

INTRODUCTION

High-fidelity **Computational Fluid Dynamics (CFD)** simulations are increasingly executed on **High Performance Computing (HPC)** systems. However, traditional CFD workflows [1] rely heavily on interactive GUI-based tools, limiting scalability and reproducibility.

Goal:

CFD workflows automation for large-scale simulations and integration with modern workflow engines.

Key ideas:

- Automating CFD pipelines using Python scripting
- Controlling simulations programmatically
- Batch execution on HPC clusters
- Enabling parameter studies and campaign experiments at scale

CHALLENGE

Typical Fluent workflows often rely on **GUI-driven preprocessing** in ANSYS SpaceClaim, **manual solver setup**, and **interactive post-processing**. This slows experimentation, prevents reproducibility, and complicates running large parameter sweeps, especially on clusters. Additionally, PyFluent [2] is not yet fully stabilised in terms of API consistency or feature coverage, which poses challenges for **building reliable end-to-end CFD automation**.

Research gap:

- Limited work on fully automated end-to-end Fluent workflows on HPC
- Few solutions integrating CFD pipelines with scalable uncertainty quantification

ENABLING TECHNOLOGIES

The proposed approach builds on modern HPC and Python ecosystem tools that have been already tested on Cyfronet's Ares cluster.

- **PyFluent** – Python interface for ANSYS Fluent
- **SLURM** – HPC workload manager
- **SLURM Array Jobs** – large-scale batch execution
- **Dask Distributed** – scalable Python framework for parallel computing
- **EasyVVUQ** – building UQ/SA workflows

Py/Ansys

slurm
workload manager

dask

EasyVVUQ

SOLUTION TO THE PROBLEM

We designed a **semi-automated CFD workflow** (Figure 1) that enables batch execution of Fluent simulations on HPC using array jobs. The workflow replaces many time-consuming processing steps with a **Python-driven pipeline**.

