy (Patel et al., 1985) Conversely, in highRe, wallfunction approaches remain accurate when y^+ is in the 30–60 range (Salim et al., 2010). Recent works further supports using wall functions with $y^+ \approx 30$ –300 (Yang et al., 2024). Depending on the target y^+ value (Table 3), the process may become iterative, as illustrated in workflow diagram (Figure 4).

5.7.1 | Determining the first layer height-

This initial layer height $y = 2y_p$ can be calculated as:

$$y_p = \frac{y^+ v}{u_r} \quad (2)$$

where u_r is the friction velocity (Equation 1), which depends on the flow conditions.

$$\tau_w = \left(\frac{1}{2} \rho V_\infty^2 \right) C_f \quad (3)$$

τ_w is the shear stress in the wall and C_f is the skin friction coefficient

$$C_f = [2\log_{10} Re - 6.5]^{-2.3} \quad (4)$$

There are two approaches to determining the height of the first layer. The first involves a series of calculations based on theoretical principles and fundamental equations to obtain y_p (Figure 5b), while the second follows a more practical, iterative approach. In some cases, the calculated values may be extremely small, depending on the flow regime.

The iterative process for determining the first layer height in CFD begins with an initial simulation using a mesh based on prior

TABLE 3 Recommended y^+ values across flow regimes, turbulence models, and biological or engineering applications.

Reynolds number	Flow regime	Target y^+	Modelling approach	Example application
<1000	Laminar	Not critical	Direct resolution (no turbulence)	Flow around small larvae, micro-swimmers, or creeping flow in thin channels
1000–10,000	Transitional	≈1 or higher	Direct resolution or low-Re models	Amphibian suction feeding (e.g. <i>Andrias japonicus</i>), insect-scale swimmers
10,000–100,000	Low turbulent	≈1	Wall-resolved RANS or LES	Swimming fish (e.g. trout), bioinspired robotic fins
>300,000	Fully turbulent	>30	Wall functions (RANS, URANS)	Flow over buildings, wings, dolphin body flow, or large aquatic animal locomotion
>1000,000	High turbulence	30–300	High-Re RANS with wall functions	Aerodynamic simulations, wind turbines or flow in urban environments

experience. According to best practices, for animals ranging from 0.3 to 3 m in length, an initial layer thickness of approximately 0.025 mm is generally effective. After the initial simulation, the y^+ values at the wall are extracted from the results, and the mesh is refined accordingly. If y^+ is too high, the first layer height should be reduced; if it is too low, it can be increased to improve computational efficiency. This iterative process continues, adjusting the mesh and rerunning the simulation until the target y^+ value is achieved, ensuring proper boundary layer resolution for the selected turbulence model. In general, maintaining a y^+ value below 10 is recommended, with values closer to 1 providing optimal accuracy (CFD-Online, 2018).

In the case of *Andrias japonicus*, an average of 12 layers was obtained (Figure 10d). The y^+ value varies across the surface, with most values remaining close to 1 (Figure 12a,b), making it imperative to prioritize its control in key areas, which are often regions of negative pressure where the boundary layer detaches. In the case of our salamander *Andrias Japonicus*, this would be along its back and tail.

5.8 | Mesh quality assessment

Once the mesh generation is complete, it is essential to verify the quality parameters. In the case of transient simulations, the dimensionless Courant number serves as a key indicator of numerical stability in time-stepping, with values ideally kept below 1 to prevent instabilities (Table 4).

It is important to emphasize that these parameters are purely geometric and serve only as indicators of mesh quality. However, meeting all these criteria does not necessarily guarantee that the mesh is final or optimal for convergence. Achieving an effective mesh requires additional considerations beyond geometric quality, such as numerical stability and solver-specific requirements.

These quality parameters are derived from the most widely used CFD software, including the commercial Ansys Fluent (ANSYS Inc., Canonsburg, PA, USA) and OpenFOAM, which is available in

two versions: [OpenFOAM.org](#), developed and maintained by The OpenFOAM Foundation in London, UK, and [OpenFOAM.com](#), managed by OpenCFD Ltd. in Bracknell, UK, as an open-source platform. Additionally, COMSOL Multiphysics (COMSOL Inc., Burlington, MA, USA) also implements similar mesh quality diagnostics.

5.9 | Relation of the mesh with turbulence models

FEA meshes are smaller than CFD meshes since only the object itself is meshed rather than the entire domain. In CFD, mesh density is highly influenced by the turbulence model used (Figure 13c). The RANS model allows for a coarser mesh by averaging turbulent fluctuations, while LES requires finer refinement to capture smaller turbulent structures, particularly near walls and within flow cores. The DES model combines both approaches, leading to increased cell counts in complex flow regions.

DNS (Direct Numerical Simulation) fully resolves all turbulence scales without simplifications (Figure 13d), demanding an extremely dense mesh and substantial computational power, making it impractical for large-scale simulations, particularly in animal studies. The FSI (Fluid-Structure Interaction) approach, which integrates CFD and FEA, requires refined meshes in critical areas to accurately model interactions between fluid flow and structural deformation. Therefore, the choice of turbulence model directly affects mesh size, influencing computational cost and simulation accuracy (Gibson et al., 2021).

