The Journal of Open Source Education

ARC4CFD: Learning how to leverage High-Performance Computing with Computational Fluid Dynamics

Jean-Pierre Hickey [1*], **Francesco Ambrogi** [1*], **Sophie Hillcoat** [1], **Jeswin Joseph** [1], and **Nipin Lokanathan** [1]

1 Department of Mechanical and Mechatronics Engineering, University of Waterloo, Canada ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 18 April 2024
Published: unpublished

Summary

Computational Fluid Dynamics (CFD) is a field of computational physics that has a very high utilization of modern Advanced Research Computing (ARC) resources (Cant, 2002). The spatial and temporal resolution required to solve modern CFD problems means that it is well suited to take advantage of the full benefits of large-scale distributed memory parallelization that is available on high-performance computing (HPC) systems on ARC infrastructure. The field of CFD has a broad and diverse user base that transcends many of the classical boundaries in science and engineering. As CFD tools have progressed over the past decades, their robustness, predictive capabilities, and user-friendliness have drastically improved, which means that these tools are increasingly being adopted by nontraditional HPC users such as new graduate students, experimentalists, theoreticians, and student design teams. Advanced Research Computing for Computational Fluid Dynamics, or ARC4CFD, is an open source, asynchronous online course (https://arc4cfd.github.io) that is developed to help learners with a basic understanding of fluid dynamics and CFD bridge the knowledge gap toward the effective usage of CFD on modern ARC resources.

Statement of need

Although most science and engineering programs offer CFD courses, which have become standard part of the curriculum in mechanical, civil, chemical, and aerospace engineering, these courses generally focus on the fundamental understanding of the physics, modelling, and numerics as well as the practical usage of CFD tools. At the end of a typical undergraduate CFD course, the student has learned to use a CFD tool to solve small-scale problems, understand the modelling assumptions, quantify numerical errors, and visualize the results. The CFD problems used in the tutorials of these courses are designed to run on students' personal or local workstations. The jump from small-scale CFD usage on a local workstation to the effective utilization of this same CFD tool, on a much larger problem size, on modern HPC systems is nontrivial and requires additional specific training. Although generic HPC and parallel computing training is widely available, targeted training material for CFD users on HPC systems is not readily found. Thus, these learners must either: a) "translate" the knowledge from generic HPC training material to the field of CFD, or b) rely on mentorship and external help to effectively utilize these computational resources. As these CFD tools are continually improving, they are increasingly being adopted to complement experimental campaigns, inform the design of experiments, or help solve theoretical fluid dynamics problems. This underserved user base is seeking focused training materials for the effective utilization of CFD on HPC systems, as they do not have the same mentorship opportunities or access to expertise as graduate students. ARC4CFD was developed to provide a CFD-specific training material for the

The Journal of Open Source Education

effective use of HPC systems in advanced research computing architectures. The course is built in an asynchronous format for a broader adoption, and many hands-on problems are integrated to help the learner further their understanding and learn independently. We value the ability to freely share and exchange with the community, and thus the focus is placed on an entirely open-source toolset, from meshing, solving, and visualizing the results. Although this course is intended for use on remote HPC systems, and more specifically on the Digital Research Alliance of Canada clusters, a user can easily follow on a local multiprocessor computer, or any other HPC system.

ARC4CFD is intended to be an introductory course that combines concepts from CFD and computing with an *end-user focus*. The course is built with the assumption that students have: a) a undergraduate-level knowledge in fluid dynamics, b) introductory knowledge in Computational Fluid Dynamics, and c) familiarity with navigating terminal and bash commands on remote systems. The target audience is: - New graduate students in computational physics or engineering. - Experimentalists and theoreticians complementing their work with numerical simulations on HPC. - Undergraduate students on student design teams interested in leveraging CFD with HPC.

Given the prerequisite knowledge and target audience, the end-user focus means that the course needs to provide both the theoretical understanding of parallel computing and the systematic, hands-on approach to set up an effective workflow for CFD on HPC. The classes are developed to first provide a conceptual understanding and then apply that understanding to specific CFD problems; a summative example is woven into the course which follows the learner through the main CFD workflow.

Course structure and learning outcomes

The course is expected to take about 16 hours to complete and is divided into three sections. Each section consists of several classes with individually-defined learning outcomes. The sections are structured in a way that the learner can first develop a foundational understanding of high-performance computing (Section 1), translate those concepts to specific challenges in CFD and establish a systematic workflow (Section 2), and then effectively manage the resulting research data (Section 3). The course guides the learner through all the necessary steps towards the effective usage of CFD on HPC, by providing a theoretical understanding of the concepts, which is supplemented with many practical examples and interactive quizzes. Section 2, the core section of the course, provides most of the CFD-specific knowledge for usage in HPC; for that reason, the classes in this section were devised to directly map onto a CFD workflow on HPC system. Therefore, each step in the CFD workflow is an individual class, so that we have:

- 2.1 Definition of the CFD workflow.
- 2.2 Plan the large-scale CFD simulation.
- 2.3 Estimate the HPC requirements.
- 2.4 Preprocess the CFD simulation.
- 2.5 Optimize the CFD for HPC.
- 2.6 Running the CFD on HPC.
- 2.7 A posteriori analysis.

