# VERIFICATION & VALIDATION process in the open-source TRUST/TrioCFD platform

A.BRUNETON, J. DARONA, R. NOP, P. LEDAC, E. SAIKALI, A. KHIZAR, U. BIEDER, N.DORVILLE, E.ADAM

Université Paris-Saclay, CEA, Service de Thermo-hydraulique et de Mécanique des Fluides, 91191, Gif-sur-Yvette, France.

Email contact of corresponding author: adrien.bruneton@cea.fr

* Overview of TRUST/TrioCFD Platform

TRUST/TrioCFD [1] is an open-source platform of fluid mechanics and thermohydraulics developed at CEA. The latter is made up of 2 codes:

* TRUST in which are implemented the general architecture and all the basic IT tools namely the classes and methods necessary to solve basic problems (operators, standard Navier-Stokes, solvers,...). Several advanced codes specialized to treat specific industrial applications are developed at CEA based on TRUST (e.g. steam generator, reactor core, chemical reactions). In each of them, specific advanced models are implemented from the TRUST toolbox and benefit from its massive parallel capabilities;
* TrioCFD is one of the advanced codes developed at CEA based on TRUST. It includes specific and advanced models, particularly in fluid mechanics and heat transfer modelling. TrioCFD is therefore able to solve more complex problems than laminar flows (or turbulent flow, but in DNS) such as turbulent flow models, fluid-structure interactions, multiphase flows or flows in porous media.

This allows the resolution of the incompressible Navier-Stokes equations to which can be added more complex models, according to a specific problem to be investigated (through TrioCFD), and offers massive parallel computing capabilities (through TRUST). Initially, designed for nuclear industry, it can also be used for a wide range of applications. Meanwhile the platform is flexible: its object-oriented architecture developed in C++ allows for the development of target-oriented applications as fast-neutron reactor simulation, nuclear propulsion, but also now fuel cells simulations or chemistry.

Developed for more than 20 years, TRUST/TrioCFD is open-source since 2015 (under the 3-Clause BSD license). The motivation for this change is two-fold:

* facilitate the collaborations with non-CEA academic or industrial partners;
* and consequently enhance the visibility of the platform at both the European and international level.

Practically speaking, TRUST/TrioCFD can be downloaded from the SourceForge website [2], and is currently being migrated to GitHub [3].

In terms of development, the code is version-controlled under GIT [4] since 2014, and a strict development process has been implemented. Each new major functionality is coded in a dedicated branch, whose integration into the main stream is only performed once:

* a code review by peer developers has been made;
* a whole set of automatic tests has been passed.
* Verification and Validation process on TRUST/TrioCFD Platform

Similarly to other simulation codes, verification and validation (V&V) are very important steps in the development of TRUST/TrioCFD. An on-going work is therefore being done to improve methods and tools on these V&V aspects.

A first *verification* step is carried out every evening via a home-made process based on Bash and Python scripts. The development branch of the two codes is extracted from their GIT repositories. The correct compilation and documentation generation are checked. A battery of tests (826 for TRUST and 1531 for TrioCFD) is run every night on a set of about 25 machines or clusters covering a large number of different Linux OS (CentOS, RedHat, Ubuntu, Fedora, Debian) with different releases and compilers. This battery of tests contains simple verification tests and all data-cases extracted from the validation tests. Only the first three time steps of the data-cases extracted from de validation database are run. This daily check ensures that no problem (such as unexpected physical impacts) is introduced in the version. Thanks to this process, if a problem occurs, a corrective action can immediately be taken.

The *validation* process is carried out at least every weekend or at each integration that may cause significant physical impacts. The results are compared with the last run of the validation database in order to know the impact of each integration. The validation set is checked automatically through a dedicated Jenkins server [5] checking the TRUST platform itself, but also its many derived applications including TrioCFD. The Jenkins pipeline used for this process has been written in a generic fashion (Groovy script), allowing to easily select the mode to perform (switching the tests in Release or Debug mode, executing the valgrind checks or not, etc.).

On top of this, the analysis of the outcome is conducted in several ways: either via the production of dedicated files that can be exploited with standard scientific visualisation softwares (ParaView and/or VisIt), or via the production of ASCII text files, corresponding to the listings of some quantity of interest at given spatial points in the domain (“probes”).

These results can then be gathered and formatted in a unique document: the end-user typically enters the text, the mathematical formulas (LaTeX-based) and some dedicated commands in a custom file which is then compiled to produce a PDF report, for which an example is shown in Figure 1 below:

*Fig. 1. Example of PDF report for TRUST/TrioCFD report*

The corresponding PDF reports serve two purposes:

* presenting the results to peers;
* allowing a high-level non-regression test process, whereby those reports are automatically compared with their counterpart from the previous major versions of the two codes. This complements the lower level tests already in place in the platform.

For TrioCFD, the validation sheets representing the most relevant simulations of each expertise area (laminar flow, thermal laminar, turbulent, thermal turbulent, fluid-structure interactions, two-phase flows with Front-Tracking) are grouped together in a validation report which is now present at each code delivery. It allows the users to be informed about the models and options to be used for each problem, to provide modelling recommendations and also to provide a state of the version at each delivery.

More recently, a new functionality has been added to the platform whereby the user can format the output data of the code using the standard Jupyter Notebooks [6]. An example of such a notebook is illustrated below in Figure 2:

*Fig.2. Example of a Jupyter Notebook for TRUST/TrioCFD report*

A dedicated Python API allowing an easy extraction of the quantities of interest from the output files, along with the rendering of relevant 3D visualizations of the simulation (using VisIt) has been implemented. More advanced users also have the possibility to directly use standard Python packages such as MatPlotLib [7].

Thanks to a daily verification process and a weekly validation, TRUST/TrioCFD benefits from a continuous integration of its developments/corrections. The origin of the physical and numerical impacts observed on the platform is thus swiftly and precisely identified. The verification/validation base ensures a coverage rate of 79.97% of the platform’s keywords. The platform’s development team is committed to continuously enriching these tools but also to improving the already substantial V&V coverage rate.

.

References

1. ANGELI P.-E., BIEDER U., FAUCHET G.: “Overview of the TrioCFD code: Main features, V&V procedures and typical applications to engineering”, NURETH-16, Chicago, IL, August 30-September 4, 2015
2. SourceForge Website to download TRUST/TrioCFD codes

https://sourceforge.net/projects/trust-platform/

https://sourceforge.net/projects/triocfd/

1. Development Platform GitHub

https://fr.github.com/

1. GIT

https://git-scm.com/

1. Jenkins

https://www.jenkins.io/

1. Description of standard Jupyter Notebooks

https://jupyter.org/

1. MatPlotLib

https://matplotlib.org/