5.10 | Mesh density and convergence

It is commonly assumed that increasing the number of cells in a mesh leads to better results. However, there is a practical limit to mesh refinement, as excessive density not only increases computational cost but also reaches a point where further refinement does not

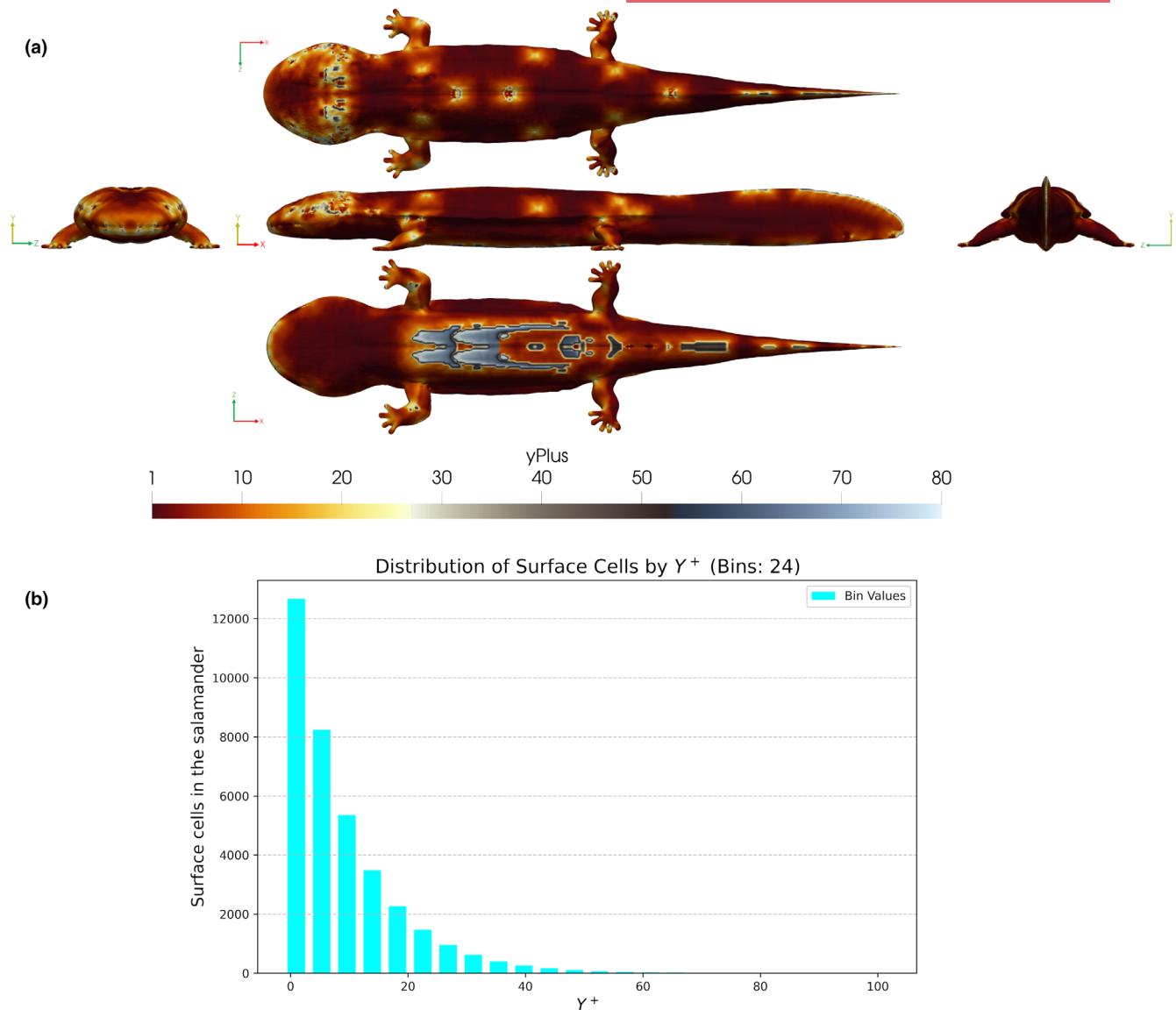


FIGURE 12 (a) γ^+ distribution over the entire surface of *Andrias japonicus*. The abdominal region and some parts of the tale, which is in contact with the substrate, lacks boundary layers, and (b) histogram of γ^+ distribution over the salamander's surface with 24 bins.

significantly improve accuracy. The key question is: When should we stop increasing mesh density to balance precision and computational feasibility?

To this end, it is essential to perform a convergence analysis. This process allows us to determine the point at which further mesh refinement no longer produces significant improvements in accuracy, while ensuring efficient use of computational resources. In the case of the mesh generated for the salamander *Andrias Japonicus*, it is observed that with 9 million elements (Figure 13a), the drag coefficient results do not vary by more than 5% as the number of cells increases. Therefore, this mesh size is considered the most efficient, as it does not require additional computational resources without providing significant improvements in accuracy. This 5% threshold marks the point where convergence begins; if greater precision is required for more sensitive comparative studies, the number of elements can be further increased. Consequently, it can be stated that with 9 million

elements the results have reached convergence. As a best practice, it is recommended to generate at least three meshes: a coarse, a medium, and a fine one, as shown in Figure 4.

However, in some cases, convergence is not obtained; in contrast to what was observed in the salamander example, alternative methods are required to achieve it. The first is called Uniform Refinement, which subdivides all mesh cells. However, this increases the number of cells exponentially along with computational cost, making it ideal for coarse meshes within the asymptotic range where error predictably decreases with refinement (Figure 13a). In contrast, Non-Uniform Refinement adds cells only in regions with higher flow dynamics, typically around the geometry of interest and in the wake (Baker, 2005), using refinement boxes as explained in the “Refinement Boxes” section. This approach improves local accuracy without globally increasing the mesh size, making it useful for complex geometries and variable flow conditions. However, in more

TABLE 4 Quality metrics most commonly used in the most widely used CFD simulation software.

Quality metric	Explanation	Desirable range	Source
Aspect ratio	Measures the proportion between the element dimensions. Values close to 1 indicate a well-proportioned element	1–3 (optimal), <5 acceptable	Ansys Fuent (2024)
Orthogonality	Indicates how perpendicular a cell is relative to its neighbours. Ideal values are close to 90°, reducing numerical diffusion	~90°	Ansys Fuent (2024)
Skewness	Assesses how much the cell shape deviates from an ideal shape (cube or tetrahedron). Lower skewness is better.	<0.25 ideal, <0.5 acceptable	Ansys Fuent (2024)
Smoothness	Evaluates the size transition between adjacent cells. Smooth transitions help maintain solver stability	Gradual changes between cells	Ansys Fuent (2024)
Determinant of the Jacobian Matrix	Measures how distorted an element is in 3D space	Values near 1 indicate low deformation	Ansys Fuent (2025)
Non-orthogonality	Measures the angle between face normals and the vector connecting cell centers. Lower values reduce interpolation errors	<70° (lower is better)	OpenFOAM (n.d.)
Cell Volume Ratio	Compares the volume of neighbouring cells. Large differences can destabilize the solution and reduce accuracy	<2 ratio between neighbours	OpenFOAM (n.d.)
Cell Determinant	Like the Jacobian determinant, evaluates cell deformation in 3D to detect inverted or collapsed cells	~1; avoid values near 0 or	OpenFOAM (n.d.)
Flatness	Evaluates how flattened a cell is, particularly in prisms or hexahedra. High flatness indicates nearly collapsed faces and poor cell quality	<0.5 preferred (closer to 0 is better)	Ansys Fuent (2024)

