Title:
A Robust and Automated Method for Checking and Repairing Surface Mesh Models to Ensure Watertightness for CFD Simulations

Authors:
Tianyu Zhou, tzhou4@ualberta.ca, University of Alberta
Carlos F. Lange, carlos.lange@ualberta.ca, University of Alberta
Michael Versteege, mhverste@ualberta.ca, University of Alberta
Yongsheng Ma, mays@sustech.edu.cn, Southern University of Science and Technology
Brian Fleck, bfleck@ualberta.ca, University of Alberta

Keywords:
3D Watertight Surface Mesh Reconstruction, Robust and Automated Mesh Repair, Geometry Preprocessor for CFD Simulations, Model Watertightness Validation, Interactive Surface Mesh Model Processor

DOI: 10.14733/cadconfP.2025.160-165

Introduction:
Simulating fluid flow, especially airflow around target objects, has a wide range of potential application areas across various fields, such as aerospace engineering, the automotive industry, and wind energy systems. Ensuring the watertightness of a model is a critical step before conducting Computational Fluid Dynamics (CFD) simulations for studying fluid flow around an object [1]. A watertight model guarantees that the simulation domain is fully enclosed, preventing errors in mesh generation and in solving equations. In summary, validating the domain watertightness before a CFD simulation is necessary prior to attempting to solve for a simulated velocity field.

Over the past few decades, watertight validation of digital geometric domain has been widely applied across diverse fields, including medical imaging [2], computer-aided design and manufacturing (CAD/CAM) [3], and building design with airflow simulations [1,4]. Significant advancements have been achieved in watertight validation, model repair, and surface reconstruction. These efforts primarily focus on enhancing the efficiency of model processing and improving the accuracy of repaired models, particularly in areas such as deviation control and sharp feature preservation [2–4]. Despite these developments, several challenges remain: varying application goals, diverse model sources and formats, and the need for domain-specific expertise often hinder the seamless application of watertight validation techniques. Moreover, existing methods frequently require extensive manual intervention, which can be time-consuming and prone to errors.

Recognizing these challenges, this paper introduces a robust and automated approach to watertight validation and repair. Our proposed method is designed to address key limitations in existing workflows by eliminating the need for advanced geometry processing expertise and minimizing manual effort. It supports a variety of input and output formats, making it versatile for different applications and data sources. The method is also equipped with user-friendly features such as step-by-step guidance, real-time error notifications, and comprehensive visualization tools. These capabilities ensure a streamlined and efficient user experience, making our approach a valuable advancement in the field of geometry preprocessing for CFD simulations and beyond.

Main Idea:
The proposed robust and automated method for model repair and watertight validation is designed to function as a geometry preprocessor for CFD simulations. Its primary objective is to ensure that the

output models are watertight, enabling them to serve as usable calculation domains in CFD simulation environments. The motivation behind this development is to create a generic and practical tool that can be seamlessly integrated into CFD workflows, specifically for simulating flow around target objects.

The overall framework for simulating flow around a target object, shown in

Fig. *1*, involves six key steps. The process begins with the client handing over the object of interest, requirements, and estimated budget to service provider (users) for flow simulation. Models of the target object and important upstream surroundings are acquired from sources such as digital libraries, CAD software, 3D scanning in formats like CAD, STL, or point clouds. These models are then processed using the proposed geometry preprocessor, which ensures watertightness, converts file formats for CFD compatibility, and performs necessary repairs and simplifications with minimal manual intervention. The details of the geometry preprocessor, shown in

Fig. *2*, will be explained further in the next few paragraphs. Once the geometry is preprocessed, an adaptive gridding process will generate a computational mesh tailored to critical regions, balancing accuracy and computational cost. The prepared geometry and grid are then integrated into a CFD platform, such as OpenFOAM, where boundary conditions and turbulence models are applied to simulate flow around the target object. The simulation results are analyzed to extract key parameters, such as pressure and velocity fields, and visualized properly. Finally, a structured report summarizes the results and key insights, delivering actionable findings or solutions to the client.