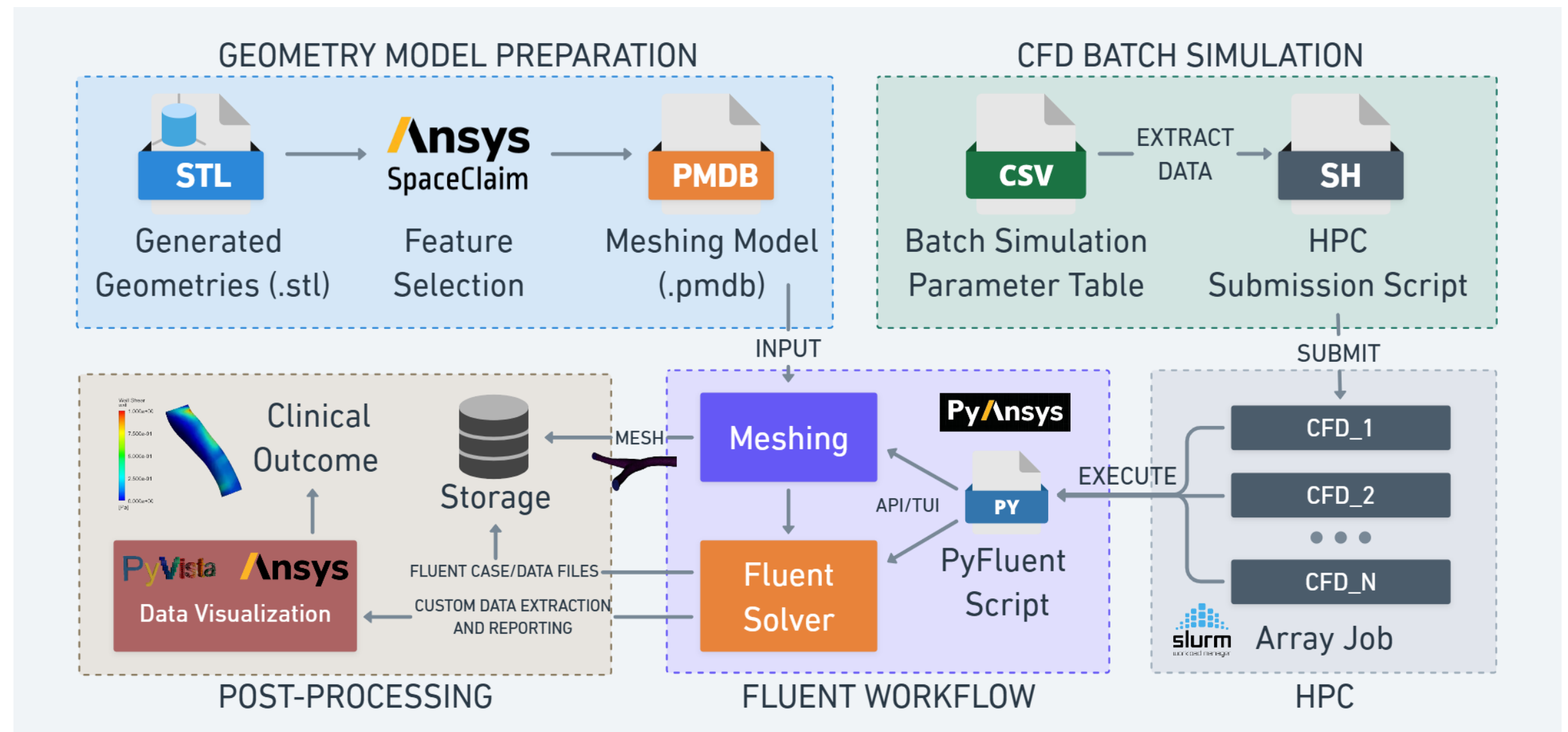


Figure 1: Basic batch execution of CFD workflow with PyFluent on HPC.

Using HPC resources makes it possible to execute **large simulation campaigns**, including systematic perturbations of boundary conditions [3] and analysis of patient-specific geometries [4]. In our studies, this approach enabled **nearly 1000 simulations**, a scale that would be difficult to achieve on local machines.

Key advantages:

- Massive scalability using HPC
- Significantly reduced computation time and manual processing
- Faster experiment preparation and batch execution
- Large parameter sweeps enabled

The workflow can also integrate with tools such as Dask Distributed and EasyVVUQ, allowing **scalable sensitivity analysis [5] and uncertainty quantification studies**.

CONCLUSIONS

Automating CFD workflows using PyFluent on HPC significantly improves the efficiency and scalability of simulation studies. The proposed workflow reduces reliance on GUI-based operations and enables batch execution of simulations on HPC systems, improving reproducibility and enabling large simulation campaigns.

Despite current limitations related to the evolving PyFluent API and partial automation of geometry preparation and result extraction, the approach demonstrates strong potential for building fully automated CFD pipelines for large-scale research.

REFERENCES

- [1] Morris, P.D. et al. Computational fluid dynamics modelling in cardiovascular medicine. *Heart* 2 (102), 18–28 (2016). doi:10.1136/heartjnl-2015-308044
- [2] PyFluent Documentation. PyAnsys Project web site: <https://fluent.docs.pyansys.com/version/stable/index.html>
- [3] Otta, M. et al. High-throughput statistical shape modelling of venous anatomy using containerised Deformetrica on Ares supercomputer; submitted to KUKDM 2026
- [4] Otta, M. et al. bioRxiv preprint, under review, 2026. doi:10.64898/2026.02.17.706277
- [5] Otta, M. et al. Towards Sensitivity Analysis: 3D Venous Modelling in the Lower Limb. ICCS 2025. doi:10.1007/978-3-031-97557-8_8

We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grants no. PLG/2025/018003 and PLG/2026/019160.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 857533 (Sano) and No 101136438 (GEMINI).



Ministry of Science and Higher Education
Republic of Poland

Funded by the initiative of the Minister of Science and Higher Education
"Support for the activities of Centers of Excellence established under the Horizon 2020 program"