specific situations where convergence is not achieved, an alternative strategy is to increase the number of cells based on Richardson Extrapolation (Roache, 1998), a method that estimates the discretization error by comparing solutions at different mesh resolutions and extrapolating towards an asymptotic limit. Another widely used approach is the Grid Convergence Index (GCI) (Slater, 2021), which provides a quantitative measure of the numerical uncertainty by assessing how the solution changes with successive mesh refinements. These methods ensure that mesh refinement leads to a reliable and converged solution rather than merely increasing computational cost.

For more complex simulations, like transient ones or those involving intricate internal flows, achieving convergence can be challenging. This occurs due to an ‘oscillatory convergence’ phenomenon, where simulation results oscillate between different convergent states instead of settling into a single solution (Ramponi & Blocken, 2012). Such situations can arise in complex flow problems, such as those found in urban environments, rotating machinery, and vehicle aerodynamics. Multiple factors can influence model convergence.

To address this issue, strategies such as adjusting the number of preceding iterations to 10^{-6} can be employed to ensure numerical convergence, which is considered a best practice, though it should not be confused with mesh independence (Figure 13b) (Ramponi & Blocken, 2012), validating with a more robust computational

model (such as Large Eddy Simulation, or LES) or using a detailed experimental setup like Particle Image Velocimetry (PIV) (Revuz et al., 2012). However, these types of simulations are uncommon in the paleobiological field, especially when characterizing the hydrodynamic performance of a specific animal. In such cases, only steady-state external flow is analysed. But it is necessary to recognize that traditional mesh convergence is not the only method for computationally validating a CFD simulation.

5.11 | Mesh generation software

Both open-source and commercial CFD software solutions handle numerical calculations, often including integrated or standalone meshing tools essential for preprocessing. Around 80 meshing programmes exist (Vascular, 2009), but the most widely used ones are listed below (Table 5).

6 | DISCUSSION AND CONCLUSIONS

The CFD simulation process is highly iterative, with each step requiring specific conditions that often need revisiting previous stages for adjustments and refinements. From the acquisition and preparation of the CAD model to the final validation of the

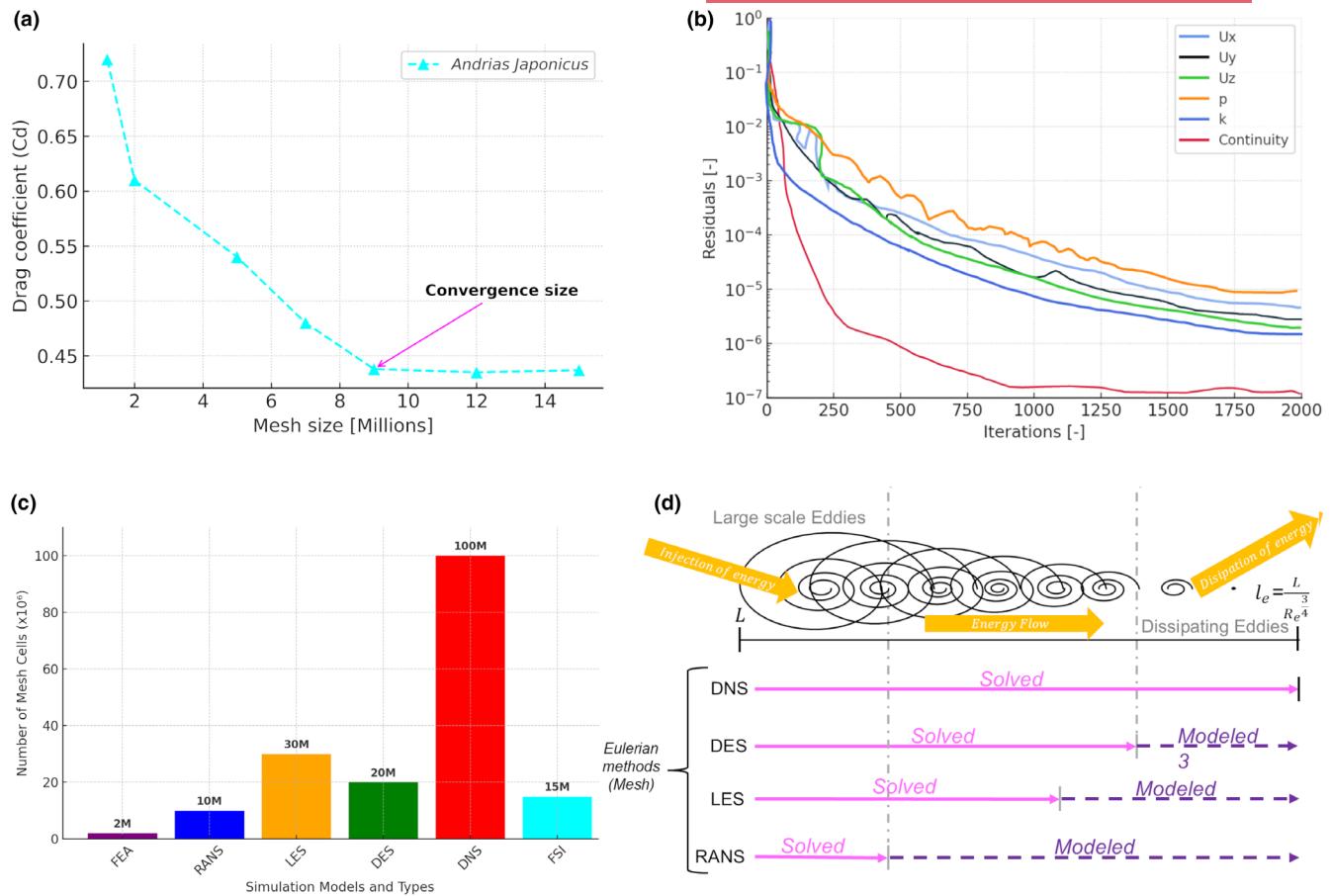


FIGURE 13 (a) Convergence of the Drag Coefficient, (b) residuals between iterations (SimScale, 2018), and (c) impact of model type and simulation method on the number of cells. The number of cells in the different meshes was determined using the geometry of a 2D *Andrias japonicus* salamander as a reference, and (d) Prediction methods in CFD, adapted from (Ansys Fuent, 2024).