The proposed geometry preprocessor is designed to ensure the watertightness and compatibility of input models for CFD simulations, as illustrated in

Fig. *2*. The process begins by importing the model file using an integrated file reader, capable of handling various formats such as CAD, STL, or point clouds. For point clouds, where surface connectivity is not explicitly defined, we first reconstruct a surface mesh using Delaunay triangulation before proceeding with further processing. Once loaded, the surface information of the model is extracted, including sampling vertices and triangle faces, to generate a surface mesh. Mesh defects, such as duplicate vertices, duplicate faces, unreferenced vertices, and degenerated faces, are identified and removed. Subsequently, derived surface mesh information, including normal vectors for each triangle face and principal curvatures for each vertex, is calculated to support downstream processing. The surface mesh is then simplified to optimize its complexity while preserving critical geometric features. A watertight detector evaluates the watertightness of the surface mesh. If the mesh is found to be non-watertight, the model repair tool is activated to iteratively fix the mesh. This repair process includes removing intersected or overlapped triangle faces, detecting boundaries and holes, and filling mesh holes to restore watertightness. The process continues until the watertight condition is satisfied. Once validated, the final surface mesh is written into an output model file, ensuring compatibility with the CFD environment.

The visualization of results is performed before the final model is generated, allowing users to verify the quality of the processed model. This visualization highlights differences between the input and output models, such as the location of detected holes and how they have been repaired. Finally, the output model files are exported and saved.

With the integration of the file reader/writer, watertight detector, model repair tool, visualization tool, and other geometry processing algorithms makes this preprocessor robust and efficient, requiring minimal manual intervention. In the next subsections, key components of this proposed method will be introduced in detail.

Integrated file reader/writer

An integrated model file reader/writer has been developed and introduced as part of the geometry preprocessor. The file reader imports model files and extracts essential geometric data, including vertices and faces, which are crucial for generating a surface mesh (see Fig. 3, Tab. 1, and Tab. 2). Conversely, the file writer saves the repaired surface mesh as an output model file, ensuring compatibility with a wide range of applications and downstream workflows, particularly for CFD simulations.

Proceedings of CAD’25, Shenzhen, China, June 23-25, 2025, 160-165

The integrated file reader/writer supports a variety of standard model formats to ensure interoperability across different sources and applications. The file reader currently supports STL, PLY, FBX, OBJ, GLTF/GLB, OFF, DAE, STEP/STP, and IGES/IGS formats. Similarly, the file writer supports most of these formats, with the exception of STEP/STP and IGES/IGS. This limitation arises because, during the process of extracting surface information from solid CAD models in STEP/STP and IGES/IGS formats, the parametric definition of the original model is lost. As a result, the processed surface mesh cannot be written back into a parametric defined solid CAD file.

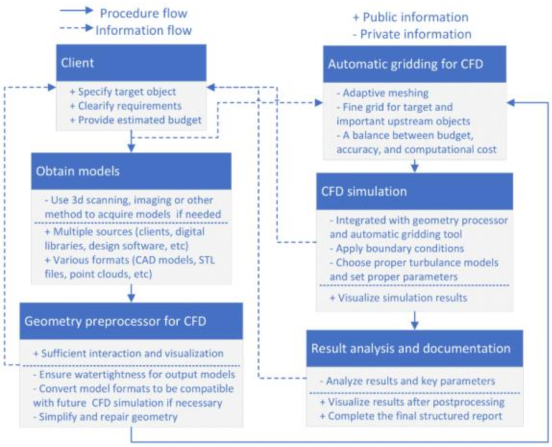


Fig. 1: The overall workflow of simulating flow around a target object.

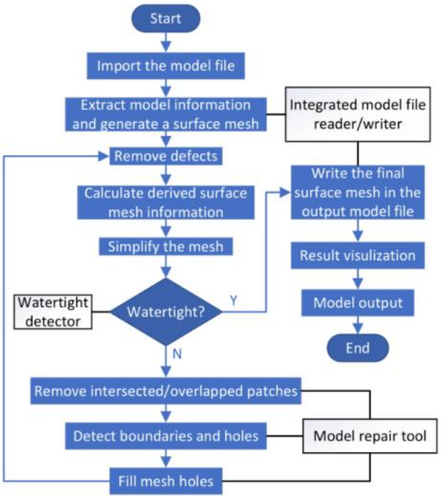


Fig. 2: Detailed workflow of the proposed geometry preprocessor for CFD simulation.

Watertight Detector

The Watertight Detector is a critical component of the proposed geometry preprocessor, designed to assess the watertightness of the extracted surface mesh. A mesh is considered watertight if it satisfies three key conditions: (1) no non-manifold edges, meaning every edge is shared by exactly two faces; (2)

no non-manifold vertices, ensuring each vertex participates in a well-defined surface topology; and (3) no self-intersections, meaning the mesh does not intersect itself anywhere [5].

