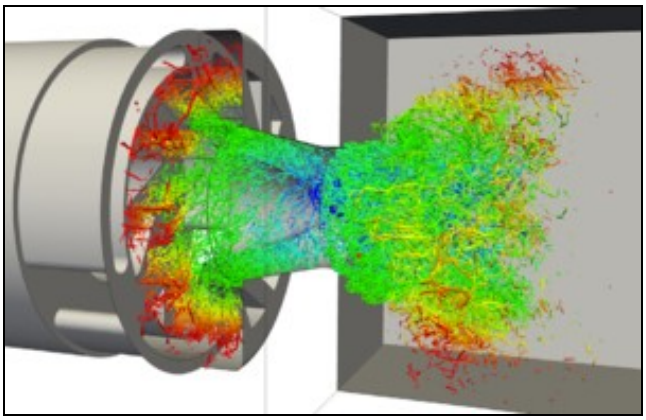
YALES2

xxx-CustomTitleStart-xxxYALES2 public pagexxx-CustomTitleEnd-xxxxxx-CustomPageTitleStart-xxxYALES2 public page - www.coria-cfd.frxxx-CustomPageTitleEnd-xxx



PRECCINSTA burner with YALES2

Contents

Motivation

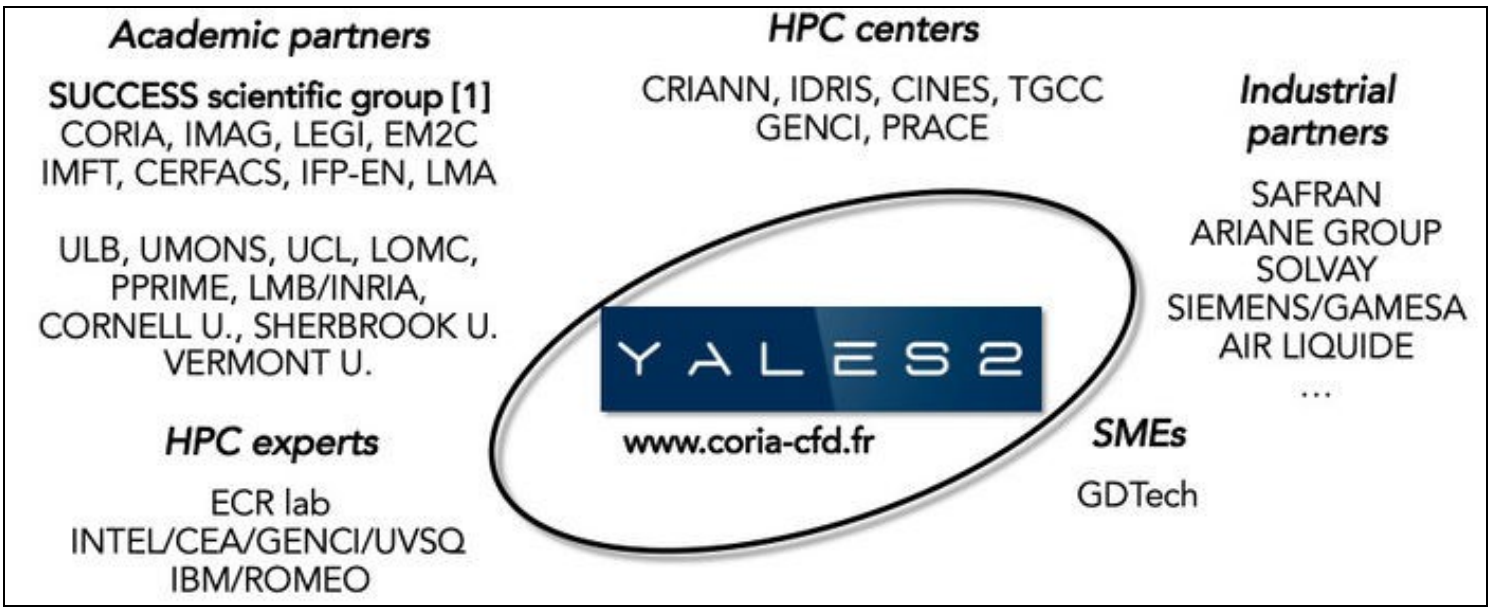
YALES2 aims at the solving of two-phase combustion from primary atomization to pollutant prediction on massive complex meshes. It is able to handle efficiently unstructured meshes with several billions of elements, thus enabling the Direct Numerical Simulation of laboratory and semi-industrial configurations.