results, the mesh plays a crucial role in ensuring both the accuracy and stability of the simulation. As illustrated in the workflow diagram (Figure 4), the steps highlighted in cyan indicate where the meshing process is directly involved, underscoring its central role in the overall workflow.

The process begins with the definition of the fluid analysis, followed by the selection of the appropriate turbulence model and dimensionality (2D, 2.5D, or 3D). Next comes mesh generation, where careful consideration of the boundary layer is essential. The target y^+ value, mesh quality criteria, and convergence are key elements to ensure reliable numerical predictions.

Systematically addressing each of the questions presented in the workflow of Figure 4 naturally leads to an iterative process. Once the final mesh is achieved and the corresponding simulation is executed, the results can be representative and provide a close approximation of the system's real behaviour. These results can be used to perform multiple analyses and even test previously formulated hypotheses. This iterative approach highlights the substantial influence of meshing on CFD simulation quality, establishing it as a critical component that must be handled with precision to achieve reliable and representative results (Blocken et al., 2007). However, real experimentation remains the most robust method for validating

these predictions or, if necessary, calibrating and adjusting the CFD model.

From a biological perspective, while expertise in numerical simulations may not be fully developed, collaboration with engineers can be significantly strengthened through well-constructed CAD models (see Section 3 of the Supporting Information). Providing accurate ecological information is essential, especially when studying prehistoric specimens, where understanding the physical properties of ancient fluids, such as atmospheric composition or water, supports physical consistency in the model. Although variations in viscosity or density may influence results, their effect is typically minor compared to that of morphological factors. Additionally, whenever possible, experimentation with extant species that share functional or morphological similarities can serve as a valuable reference to validate computational results. However, the greatest challenge lies in biologically interpreting the outcomes of these simulations with a critical and scientifically sound perspective. CFD generates highly detailed visualizations of fluid interactions, but extracting meaningful evolutionary, ecological, or biomechanical insights requires interdisciplinary dialogue between biologists and engineers to ensure that the conclusions drawn are both computationally robust and biologically plausible.⁷

TABLE 5 Meshing software.

Software	Company	Type of software	Friendly interfaz	Hybrid meshes	Adaptive meshing	Prisms to boundary layer	Structured+ unstructures meshes	Mesh automation	Quality metrics	Support+ documentation
ICEM	Ansys, Inc	Commercial	More or less	Yes	No	Yes	Yes	More or less	Yes	Yes+
Meshing	Ansys, Inc	Commercial	Yes	Yes	More or less	Yes	Yes	Yes	Yes	Yes
Gambit	Ansys, Inc	Commercial	More or less	Yes	No	Yes	Yes	No	More or less	No
Gmsh	Christophe Geuzaine & Jean-François Remacle	Open source	More or less	Yes	No (manual refinement possible)	Yes	Yes	No	Yes	Yes
T-grid	Ansys, Inc	Commercial	More or less	Yes	No	Yes	Yes	More or less	Yes	Yes
ANSA	BETA CAE Systems	Commercial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
STAR-CCM+	Siemens	Commercial	Yes	Yes	More or less	Yes	Yes	Yes	Yes	Yes
Pointwise, Inc	Pointwise, Inc	Commercial	More or less	Yes	No	Yes	Yes	More or less	Yes	Yes
CF Mesh+	Creative Fields	Open source	More or less	Yes	More or less	Yes	Yes	More or less	Yes	Yes
Ennova	Ennova Technologies, Inc.	Commercial	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SnappyHexMesh	OpenFoam	Open source	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes+
BlockMesh	OpenFoam	Open source	No	No	No	Yes	No	Yes	Yes	Yes
Discretizer	Discretizer	Open source	Yes	No	No	No	Yes	No	No	More or less
enGrid	nGrid and collaborators	Open source	More or less	No	Yes	Yes	More or less	No	More or less	More or less

7 | CONCLUSIONS

To represent living organisms in biomechanical simulations, the mesh design must adapt to the complexity and morphology of the structure being analysed. The main challenge is to find a balance between mesh resolution and computational efficiency.

Simple geometries, such as isolated bone sections, can be modelled with basic meshes, reducing computational costs without compromising accuracy. However, more complex structures, including soft tissues, joints, and flexible elements, require higher refinement to accurately capture their deformable behaviour and interaction with the flow. While this increases simulation time, it provides more precise biomechanical analyses. In this context, unstructured hybrid meshes are particularly effective for modelling living organisms, allowing for localized refinement in key regions without excessive computational cost.

Meshing is more than just a step in CFD; it is the core of simulations and the most iterative and time-consuming process (Figure 4). Even with a well-defined CAD model and properly established boundary and initial conditions, a poorly designed mesh will compromise the accuracy of the results. While automatic meshing tools can be useful, their solutions are not always optimal, lacking the ability to intelligently adapt to complex geometries. Developing expertise in meshing techniques is essential for achieving accurate predictions.

Computational simulations have revolutionized the study of biomechanics, enabling researchers to infer from fossil evidence how extinct species moved, interacted with their environment, and adapted over time, thus complementing traditional paleontological methods. However, these remain approximations, constrained by data availability, model accuracy, and computational limitations.

