

# NIMPHS: Numerous Instruments to Manipulate and Post-process Hydraulic Simulations

Félix Olart<sup>1,2</sup>, Thibault Oudart<sup>1</sup>, Olivier Bertrand<sup>1</sup>, and Mehdi-Pierre Daou<sup>1</sup>

1 ARTELIA, France 2 Université Gustave Eiffel, France

DOI: [10.21105/joss.04868](https://doi.org/10.21105/joss.04868)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

---

Editor: [Hugo Ledoux](#) ↗ 

## Reviewers:

- [@vbassn](#)
- [@liberostelios](#)

Submitted: 22 September 2022

Published: 13 March 2023

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

The task to communicate and show results from fluid dynamics simulations to non-specialist audiences is sometimes problematic. These audiences may not fully understand common graphical outputs generated through scientific visualization tools which are often based on a variety of color palettes. Therefore, a Blender ([Blender Online Community, 2022](#)) add-on, NIMPHS, has been developed. This add-on aims to ease the process of creating interesting 3D renderings of fluid simulations. This tool provides features to import, manipulate and post-process data from such simulations through the Blender rendering pipeline in order to leverage the power of its two renderers: Eevee and Cycles. Some examples are shown in Figure 1 and Figure 2.

The add-on is developed using open source tools from ARTELIA, and using PyVista ([Sullivan & Kaszynski, 2019](#)), a high-level python API for VTK ([Schroeder et al., 2006](#)). It also makes use of another Blender add-on: Stop-Motion-OBJ ([Jensen, 2022](#)) to ease the creation of animations. NIMPHS tries to unify the way data are manipulated among the different file formats it supports. Indeed, data structures often present similarities in their topologies in CFD (Computational Fluid Dynamics), so it is possible to have shared features.

Fast and memory efficient algorithms to manipulate data using Python in Blender are implemented, for that NumPy ([Harris et al., 2020](#)) is used. When it comes to specific rendering tasks, such as volumes, the multiprocessing modules numba ([Lam et al., 2015](#)) and pyopendb ([Museth, 2013](#)) are used to generate volume sequences in a reasonable amount of time.

## Statement of need

The process of modeling fluid flows is often divided into three main steps: pre-processing, simulation, and post-processing. The first one consists in choosing the right equations and the extent of the model in order to represent the physical phenomena which will be of interest. Since these equations are only solved for a finite number of points in space, there is a need to discretize the modeled space into a mesh. Then, the simulation step makes use of fluid dynamics solvers such as OpenFOAM ([Jasak, 2009](#)) and TELEMAR-MASCARET ([Hervouet, 2007](#)) to solve the previously chosen equations. Finally comes the step of post-processing the generated data in order to visualize the evolution of physical phenomena through time and space.

Post-processing data in the field of computational fluid dynamics is an important part of the modeling process. Engineers and researchers need visualizations to understand, validate and fix the results of their simulations in comparison to real world observations and laboratory experiments. There are several tools at our disposal to generate graphical outputs of such models. We can quote: ParaView ([Ahrens et al., 2005](#)), QGIS ([QGIS Development Team, 2022](#)), Fudaa ([Lacombe & Pasteur, 2008](#)), or Tecplot ([MITCHELL, 2000](#)). These powerful software are commonly used by scientists to provide outputs in order to analyze them and then

take diverse decisions or communicate their results.

However, when it comes to non-specialist audiences these graphical outputs can be very difficult to understand. Scientific visualizations are often minimalist and only represent specific data, which often does not offer the best keys to understand the results. Thus, one idea is to generate photo-realistic renderings of these data in order to produce better visuals for these audiences. Such graphical outputs could ease the understanding of physical phenomena which take place in fluid simulations.

Some of the previously quoted tools try to tackle this issue by providing photo-realistic 3D renderers (e.g., Intel OSPRay (Wald et al., 2017) in ParaView). However, it is often difficult to add visually convincing scenes and contexts to significantly better these graphical outputs. Therefore, the tool NIMPMS is designed to be used for such applications. It is an add-on for Blender, the free and open source 3D creation suite.

## Outlook and use cases

The tool NIMPMS was first designed to post-process hydraulic simulations. However CFD tools can tackle a much larger set of problematics. Thus, NIMPMS can also manage non-hydraulic simulations (see Figure 3 for example). Moreover, the add-on is designed to ease the support and implementation of other file formats which have close similarities with foam (OpenFOAM) and Serafin (TELEMAC-MASCARET) file formats (which are already supported).