YALES2 was developed from 2007 to 2010 by V. Moureau and is maintained since 2011 by V. Moureau and G. Lartigue, joined later by P. Bénard and K. Bioche at CORIA and several other people in research laboratories.

More information may be found in the following presentation: YALES2 presentation

Community

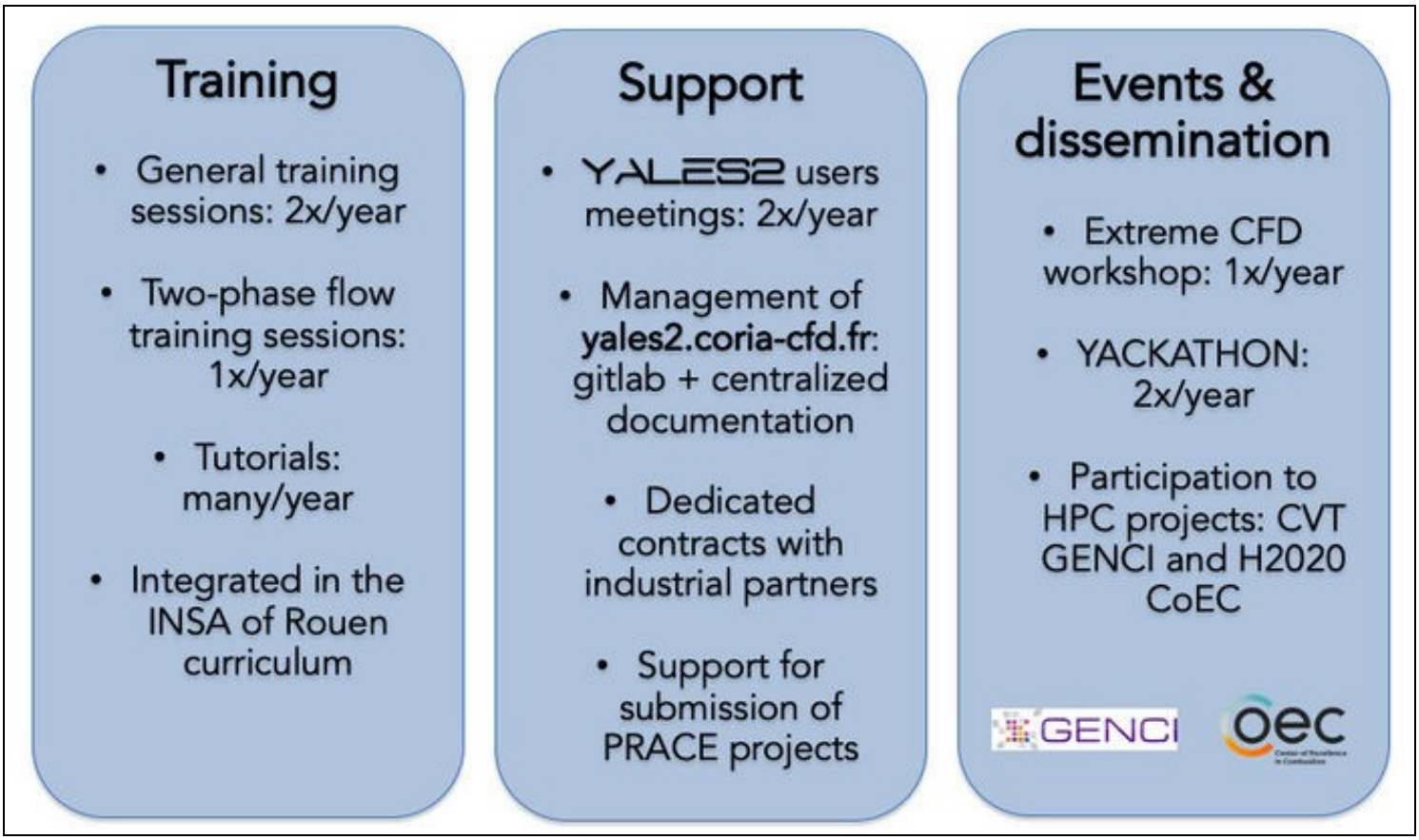
YALES2 is developed by a large community with more than 500 researchers/engineers who were trained by the CORIA laboratory since 2009. The community regroups academic partners, HPC centers, industrial partners, HPC experts, SMEs and more. The code is also used for CFD training in academic courses at INSA of Rouen in the Energy and Propulsion department.