Even with 3D models, reality is never fully captured. These simulations require greater computational resources and may introduce new sources of error. In many cases, a well-designed 2D simulation with appropriate turbulence models, transient conditions, and correctly formulated boundary and initial conditions can provide more accurate predictions than an overly simplistic 3D model. In biological research, assuming that 3D simulations are always superior is a misconception, as accuracy and computational efficiency depend on selecting the appropriate level of complexity for each study.

The validation of CFD models with experimental data or living species is fundamental for improving results' reliability. Only through proper validation can a model be considered robust and successful, as illustrated in Figure 4. For extinct organisms, simulations can be compared with experiments on modern evolutionary relatives, providing a biologically coherent reference framework. Additionally, interdisciplinary collaboration between palaeontologists and engineers improves model accuracy, ensuring that computational findings are both biologically meaningful and methodologically robust.

The integration of artificial intelligence (AI) into CFD is transforming the field, optimizing models and accelerating calculations

(Bhatia et al., 2024). Being one of the least trivial stages in CFD, meshing will inevitably follow this trend, with AI-driven adaptive meshing reducing manual iterations and improving efficiency in complex geometries (Lorsung & Farimani, 2022). However, while these advancements promise significant progress, the true challenge lies in integrating AI effectively without compromising the precision and expertise that human intervention provides. The future of CFD will not be about replacing traditional methods, but about refining and enhancing them through intelligent automation.

AUTHOR CONTRIBUTIONS

Matheo López-Pachón: Conceptualisation, methodology, writing—original draft, data analysis. Jordi Marcé-Nogué: Supervision, funding acquisition, project administration, writing, review and editing.

ACKNOWLEDGEMENTS

The authors thank the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 945413. Thanks to Shin-ichi Fujiwara, Ryoko Matsumoto, and Takumi Watanabe for their invaluable assistance in scanning the *Andrias japonicus* specimen, which was kindly provided by the Kanagawa Prefectural Museum of Natural History. J.M.-N. thanks the Universitat Rovira i Virgili (2023PFR-URV-1239) and the Agency for Management of University and Research Grants (SGR2021-01239). We are grateful to Alessio Veneziano for his assistance in improving the clarity and readability of our manuscript, Steban Gómez Cañón for his 3D modelling contributions, and Iván Darío Gómez for his technical support in OpenFOAM.

FUNDING INFORMATION

European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 945413.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

PEER REVIEW

The peer review history for this article is available at <https://www.webofscience.com/api/gateway/wos/peer-review/10.1111/2041-210X.70146>.

DATA AVAILABILITY STATEMENT

This article does not report any data or code; therefore, no data or code are available.

ORCID

Matheo López-Pachón  <https://orcid.org/0000-0002-9053-3876>
 Jordi Marcé-Nogué  <https://orcid.org/0000-0001-9852-7027>

REFERENCES

- Abu-Zidan, Y., Mendis, P., & Gunawardena, T. (2021). Optimising the computational domain size in CFD simulations of tall buildings. *Heliyon*, 7(4), e06723. <https://doi.org/10.1016/j.heliyon.2021.e06723>