The detector utilizes a topological analysis algorithm to evaluate edge-manifold and vertex-manifold properties by verifying that each edge is shared by exactly two faces and each vertex connects to a well-defined set of faces. Self-intersections are detected through geometric queries, ensuring no overlapping triangles exist. If all three conditions are satisfied, the Watertight Detector confirms that the surface mesh is watertight. Otherwise, a notification is generated to inform the user that the model requires repair. This automated validation process provides detailed feedback to the user, enhancing interaction by identifying specific issues and guiding necessary repairs.

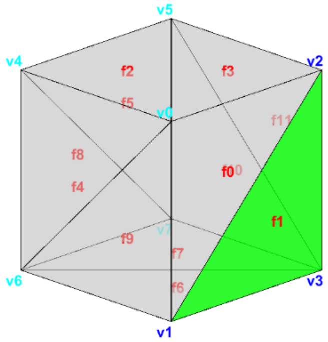


Fig. 3: The surface mesh of an object (cube) with a highlighted triangle face and its vertices.

Vertex	*X*	*Y*	*Z*
v0	0	1	1
v1	0	1	0
v2	0	0	1
v3	0	0	0

Tab. 1: The sample data set of vertices.

Face	*Vertex1*	*Vertex2*	*Vertex3*
f0	v0	v1	v2
f1	v2	v1	v3
f3	v4	v0	v5
f4	v5	v0	v2

Tab. 2: The sample data set of faces.

Model Repair Tool

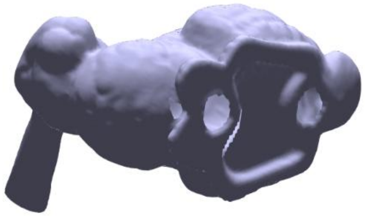
The Model Repair Tool is an essential component of the proposed geometry preprocessor, designed to address and resolve issues in non-watertight meshes. It combines three primary operations—removing intersected/overlapped patches, detecting boundaries and holes, and filling mesh holes—to repair the surface mesh and ensure a fully watertight output mesh. Our model repair tool employs curvature-aware interpolation and constrained triangulation techniques to fill holes while preserving original geometry as much as possible. Sharp features are maintained using anisotropic mesh adaptation to prevent unintended shape distortion.

By integrating these three operations, the Model Repair Tool effectively resolves mesh defects, ensuring the final output is watertight and ready for use in CFD simulation environments. This automated approach significantly reduces the need for manual intervention while providing comprehensive visualization throughout the repair process, enhancing efficiency, automation, and transparency in the geometry processing workflow.

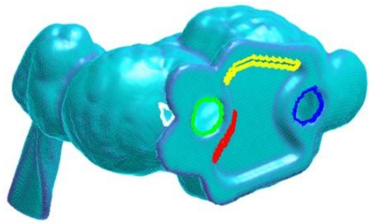
Case Study:

In this section, the proposed method is validated through the watertight check and model repair of a widely recognized and used test model, the Stanford Bunny [6]. An application plug-in has been developed in MATLAB, integrating several custom-designed algorithms by the authors, built-in functions from the MATLAB Lidar Toolbox, and additional tools from a mesh processing toolbox [7]. For this case study, the OBJ file of the Stanford Bunny is used as the input model. The input is non-watertight and contains several holes and other defects. After applying the proposed method, holes are filled, defects are removed, and a watertight STL file is successfully generated as the output model. The watertightness of the output model is verified through a comprehensive validation process, including edge-manifold check, vertex-manifold check, self-intersection test, isolated vertex detection, etc

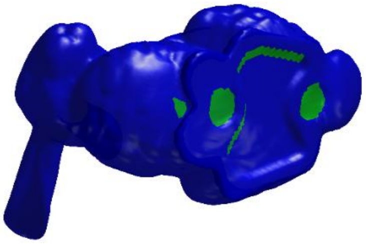
Fig. 4 illustrates the model repair and watertight validation processes of the Stanford Bunny using the developed MATLAB application based on the proposed method. Fig. 4 (a) shows the original model in OBJ format, which is non-watertight and has numerous defects, such as holes at the bottom. Fig. 4 (b) highlights the extracted surface mesh after watertight checking and marking detected holes and other defects using the watertight detector. Fig. 4 (c) displays the repaired surface mesh, where holes have been filled and defects have been removed, resulting in a watertight model. Finally, Fig. 4 (d) presents the output model in STL format, which is validated for watertightness and ready for CFD simulation. This case study demonstrates how the proposed method can turn a defective non-watertight 3D model into a watertight, simulation-ready geometry, ensuring reliability for downstream applications.



