CFD.xyz/ROM.js: An open-source framework for generating and visualizing parametric CFD results on the web

A Preprint

Carlos Peña-Monferrer∗
SIMZERO
Gran Vía Tárrega Monteblanco 17
12006 Castellón (Spain)
carlos@simzero.com

Carmen Diaz-Marin
Private person
mdc.diaz.marin@gmail.com

March 16, 2022

ABSTRACT

We present in this technical note an open-source web framework for the generation and visualization of parametric CFD results from surrogate models. It consists of a JavaScript module (rom.js) and a React JS web app (cfd.xyz) to explore fluid dynamics problems efficiently and easily for a wide range of parameters. rom.js is a JavaScript port of a set of open-source packages (Eigen, Splinter, VTK/C++ and ITHACA-FV) to solve the online stage of reduced-order models (ROM) generated by the ITHACA-FV tool. It can be executed outside a web browser within a backend JavaScript runtime environment, or in a given web solution. This methodology can also be extended to methods using machine learning. The rom.js module was used in cfd.xyz an open-source web service to deliver a collection of interactive CFD cases in a parametric space. The framework provides a proof of technology for OpenFOAM tutorials, showing the whole process from the generation of the surrogate model to the web browser. It also includes a standalone web tool for visualizing users’ ROMs by directly dragging and dropping the output folder of the offline stage. Beyond the current proof of technology, this enables a collaborative effort for the implementation of OpenFOAM surrogate models in applications demanding real-time solutions such as digital twins and other digital transformation technologies.

Keywords CFD · ROM · OpenFOAM · Parametric visualization · Web app · JavaScript · Data-driven

1 Introduction

Recent technological developments have made it possible to accelerate computational fluid dynamics (CFD) modeling through physics-based or data-driven surrogate modeling. This acceleration is key for enabling the integration of such solutions with real-life systems for dynamically controlling complex processes, but also allows an interactive analysis of the parametric space of CFD simulations.

Different techniques using machine learning (ML) and/or reduced-order modeling (ROM) applied to CFD have been described in the literature [1,2,3,4,5]. Furthermore, several open-source packages have been released in the past few years [7,8,9,10,11]. We aim with the developments presented in this work to create a shared space where canonical and industrial CFD problems can be visualized and analyzed without carrying out a simulation, or as a preliminary step for optimizing parameters of new simulations. Having an open-source centralized service has several advantages, not only from an educational, optimization and reproducibility point of views but also from a CO2 footprint perspective. Computation and data processing are associated with increasing greenhouse gas emission. For instance, an N-body simulation was evaluated with different programming languages, showing a carbon footprint of around 5 kg of CO2.

∗Corresponding author
Freitag et al. [13] reported that emissions of just one machine learning algorithm for natural language processing were 284,019 kg of carbon dioxide equivalent (CO₂ₑ), five times the life emissions of a car [14]. Although simpler training models represent a significantly lower figure of 4.5 kg of CO₂ [15], the use of high-performance computing simulations and ML is expected to increase significantly in the coming years.

With our development, we predict the following benefits for the community:

- **Educational enhancement**: The use of the web app can accelerate the learning of fluid dynamics by reducing the entry barrier of simulations. For example, a better understanding of how viscosity, gravity or inlet velocity affects fluid dynamics can be easily observed directly in the browser without any software installation or previous knowledge of how to use that software.

- **Optimization**: A preliminary analysis of similar physical problems through data-driven solutions can provide useful information for a given industrial application. This might help at early stage of designs by optimizing time and resources.

- **Reproducibility**: The framework provides reproducible examples of CFD and ROM for OpenFOAM tutorials that can be extended to cover more cases.

- **Worldwide duplicity reduction**: The use of CFD simulations around the world is increasing rapidly and is applied nowadays from very specific industrial fields to daily applications. While this is undoubtedly deepening our knowledge of fluid mechanics, it results in an inefficient use of resources in global terms. The adoption of an open-source web framework, whether sharing offline data, online data or both, would save multiple simulations of similar scenarios, thus reducing the carbon footprint.