- Anderson, P., Bright, J., Gill, P., Palmer, C., & Rayfield, E. (2011). Models in palaeontological functional analysis. *Biology Letters*, 8, 119–122. <https://doi.org/10.1098/rsbl.2011.0674>
- Ansys Fuent. (2024). *Fluent theory guide*. Ansys Help. https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v252/en/flu_tg/flu_tg.html
- Ansys Fuent. (2025). *Fluent theory guide*. Ansys Help. https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v252/en/flu_tg/flu_tg.html
- Arlinger, B. (1975). Calculation of transonic flow around axisymmetric inlets. *AIAA Journal*, 13, 1614–1621. <https://doi.org/10.2514/3.60584>
- Bahary, M. (1994). Experimental and computational studies of hydrodynamics in two and three phase fluidized beds. <https://doi.org/10.2172/10123398>
- Baker, T. (1986). Mesh generation by a sequence of transformations. *Applied Numerical Mathematics*, 2, 515–528. [https://doi.org/10.1016/0168-9274\(86\)90005-X](https://doi.org/10.1016/0168-9274(86)90005-X)
- Baker, T. J. (2005). Mesh generation: Art or science? *Progress in Aerospace Sciences*, 41(1), 29–63. <https://doi.org/10.1016/j.paerosci.2005.02.002>
- Bhatia, D., Loukas, J., Cabrera, A., & Lyras, K. (2024). A deep learning computational fluid dynamics solver for simulating liquid hydrogen jets. *Physics of Fluids*, 36(5), 057120. <https://doi.org/10.1063/5.0206562>
- Blocken, B. (2015). Computational fluid dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91, 219–245. <https://doi.org/10.1016/j.buildenv.2015.02.015>
- Blocken, B., Stathopoulos, T., & Carmeliet, J. (2007). CFD simulation of the atmospheric boundary layer: Wall function problems. *Atmospheric Environment*, 41, 238–252. <https://doi.org/10.1016/j.atmosenv.2006.08.019>
- Blocken, B., Stathopoulos, T., Carmeliet, J., & Hensen, J. L. M. (2011). Application of computational fluid dynamics in building performance simulation for the outdoor environment: An overview. *Journal of Building Performance Simulation*, 4(2), 157–184. <https://doi.org/10.1080/19401493.2010.513740>
- Bourke, J., Porter, W., Ridgely, R., Lyson, T., Schachner, E., Bell, P., & Witmer, L. (2014). Breathing life into dinosaurs: Tackling challenges of soft-tissue restoration and nasal airflow in extinct species. *Anatomical Record*, 297, 2148–2186. <https://doi.org/10.1002/ar.23046>
- Brown, C., & Kirk, A. (2023). Characterizing lift, drag, and pressure differences across wandering salamanders (*Aneides vagrans*) with computational fluid dynamics to investigate aerodynamics. *Journal of Morphology*, 284, e21583. <https://doi.org/10.1002/jmor.21583>
- Cadence. (2016). I'm Patrick Baker and this is how I mesh. https://community.cadence.com/cadence_blogs_8/b/cfd
- Cedrón, R. (2018). CFD applied to BIFASIC flows and numerical models for the analysis and design of pipe networks.
- CFD-Online. (2018). CFD-Online. <https://www.cfd-online.com>
- Eiseman, P. (1979). A multi surface method of coordinate generation. *Journal of Computational Physics*, 33, 118–150. [https://doi.org/10.1016/0021-9991\(79\)90031-7](https://doi.org/10.1016/0021-9991(79)90031-7)
- Erhunmwun, I., & Ikponmwosa, U. (2017). Review on finite element method. *Journal of Applied Sciences and Environmental Management*, 21, 999. <https://doi.org/10.4314/jasem.v21i5.30>
- Esteve, J., & López-Pachón, M. (2023). Swimming and feeding in the Ordovician trilobite Microparia speciosa shed light on the early history of nektonic life habits. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 625, 111691. <https://doi.org/10.1016/j.palaeo.2023.111691>
- Esteve, J., Lopez-Pachon, M., Ramirez, C. G., & Gomez, I. (2021). Fluid dynamic simulation suggests hopping locomotion in the Ordovician trilobite Placoparia. *Journal of Theoretical Biology*, 531, 110916. <https://doi.org/10.1016/j.jtbi.2021.110916>
- Fortuny, J., Marcé-Nogué, J., Heiss, E., Sanchez, M., Gil, L., & Galobart, À. (2015). 3D bite modeling and feeding mechanics of the largest living amphibian, the Chinese Giant salamander *Andrias davidianus* (Amphibia: Urodea). *PLoS One*, 10(4), e0121885. <https://doi.org/10.1371/journal.pone.0121885>
- Ganovelli, F., Corsini, M., Pattanaik, S., & Di Benedetto, M. (2014). *Introduction to computer graphics: A practical learning approach*. <https://doi.org/10.1201/b15978>
- Gerck, E. (2022). On the existence of solutions to the Navier-Stokes equations.
- Gersten, K. (2009). *Hermann Schlichting and the boundary-layer theory*. (Vol. 102, pp. 3–17) https://doi.org/10.1007/978-3-540-95998-4_2
- Gibson, B. M., Furbish, D. J., Rahman, I. A., Schmeeckle, M. W., Laflamme, M., & Darroch, S. A. F. (2021). Ancient life and moving fluids. *Biological Reviews*, 96(1), 129–152. <https://doi.org/10.1111/brv.12649>
- Grohganz, M., Ferrón, H., Johanson, Z., & Donoghue, P. (2023). Testing hypotheses of pteraspid heterostracan feeding using computational fluid dynamics. *Journal of Vertebrate Paleontology*, 43, e2272974. <https://doi.org/10.1080/02724634.2023.2272974>
- Gutarría Diaz, S., Moon, B., Rahman, I., Palmer, C., Lautenschlager, S., Brimacombe, A., & Benton, M. (2019). Effects of body plan evolution on the hydrodynamic drag and energy requirements of swimming ichthyosaurs. *Proceedings of the Royal Society B: Biological Sciences*, 286, 20182786. <https://doi.org/10.1098/rspb.2018.2786>
- Haug, E. (1989). *Computer-aided kinematics and dynamics of mechanical systems volume-I*.
- He, J., Feng, C., Kuang, L., Han, L., Jia, W., Bai, H., & Jiang, J. (2024). A numerical study on the hydrodynamics of a swimming crocodile model. *Physics of Fluids*, 36, 035108. <https://doi.org/10.1063/5.0191371>
- Hebdon, N., Ritterbush, K. A., & Choi, Y. (2020). Computational fluid dynamics modeling of fossil ammonoid shells. *Palaeontologia Electronica*, 23(1), 1–20. <https://doi.org/10.26879/956>
- Heiss, E., Natchev, N., Gumpenberger, M., Weissenbacher, A., & Van Wassenbergh, S. (2013). Biomechanics and hydrodynamics of prey capture in the Chinese giant salamander reveal a high-performance jaw-powered suction feeding mechanism. *Journal of the Royal Society Interface*, 10(82), 20121028. <https://doi.org/10.1098/rsif.2012.1028>
- Hernandez Perez, V., Abdulkadir, M., & Azzopardi, B. (2011). Grid generation issues in the CFD modelling of two phase flow in a pipe. *The Journal of Computational Multiphase Flows*, 3(1), 13–26. <https://doi.org/10.1260/1757-482X.3.1.13>
- Huston, R. L. (1991). Multibody dynamics—Modeling and analysis methods. *Applied Mechanics Reviews*, 44(3), 109–117.
- Khatib-Shahidi, B., & Smith, R. (2010). Analysis comparison between CFD and FEA of an idealized concept V-hull floor configuration in two dimensions. 14.
- Kong, F., Kheyfets, V., Finol, E., & Cai, X. (2019). Simulation of unsteady blood flows in a patient-specific compliant pulmonary artery with a highly parallel monolithically coupled fluid-structure interaction algorithm. *International Journal for Numerical Methods in Biomedical Engineering*, 35, e3208. <https://doi.org/10.1002/cnm.3208>
- Lee, K., Huang, M., Yu, N., & Rubbert, P. (1980). Grid generation for general three-dimensional configurations.
- Lei, M., Willis, M., Schmidt, B., & Li, C. (2023). Numerical investigation of odor-guided navigation in flying insects: Impact of turbulence, wingbeat-induced flow, and Schmidt number on odor plume structures. *Biomimetics*, 8, 593. <https://doi.org/10.3390/biomimetics8080593>
- Li, Y., Castro, A. M., Sinokrot, T., Prescott, W., & Carrica, P. (2015). Coupled multi-body dynamics and CFD for wind turbine simulation including explicit wind turbulence. *Renewable Energy*, 76, 338–361. <https://doi.org/10.1016/j.renene.2014.11.014>
- Liu, F. (2017). A thorough description of how wall functions are implemented in OpenFOAM.
- Liu, H., Wassersug, R., & Kawachi, K. (1996). Computational fluid dynamic study of tadpole swimming. *The Journal of Experimental Biology*, 199, 1245–1260. <https://doi.org/10.1242/jeb.199.6.1245>
- Liu, Y., Xiao, Q., & Inceciik, A. (2017). A coupled CFD/multibody dynamics analysis tool for offshore wind turbines with aeroelastic blades. V010T09A038 <https://doi.org/10.1115/OMAE2017-61062>