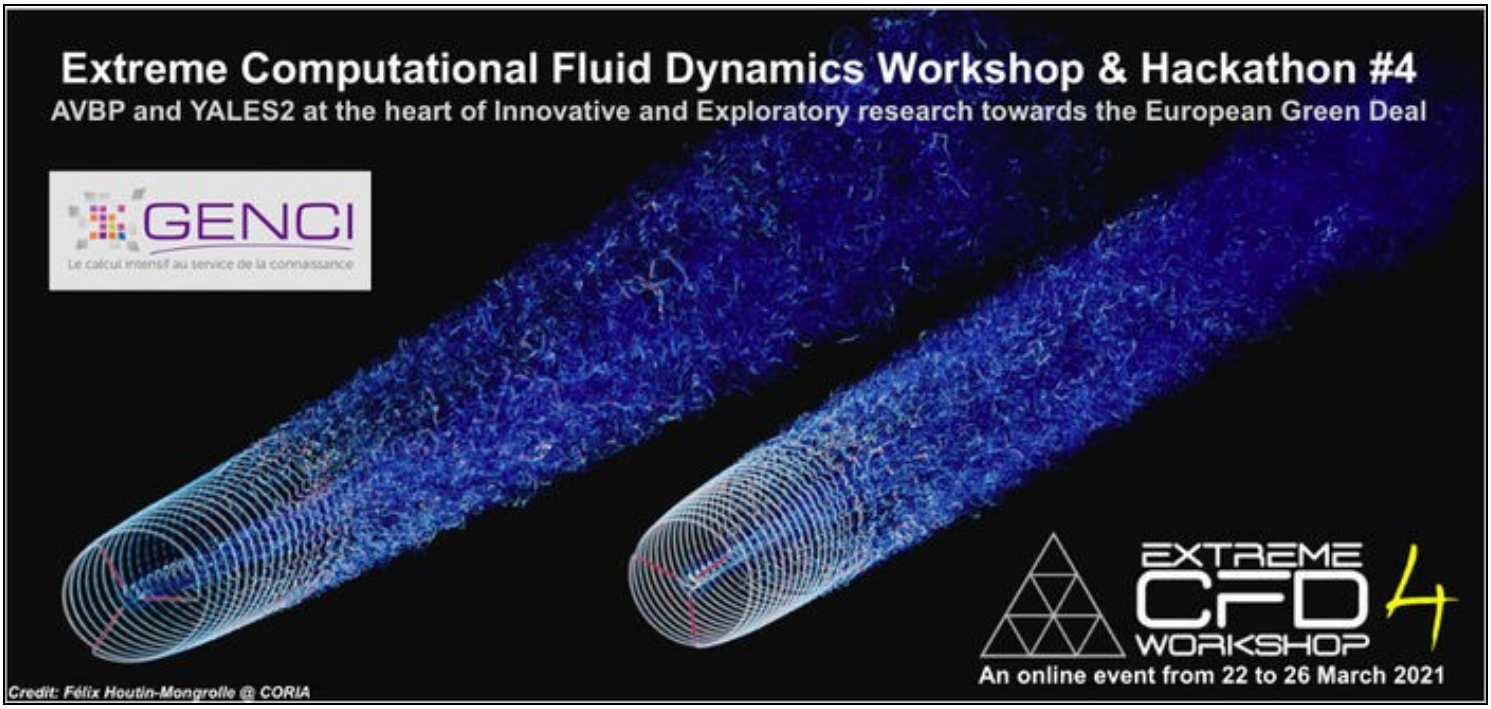
YALES2 network

Commitments

The YALES2 team is committed to supporting code users through