- **CFD sharing flexibility**: The framework was designed to welcome open-source contributions of CFD cases (offline stage), but also to only showcase the surrogate model, thereby being compatible with organizations willing to protect their CFD models.

In addition, the web app also provides a standalone tool that directly connects the files of the surrogate model with the parametric visualization by only dragging and dropping the ROM output folder. This results in a convenient way of inspecting results of the surrogate models without generating a data file per parameter.

The work shown in this paper was focused on the link between visualization and surrogate models using modern web technologies. Different open-source tools for scientific visualization are already available. Kitware [16] developed a variety of software such as vtk.js [17], ParaViewWeb [18] and ParaView Glance [19] for visualizing scientific data on the web. Regarding the visualization of parametric simulations, RBniCS [20, 21] developed a Python framework for reduced-order modeling with FEniCS. Examples using RBniCS are shown in ARGOS [22] for different mathematical problems. Finally, ParaView recently integrated a Python plugin for viewing inference results and monitoring the training process in real time for deep-learning surrogates [23].

Our current work was focused on the release of two tools, rom.js [24] and cfd.xyz [25], both made open source. rom.js is a JavaScript port of different packages such as Eigen [26], Splinter [27], VTK/C++ [28], and ITHACA-FV [4, 5, 8] for interactively solving the online phase of reduced-order models generated by ITHACA-FV. Finally, cfd.xyz is a React JS web app that integrates rom.js and vtk.js to deliver a user-friendly, modular tool. A starting point for the project is to showcase an example for a turbulent steady-state OpenFOAM tutorial (pitzDaily) showing how to generate the surrogate models and integrate them in the web app. Future work will focus on integrating other ROM techniques, ML and CFD packages on the framework and incorporating other industrial problems.

## 2 Methods

A general overview of the presented tools is shown in Fig. 1. The JavaScript module, rom.js, for online reduced-order modeling was developed and made available publicly in an npm registry as @simzero/rom [29] and a Git repository [24]. The React JS web app, cfd.xyz, is an open-source and cross-platform tool for generating and visualizing CFD data. It is released as a service at https://cfd.xyz and in a Git repository [25]. The results shown in this work can be reproduced with the v1.0.0-rc.2 versions of these repositories.

### 2.1 rom.js - An open-source JavaScript module for the online stage of reduced-order modeling

The module rom.js contains a WebAssembly ported version of Eigen, Splinter, VTK/C++ and ITHACA-FV to compute the online stage of ROM models generated by ITHACA-FV. It supports turbulent and laminar steady-state cases in the current version. The module consisted of an optimized single JavaScript file of around 5 MB, which is relatively small considering the kind of problem to be solved. A small subset of Eigen was used. The main use of Eigen in this
package is dealing with matrix operations and solving the ROMs online stage. Splinter was ported to compute the
turbulent viscosity using radial basis functions as described in [30]. Finally, VTK/C++ was used to reconstruct
the calculated values from Eigen to the original VTK unstructured grid, and to apply VTK filters. It is worth noticing
that vtk.js supports web visualization, but not unstructured grids and related components. The integration of VTK/C++
in rom.js allows the reconstruction of the calculated volumetric fields to VTK format. It also enables a later connection
with vtk.js components for the visualization in a web browser environment. The repository contains examples that can
be executed within a runtime environment such as Node.js.

A workflow was developed for generating the CFD snapshots directly with OpenFOAM. These snapshots were later
used for building the ROM with a new application based on the ITHACA-FV library. Around 800 simulations were
performed using a massively parallel approach of single core runs for the pitzDaily tutorial. ROM data and command
line user inputs of inlet velocity and viscosity were then used to generate the new fields. The total execution time for
calculating the new fields and reconstructing them was 0.06 s. A comparison of the velocity magnitude for OpenFOAM
(offline), ITHACA-FV C++ (online) and the rom.js ported version are shown in Fig. 2 for three different conditions.