- Lorsung, C., & Farimani, A. B. (2022). MeshDQN: A deep reinforcement learning framework for improving meshes in computational fluid dynamics (No. arXiv:2212.01428). *arXiv*. <https://doi.org/10.48550/arXiv.2212.01428>
- ManchesterCFD Team. (2024). ManchesterCFD research and consultancy. Retrieved September 14, 2024, from <https://www.manchestercfd.co.uk/>
- Marcé Nogué, J., Fortuny, J., Gil, L., & Sánchez, M. (2015). Improving mesh generation in finite element analysis for functional morphology approaches. *Spanish Journal of Palaeontology*, 30, 117–132. <https://doi.org/10.7203/sjp.30.1.17227>
- Marcon, J., Turner, M., Peiró, J., Moxey, D., Pollard, C., Bucklow, H., & Gammon, M. (2019). High-order curvilinear hybrid mesh generation for CFD simulations. <https://doi.org/10.48550/arXiv.1901.00992>
- OpenFOAM. (n.d.). OpenFOAM. Retrieved August 20, 2025, from <https://openfoam.org/>
- Openfoam. (2024). User guide of OpenFoam v11. <https://www.openfoam.com/documentation/user-guide>
- Patel, V., Rodi, W., & Scheuerer, G. (1985). Turbulence models for Near-Wall and low Reynolds number flows: A review. *AIAA Journal*, 23, 1308–1319. <https://doi.org/10.2514/3.9086>
- Pérezagua López, R. L. (2023). Estado del problema del milenio: Las ecuaciones de Navier-Stokes. *Revista Nuclear, España*. <https://www.revistanuclear.es/divulgacion/estado-del-problema-del-milenio-las-ecuaciones-de-navier-stokes/>
- Piegł, L., & Tiller, W. (1997). The NURBS book (2nd ed.).
- Powalla, D., Hoerner, S., Cleynen, O., & Thévenin, D. (2022). A numerical approach for active fish behaviour modelling with a view toward hydropower plant assessment. *Renewable Energy*, 188, 957–966. <https://doi.org/10.1016/j.renene.2022.02.064>
- Prandtl, L. (1904). Über Flüssigkeitsbewegung bei sehr kleiner Reibung. *Verhandl. III. Intern. Math. Kongr. Heidelberg*, 2, 484.
- Rahman, I., Davies, T., Lautenschlager, S., Cunningham, J., Rayfield, E., & Donoghue, P. (2017). Setting standards for the publication of virtual fossils. <https://doi.org/10.1130/abs/2017AM-298962>
- Rahman, I., O’Shea, J., Lautenschlager, S., & Zamora, S. (2020). Potential evolutionary trade-off between feeding and stability in Cambrian cinctan echinoderms. *Palaeontology*, 63, 689–701. <https://doi.org/10.1111/pala.12495>
- Rahman, I. A. (2017). Computational fluid dynamics as a tool for testing functional and ecological hypotheses in fossil taxa. *Palaeontology*, 60(4), 451–459. <https://doi.org/10.1111/pala.12295>
- Ramponi, R., & Blocken, B. (2012). CFD simulation of cross-ventilation for a generic isolated building: Impact of computational parameters. *Building and Environment*, 53, 34–48. <https://doi.org/10.1016/j.buildenv.2012.01.004>
- Rao, S. S. (2005). *Finite element method in engineering*. Elsevier.
- Revuz, J., Hargreaves, D., & Owen, J. (2012). On the domain size for the steady-state CFD modelling of a tall building. *Wind and Structures*, 15, 313–329. <https://doi.org/10.12989/was.2012.15.4.313>
- Roache, P. J. (1998). *Fundamentals of computational fluid dynamics*. Hermosa Publishers.
- Rogers, D. F. (2001). *An introduction to NURBS: With historical perspective*. Morgan Kaufmann Publishers.
- Salim, S. M., Ariff, M., & Cheah, S. C. (2010). Wall $y+$ approach for dealing with turbulent flows over a wall mounted cube. *Progress in Computational Fluid Dynamics, an International Journal*, 10, 341–351. <https://doi.org/10.1504/PCFD.2010.035368>
- Salim, S. M., & Cheah, S. C. (2009). Wall $y+$ strategy for dealing with wall-bounded turbulent flows. *International MultiConference of Engineers and Computer Scientists, IMECS 2009*, 2, 1–6.
- Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2), 187–260. <https://doi.org/10.1145/356924.356930>
- Satyam, S., Fabregat Sanjuan, A., & Huera Huarte, F. J. (2024). Two dimensional simulation of the flow around a three straight bladed vertical axis wind turbine for different pitch angles. In *OMAE2024, Volume 6 Polar and Arctic Sciences and Technology, CFD, FSI, and AI*. American Society of Mechanical Engineers (ASME). <https://doi.org/10.1115/OMAE2024-125803>
- Schlüzen, H., Gräwe, D., Bohnenstengel, S., Schlüter, I., & Koppmann, R. (2011). Joint modelling of obstacle induced and mesoscale changes-current limits and challenges. *Journal of Wind Engineering and Industrial Aerodynamics*, 99, 217–225. <https://doi.org/10.1016/j.jweia.2011.01.009>
- Shabana, A. A. (2020). *Dynamics of multibody systems*. Cambridge University Press.
- Shahril, K., Shamsuddin, T. K. A., Nizam, S., Najib, M., & Azid, I. (2024). *Methods and applications in fluid structure interaction (FSI)* (pp. 115–123). https://doi.org/10.1007/978-3-031-51859-1_13
- Shewchuk, J. R. (1996). Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied computational geometry* (pp. 203–222). Springer.
- SimScale. (2018). How to check convergence of a CFD simulation? <https://www.simscale.com/knowledge-base/how-to-check-convergence-of-a-cfd-simulation/>
- Skews, B. (2016). Hydrodynamics of an extinct amphibian. *Journal of Applied Fluid Mechanics*, 9, 2899–2903. <https://doi.org/10.29252/jafm.09.06.25634>
- Slater, J. W. (2021). Spatial convergence tutorial. <https://www.grc.nasa.gov/WWW/wind/valid/tutorial/spatconv.html>
- Theodorsen, T., & Garrick, I. E. (1979). General potential theory of arbitrary wing sections. In *Classical aerodynamics theory* (pp. 257–290). NASA Ames Research Center. <https://ntrs.nasa.gov/citations/19800006774>
- Thompson, J. F., Soni, B. K., & Weatherill, N. P. (1999). *Handbook of grid generation*. CRC Press LLC.
- Tominaga, Y., Mochida, A., Murakami, S., & Sawaki, S. (2008). Comparison of various revised $k-\epsilon$ models and LES applied to flow around a high-rise building model with 1:1:2 shape placed within the surface boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(4), 389–411. <https://doi.org/10.1016/j.jweia.2008.01.004>
- Troelsen, P., Wilkinson, D., Seddighi, M., Allanson, D., & Falkingham, P. (2019). Functional morphology and hydrodynamics of plesiosaur necks: Does size matter? *Journal of Vertebrate Paleontology*, 39, e1594850. <https://doi.org/10.1080/02724634.2019.1594850>
- Tucker, V. (1998). Gliding flight: Speed and acceleration of ideal falcons during diving and pull out. *The Journal of Experimental Biology*, 201, 403–414. <https://doi.org/10.1242/jeb.201.3.403>
- Unnikrishnan, S., & Somanath, S. (2024). *Introduction to finite element analysis*. Springer. <https://doi.org/10.1007/s10546-011-9667-4>
- Vascular, S. (2009). Meshing guide. <https://simvascular.github.io/documentation/meshing.html>
- Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: The finite volume method* (2nd ed.). Pearson Education Ltd.
- Wang, Z., Edgecombe, G., & Hou, J.-B. (2024). Function of flow wakes for queuing trilobites: Positioning rather than drag reduction—Criteria for drag force assessment in palaeontological CFD simulations. *Palaeogeography Palaeoclimatology Palaeoecology*, 646, 112239. <https://doi.org/10.1016/j.palaeo.2024.112239>
- White, F. M. (2009). *Fluid mechanics* (6th ed.). McGraw-Hill.
- Yang, X. I. A., Chen, P. E. S., Zhang, W., & Kunz, R. (2024). Predictive near wall modelling for turbulent boundary layers with arbitrary pressure gradients. *Journal of Fluid Mechanics*, 993, A1. <https://doi.org/10.1017/jfm.2024.565>
- Young, D., & Hildebrand, F. (1957). Introduction to numerical analysis. *The American Mathematical Monthly*, 64, 128. <https://doi.org/10.2307/2310412>