The structure of Section 2 provides the learner with a comprehensive overview of a standard workflow for a CFD problem on HPC systems. A summative example, which is based on the simulation of a backward facing step, is integrated into the classes of Section 2 and follows the main steps of the CFD workflow.

There are five learning outcomes for ARC4CFD. At the end of the course, the learner should be able to:

1. Define the main concepts of parallel and high-performance computing.

2. Conduct an a priori estimate of the computational cost of a CFD simulation.
3. Explain the modelling impact of the assumptions on HPC cost.
4. Optimize the simulation parameters of a CFD problem for HPC.
5. Develop a research data management strategy for a CFD workflow.

The learning outcomes for each class are presented and reinforced with questions at the end of each lecture.

Philosophy of learning

ARC4CFD is an asynchronous online course that accompanies the learner in developing skills to effectively utilize Advanced Research Computing for Computational Fluid Dynamics. The course is built using an *integrative learning* approach in which concepts from a wide range of scientific disciplines are brought together (e.g., parallel computing, programming, numerics, fluid dynamics, CFD) and directly put into practice through hands-on examples and quizzes. This course is built around the usage of an entirely open-source meshing, solving, and visualization toolset. This is a key feature of the course that we hope will encourage a broad adoption of the training material. For this reason, the course is licensed under the Creative Commons BY-NC-SA license, and all the original files (e.g., svg files or Python scripts) are included in the repository. The main open-source toolkits used include: Gmsh (meshing), SU2 (CFD solver), OpenFoam (CFD solver), Paraview (visualization), and Python (postprocessing and analysis).

The course is developed to: 1) emphasize the development of a systematic approach for the effective utilization of CFD usage on ARC systems, 2) provide a high-level theoretical understanding of the main concepts for using CFD on HPC, and 3) provide hands-on examples for the learners to put these concepts into practice. The multimodal course content includes written content, videos, interactive quizzes, and hands-on examples that cover various aspects of ARC.

Course features

The asynchronous courses are built within a GitHub Pages website ARC4CFD using the static site generator Astro and the Astro Starlight theme. This Astro template enables highly interactive engagement through the use of self-correcting quizzes, direct copy-pasting of code snippets, and visually appealing course navigation. The main website is supplemented with a more traditional Git repository which contains all the examples used in the course. Many of the courses feature a short introductory video to guide the learners through the context of each class.

A unique aspect of the training material is found in Section 3 on Research Data Management (RDM). As RDM approaches are continually being integrated into scientific disciplines, we opted to provide some CFD-specific perspective to the RDM discussion. The content will undoubtably evolve with these concepts, but we hope that this training material will be of assistance to CFD users that are increasingly being asked to develop Data Management Plans (DMPs) as part of grant proposals.

Story of the project

This course was conceived primarily from the need to train new graduate students in the Multi-Physics Interaction Lab at the University of Waterloo to bridge the gap between their undergraduate CFD education and the effective utilization of these CFD tools on high-performance computing. The typical undergraduate CFD coursework focuses on the numerical modelling, and usage of CFD tools, whereas the majority of HPC training

The Journal of Open Source Education

material tends to emphasize the computer science aspects of HPC. The effective utilization of modern Advanced Research Computing facilities requires the integration of knowledge from both CFD and HPC. It was from the need to centralize the information on the usage of CFD in ARC that the idea for this course was born. Previously, this information was transmitted, piecemeal, to students as needed during their graduate training. This led to inhomogeneous training, which often left unresolved knowledge gaps for these otherwise highly qualified graduate students.

A timely opportunity arose during a sabbatical through a call for proposals from Compute Ontario for the development of training material for underserved HPC users. This opportunity provided the means to build a team to centralize the necessary information and build this course. The team built the course by focusing on three user groups: 1) new graduate students that plan to use HPC resources to run CFD simulations, 2) theoreticians and experimentalists that want to use CFD on HPC to supplement their work, and 3) undergraduate design team members who want to develop skills in HPC. Inspired by the "12-steps to the Navier-Stokes" course (Barba & Forsyth, 2019), which is a staple of our graduate student training, we tried to recreate a very systematic approach towards the usage of CFD on HPC systems. The initial idea of building everything within a Jupyter notebook eventually shifted to an interactive webpage with video content due to the challenges of directly using HPC systems through a pythonic interface.

Looking ahead, this course will continue to evolve over the coming year and will serve as the basis for a synchronous course that will be given as part of Compute Ontario's summer school. As learners benefit from the course and we continually adjust the course based on user comments, we will be able to improve the content to meet the changing needs of new graduate students in CFD, experimentalists and theoreticians, and undergraduates that are interested to learn CFD on HPC.

Acknowledgments

This course was developed in the Multi-Physics Interaction Lab (www.mpilab.ca) with financial support from Compute Ontario.

References

Barba, L., & Forsyth, G. (2019). CFD python: The 12 steps to navier-stokes equations. *Journal of Open Source Education*, *2*(16), 21. https://doi.org/10.21105/jose.00021

Cant, S. (2002). High-performance computing in computational fluid dynamics: Progress and challenges. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *360*(1795), 1211–1225.