Figure 2: Comparison of velocity magnitude for OpenFOAM, ITHACA-FV (C++) and rom.js.

2.2 cfd.xyz - An open-source web platform for generating and visualizing of CFD data

A single page React JS app, cfd.xyz, was designed in a modular way to include content containers in the form of cards
for the different OpenFOAM tutorials. Every tutorial available in the platform contains the interactive visualization of a
case for a range of different parameters (e.g., viscosity, inlet velocity, ...). The data for a given set of parameters can be
also downloaded for further post-processing on a dedicated desktop or web tool. Fig. 3 shows screenshots of different
stages of the web app.

The web app is client-based and the communication with the server is mainly for sending the bundle to the user. The
visualization of the cases for every set of parameters was achieved with vtk.js and rom.js. Fig. 4 shows an overview of
this process for a given case. The volumetric mesh (unstructured grid) and ROM data is fetched from remote or local
storage and initialized in the WebAssembly side with rom.js. A rendering scene is initially set-up and rendered. The user
can interact with the 3D view and modify the parameters defined for the case. A change of these parameters triggers the
calculation and visualization of the new fields. Additionally, the data can be directly visualized or downloaded as VTK
files or images.
3 Conclusions

We presented in this technical note an open-source framework for parametric generation and visualization of CFD data on the web. The framework consisted of two main components: a JavaScript module (rom.js) and a React JS web app (cfd.xyz). Working together, these tools create a shared space where canonical and industrial CFD problems can be visualized and analyzed on the web without carrying out a simulation. The models and workflows for generating the surrogate models are also shared in a reproducible way. New cases can be easily integrated on the web with the implemented methodology. The current version is based on OpenFOAM as the CFD tool and ITHACA-FV as the reduced-order modeling (ROM) one. Further implementations will be considered for including other ROM or machine learning packages, as well as other CFD tools.

The ported code was applied to the pitzDaily OpenFOAM tutorial, resulting in a JavaScript module of 5 MB and an execution time of 0.06 s. for calculating and reconstructing new fields for a given set of parameters. This permitted a smooth interaction through the different parameters on the web version, and enables future implementations of web tools using CFD.
Figure 4: High-level overview of the components in cfd.xyz.
References

[1] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, and F. Chinesta, “Proper generalized decomposition based dynamic data-driven control of thermal processes,” *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, pp. 29–41, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782511003641

[2] C. Ding and K. P. Lam, “Data-driven model for cross ventilation potential in high-density cities based on coupled cfd simulation and machine learning,” *Building and Environment*, vol. 165, p. 106394, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0261926819300043

[3] H. Ganti and P. Khare, “Data-driven surrogate modeling of multiphase flows using machine learning techniques,” *Computers & Fluids*, vol. 211, p. 104626, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793020301985

[4] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza, “POD-Galerkin reduced order methods for CFD using Finite Volume Discretisation: vortex shedding around a circular cylinder,” *Communications in Applied and Industrial Mathematics*, vol. 8, no. 1, pp. 210–236, (2017).

[5] G. Stabile and G. Rozza, “Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations,” *Computers & Fluids*, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793018300422

[6] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, “Machine learning–accelerated computational fluid dynamics,” *Proceedings of the National Academy of Sciences*, vol. 118, no. 21, 2021. [Online]. Available: https://www.pnas.org/content/118/21/e2101784118

[7] Illinois Rocstar LLC’s, “AccelerateCFD Community Edition,” https://github.com/IllinoisRocstar/AccelerateCFD_CE, accessed: 28.02.2022.

[8] SISSA mathLab, “ITHACA-FV - In real Time Highly Advanced Computational Applications for Finite Volumes - ROMs for OpenFOAM,” https://mathlab.sissa.it/ITHACA-FV, accessed: 28.02.2022.