The examples shown in Figure 1, Figure 2 and Figure 3 are all attempts at photo-realistic renderings of fluid simulations. However, since Blender is a 3D creation suite, even more creative renderings can be achieved. In particular, we can quote the visual scripting tool geometry nodes available since Blender 2.92 to help in this task.

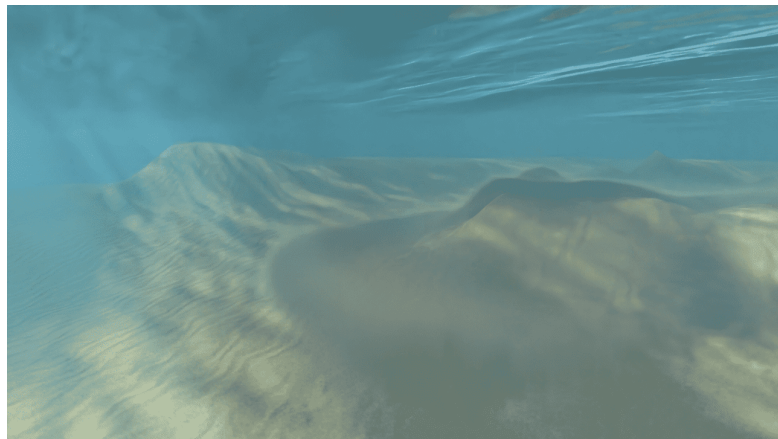
## Acknowledgements

This project has been funded by ARTELIA as part of Félix Olart's internship at ARTELIA.

## Figures



**Figure 1:** The visualized model is the Gironde estuary, France. Simulation made with TELEMAC-2D by ARTELIA and post-processed using NIMPMS in Blender.



**Figure 2:** Model made by ARTELIA, simulated using TELEMAC-3D and post-processed with NIMPHS. This is a visualization of how sediments disperse after a dumping operation.



**Figure 3:** A cylindrical liquid jet glycerol surrounded by silicone oil plunges into a bath of glycerol. Simulation made with OpenFOAM by ARTELIA and LEGI (Laboratoire des Ecoulements Géophysiques et Industriels). Rendered in Blender using NIMPHS.

## References

- Ahrens, J. P., Geveci, B., & Law, C. C. (2005). ParaView: An end-user tool for large-data visualization. *The Visualization Handbook*. <https://doi.org/10.1016/b978-012387582-2/50038-1>
- Blender Online Community. (2022). *Blender - a 3D modelling and rendering package*. Blender Foundation. <http://www.blender.org>
- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hervouet, J. M. (2007). *Hydrodynamics of free surface flows: Modelling with the finite element method*. Wiley. ISBN: 9780470319635

- Jasak, H. (2009). OpenFOAM: Open source CFD in research and industry. *International Journal of Naval Architecture and Ocean Engineering*, 1, 89–94. <https://doi.org/10.3744/JNAOE.2009.1.2.089>
- Jensen, J. (2022). Stop-motion-OBJ. In *GitHub repository*. <https://github.com/neverhood311/Stop-motion-OBJ>; GitHub.
- Lacombe, J.-M., & Pasteur, O. (2008). FUDAA: An open-source framework for the integration of simulation codes, pre-processing, and post-processing tools. *Signal-Image Technologies and Internet-Based System, International IEEE Conference on*, 0, 593–596. <https://doi.org/10.1109/SITIS.2008.39>
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. <https://doi.org/10.1145/2833157.2833162>
- MITCHELL, K. E. (2000). Tecplot 8.0. *Science*, 290(5499), 2097–2097.
- Museth, K. (2013). VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. Graph.*, 32(3). <https://doi.org/10.1145/2487228.2487235>
- QGIS Development Team. (2022). *QGIS geographic information system*. QGIS Association. <https://www.qgis.org>
- Schroeder, W., Martin, K., Lorensen, B., & Kitware, I. (2006). *The visualization toolkit: An object-oriented approach to 3D graphics*. Kitware. ISBN: 9781930934191
- Sullivan, C. B., & Kaszynski, A. A. (2019). PyVista: 3D plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, 4(37), 1450. <https://doi.org/10.21105/joss.01450>
- Wald, I., Johnson, G., Amstutz, J., Brownlee, C., Knoll, A., Jeffers, J., Gunther, J., & Navratil, P. (2017). OSPRay - a CPU ray tracing framework for scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 931–940. <https://doi.org/10.1109/TVCG.2016.2599041>