training, meetings, projects or events.



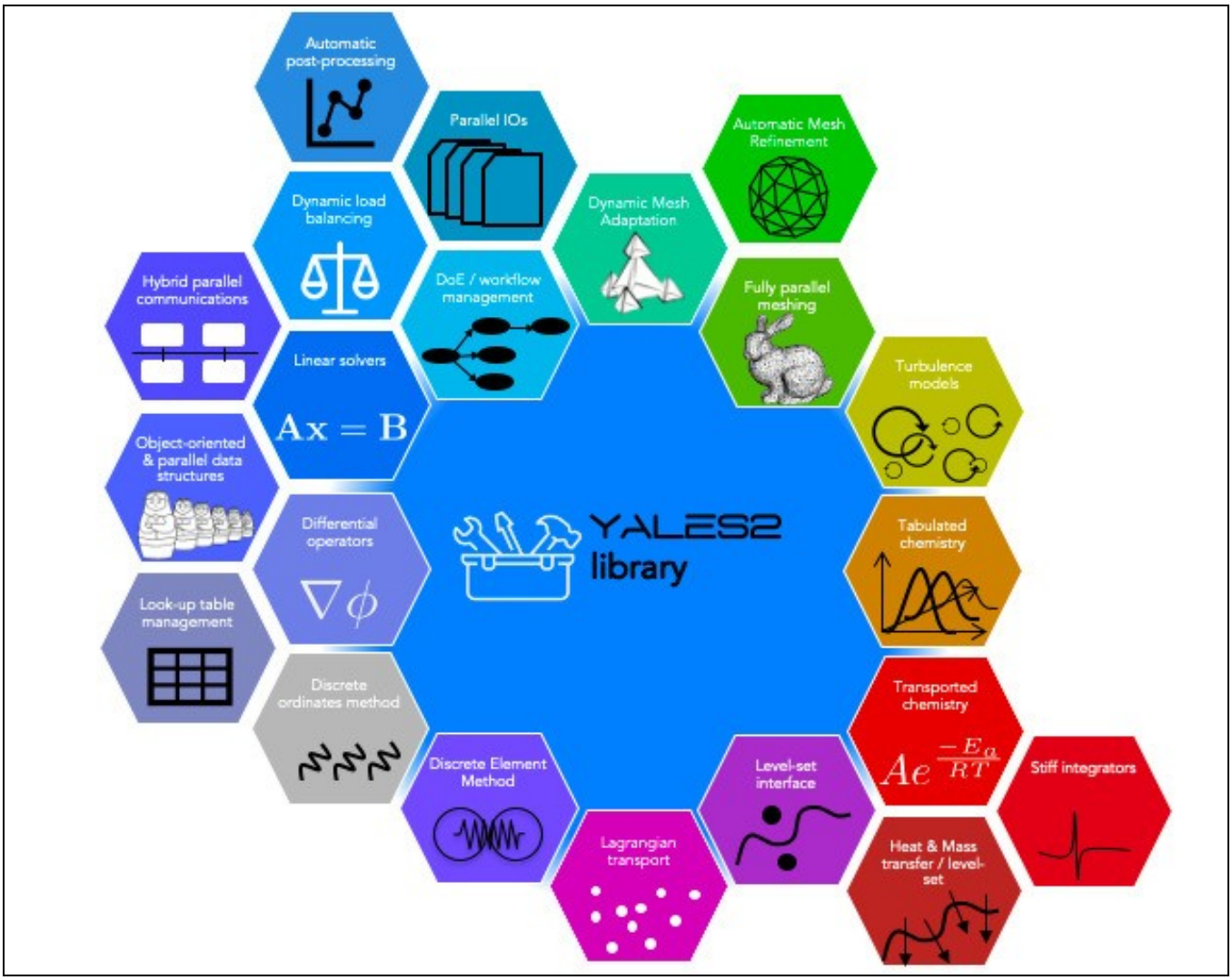
YALES2 team commitment
Here an example of event you can participate to:



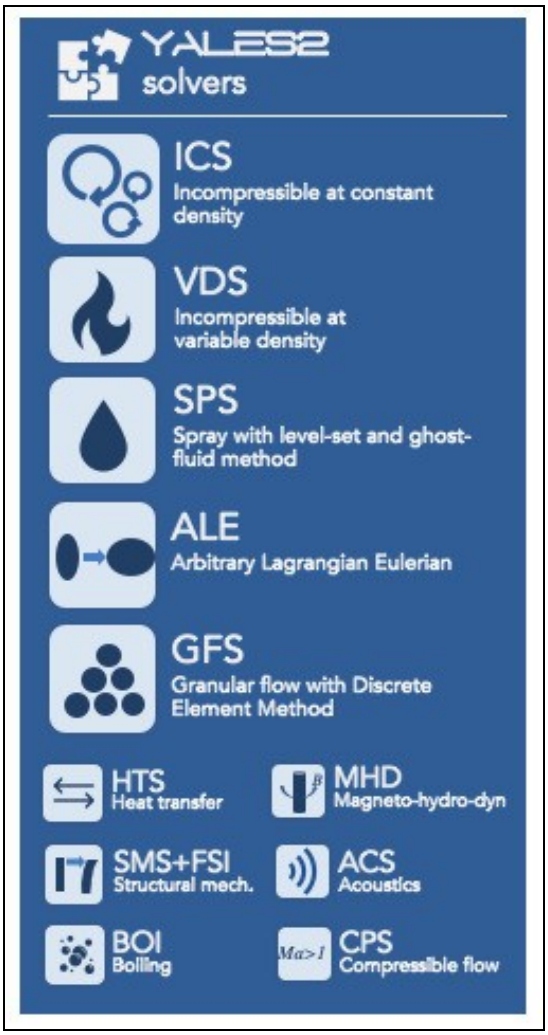
Extreme CFD event, https://ecfd.coria-cfd.fr/index.php/Ecfd:ecfd_4th_edition

YALES2 Library and solvers

The numerical library YALES2LIB consists of all the numerical methods required to develop solvers:



YALES2 library
We have plenty of solvers today, here are the main ones:



YALES2 solvers

Agile development

The fast development of the YALES2 platform comes mainly from the agile development project management methodology. It relies on a number of tools:

- programming: modular structure of the code with more than 200 objects and 420 modules
- non-regression and testing: private gitlab forge, nightly pipelines with more than 300 automatic jobs
- fast compiling: automatic dependencies, two pass compiling, 1m15s to compile 850'000 lines of fortran
- easy debugging: 2 compilation modes (optim, debug), many helpers (memory consumption, number of arrays, ...)

A few figures:

- 16 major releases since 2007
- 850 000 object-oriented Fortran 2008 lines for YALES2_2023.04
- 15 600+ commits
- 200+ active branches
- 1000+ merge requests
- 600+ members on the gitlab projects
- 100+ contributors

Gallery

Some computation examples are given in the gallery and on the Youtube video channel [1]