[9] A. Weiner, “Machine learning applied to CFD,” https://github.com/AndreWeiner/machine-learning-applied-to-cfd, accessed: 28.02.2022.

[10] P. Pant, “DL-ROM: Deep Learning for Reduced Order Modelling,” https://github.com/pranshupant/DL-ROM, accessed: 28.02.2022.

[11] R. Maulik, “In-situ data analyses and machine learning with OpenFOAM and Python,” https://github.com/argonne-icf/PythonFOAM, accessed: 28.02.2022.

[12] S. P. Zwart, “The Ecological Impact of High-performance Computing in Astrophysics,” *Nature Astronomy*, vol. 4, no. 9, pp. 819–822, Sep. 2020, arXiv: 2009.11295. [Online]. Available: http://arxiv.org/abs/2009.11295

[13] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, “The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations,” *Patterns*, vol. 2, no. 9, p. 100340, 2021.

[14] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in NLP,” *arXiv:1906.02243 [cs]*, Jun. 2019, arXiv: 1906.02243. [Online]. Available: http://arxiv.org/abs/1906.02243

[15] L. Biewald, “Deep learning and carbon emissions.” [Online]. Available: https://towardsdatascience.com/deep-learning-and-carbon-emissions-79723d5bc86e

[16] Kitware Inc., “Building Solutions on Open Source Technologies,” https://www.kitware.com/open-source, accessed: 28.02.2022.

[17] Kitware Inc., “vtk.js - The Visualization Toolkit for JavaScript,” https://kitware.github.io/vtk-js, accessed: 28.02.2022.

[18] Kitware Inc., “ParaViewWeb - Web framework for building interactive visualization relying on VTK or ParaView to produce visualization data,” https://www.paraview.org/web, accessed: 28.02.2022.

[19] Kitware Inc., “ParaView Glance - An open-source web application for visualizing volumetric images, molecular structures, geometric objects, and point clouds.” https://kitware.github.io/paraview-glance, accessed: 28.02.2022.

[20] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, ser. SpringerBriefs in Mathematics. Springer International Publishing, 2015. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-22470-1

[21] RBniCS Project, “RBniCS - reduced order modelling in FEniCS,” https://www.rbnicasproject.org, accessed: 28.02.2022.
[22] SISSA mathLab, “Advanced Reduced Groupware Online Simulation Platform,” accessed: 28.02.2022. [Online]. Available: https://argos.sissa.it

[23] F. Mazen and A. Schieb, “Deep-learning surrogate models in ParaView: Viewing inference results and monitoring the training process in real time with Catalyst,” https://www.kitware.com/deep-learning-surrogate-models-in-paraview-viewing-inference-results-and-monitoring-the-training-process-in-real-time-with-catalyst, accessed: 28.02.2022.

[24] SIMZERO, “rom.js - A JavaScript module for solving the online stage of ROM,” https://github.com/simzero-oss/rom-js, accessed: 28.02.2022.

[25] SIMZERO, “cfd.xyz - An open-source web app for generating and visualizing parametric CFD results,” https://github.com/simzero-oss/cfd-xyz.

[26] G. Guennebaud, B. Jacob et al., “Eigen v3,” http://eigen.tuxfamily.org, 2010.

[27] B. Grimstad et al., “SPLINTER: a library for multivariate function approximation with splines,” http://github.com/bgrimstad/splinter, accessed: 28.02.2022.

[28] W. Schroeder, K. Martin, and B. Lorensen, The Visualization Toolkit - An Object-Oriented Approach To 3D Graphics, 4th ed. Kitware, Inc., 2006.

[29] SIMZERO, “rom.js JavaScript module,” https://www.npmjs.com/package/@simzero/rom

[30] S. Hijazi, G. Stabile, A. Mola, and G. Rozza, “Data-driven POD-Galerkin reduced order model for turbulent flows,” Journal of Computational Physics, vol. 416, p. 109513, Sep. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999120302874