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

Figure S1. (a) Control nodes in NURBS curves in the 2D model, (b) control nodes in NURBS surfaces in the 3D model, (c) parts included in the CFD mesh for a simple 2D model, and (d) parts included in the CFD mesh for the 3D model, adapted from (ManchesterCFD, 2024). Different types of meshes: (e) type of elements, (f) real specimen, (g) CAD model, (h) FEA model, and (i) CFD model.

Figure S2. Different mapping meshes in the salamander *Andrias japonicus*. (a) O-mesh: mapping into the unit circle (Theodorsen & Garrick, 1979), (b) C-mesh: Coordinate mapping for the tail of the salamander (Arlinger, 1975), (c) H-mesh: section through the frontal symmetry of the salamander, (d) Overset method, (e) Patched method, (f) Composite method, and (g) a multiblock C-H mesh (Baker, 1986, 2005), (Eiseman, 1979).

Figure S3. *Andrias japonicus* CAD position: (a) Relaxed pose, (b) Swimming pose, (c) Stalking or breathing position, (d) Real *Andrias japonicus*, and (e) CAD *Andrias japonicus*.

Figure S4. (a) Water-blocking effect, (b) local venturi effect, and (c) global venturi effect, taken from (Abu-Zidan et al., 2021).

Figure S5. Domain Shape-Related Errors in *Andrias japonicus*. (a) Water-blocking effect around the salamander's body. (b) Velocity error (%) in the computational mesh. Minimal Presence of Global and Local Venturi Effects (GVE & LVE).

Figure S6. Illustration of characteristic length (L) for various objects in fluid dynamics. (a) Vertical structures such as buildings, where height is the predominant dimension. (b) Large terrestrial vertebrates like sauropod dinosaurs, whose characteristic length depends on posture. (c) Elongated marine organisms such as oarfish, where height dominates when swimming vertically. (d) Horizontally oriented

marine animals like whales, where the characteristic length is typically their total body length (Vogel, 1994). (e) Quadrupeds like elephants, where characteristic length varies with movement and posture. (f) Streamlined man-made objects such as cars, where length dictates aerodynamic behavior. (g) Humans, whose characteristic length depends on movement orientation. (h) Salamander *Andrias japonicus*.

Figure S7. Mesh adaptation based on flow direction in Trilobite Microparia, taken from (Esteve & López-Pachón, 2023). (a) Computational domain for crossflow and oblique directions (e.g. 45°). (b) Symmetric domain, where only half of the trilobite is meshed, applied for frontal and rear flow simulations.

Figure S8. Example of domain dimensions in the amphibian *Andrias japonicus*. (a) Dimensions surrounding the geometry of the animal in a resting position on the ground, (b) dimensions surrounding the geometry of the amphibian in a swimming position, maintaining a distance from the ground, (c) blockage ratio, (d) velocity profiles upstream of the animal when positioned on the ground, adapted from (Blocken et al., 2007), and (e) refinement boxes around the body.

How to cite this article: López-Pachón, M., & Marcé-Nogué, J. (2025). The crucial role of meshing in computational fluid dynamics simulations for organic geometries in paleobiology: Describing fluid dynamics performance through best practices. *Methods in Ecology and Evolution*, 16, 2170–2194.

<https://doi.org/10.1111/2041-210X.70146>