(a)



(b)



(c)

(d)

Fig. 4: Geometry preprocessing of Stanford Bunny for CFD simulation: (a) Original model (.obj), (b) Extracted surface mesh with holes detected and marked, (c) Processed surface mesh with holes filled and defects removed, (d) Final output model (.stl) for future CFD simulation with watertight validated.

Conclusions:

This study presented a robust and automated approach for watertight validation and repair of 3D models, addressing the critical challenges associated with geometry preprocessing for applications such as simulating flow around a target object with CFD. The proposed method integrates an advanced file reader/writer, a watertight detector, and a comprehensive model repair tool, ensuring compatibility with various file formats and minimizing manual intervention. Through key operations, such as removing intersected or overlapped patches, detecting boundaries and holes, and filling mesh holes, the method efficiently transforms defective, non-watertight models into watertight, simulation-ready geometries, providing a robust and practical solution for geometry preprocessing for subsequent CFD simulations. The proposed approach was validated using the Stanford Bunny, demonstrating its ability to detect and remove defects, fill holes, and validate watertightness. In our case study, the key features of the proposed method have been validated: robust, supporting various file formats; highly automated, requiring minimal user expertise; and user-friendly, with step-by-step guidance, notifications, and visualization tools. These features ensure efficient and reliable transformation of non-watertight models into watertight models ready for future CFD simulation.

In the future, the authors would like to explore the development of alternative platforms to implement similar functionalities in a completely open-source environment, such as Python, enhancing accessibility and adaptability for all users. Efforts will also focus on expanding the extracted surface model from triangle meshes to polygon meshes, enabling greater versatility in handling complex geometries. Additionally, the method will be tested on a wider range of cases, particularly practical engineering models, to validate its effectiveness and robustness in real-world applications.

Acknowledgements:

The authors gratefully acknowledge Flexible Machines Corporation (https://www.flexiblemachines.ca/) for offering insights into potential development frameworks for future HVAC and wind energy systems in urban buildings. This served as the primary motivation for this research, aimed at developing a practical geometry preprocessing tool to ensure the watertightness of surface mesh models for CFD simulations. In addition, the authors thank Gaurav Rasaily (rasaily@ualberta.ca), a research assistant from the University of Alberta, for his contribution in developing CFD simulation tools for downstream testing of the output models. The authors also wish to thank Dr. Nicolas Douillet (nicolas.douillet9@gmail.com) for providing the open-source MATLAB Toolbox for Mesh Processing and for his constructive suggestions in the model processing workflow.

Tianyu Zhou, https://orcid.org/0000-0001-6735-387X
Carlos F. Lange, https://orcid.org/0000-0001-9390-8728
Michael Versteege, https://orcid.org/0000-0001-7534-9694
Yongsheng Ma, https://orcid.org/0000-0002-6155-0167
Brian Fleck, https://orcid.org/0000-0002-2747-3719

References:

[1] Zheng, S.; Zhai, Z.J.; Wang, Y.; Xue, Y.; Duanmu, L.; Liu, W.: Evaluation and comparison of various fast fluid dynamics modeling methods for predicting airflow around buildings, Building Simulation, 15(6), 2022, 1083–1095. https://doi.org/10.1007/s12273-021-0860-1

[2] Zhou, H.; Sun, P.; Ha, S.; Lundine, D.; Xiong, G.: Watertight modeling and segmentation of bifurcated Coronary arteries for blood flow simulation using CT imaging, Computerized Medical Imaging and Graphics, 53, 2016, 43–53. https://doi.org/10.1016/j.compmedimag.2016.06.003

[3] Ye, Y.; Wang, Y.; Cao, J.; Chen, Z.: Watertight surface reconstruction method for CAD models based on optimal transport, Computational Visual Media, , 2024. https://doi.org/10.1007/s41095-023-0355-3

[4] Cai, Y.; Fan, L.: An efficient approach to automatic construction of 3d watertight geometry of buildings using point clouds, Remote Sensing, 13(10), 2021. https://doi.org/10.3390/rs13101947

[5] Charton, J.; Baek, S.; Kim, Y.: Mesh repairing using topology graphs, Journal of Computational Design and Engineering, 8(1), 2021, 251–267. https://doi.org/10.1093/jcde/qwaa076

[6] Stanford University Computer Graphics Laboratory: The Stanford Bunny. Available at: https://graphics.stanford.edu/data/3Dscanrep/. Accessed January 14, 2025.

[7] Nicolas Douillet: Mesh processing toolbox, (https://github.com/NicolasDouillet/mesh_processing_toolbox/releases/tag/v3.3), GitHub. Retrieved January 15, 2025.