CV2 OpenFoam Documentation

Laurence Bonneau, Vincent Robin

July 2020

1 Main directories

- The *solver* directory contains the Open-Foam solver (based on sonicFoam) designed to CV2 simulations. The current version *CV2Foam250620* is stored in the *applications* folder. Specific wall of the law has also been developed and stored in the *src* folder.
- The *preprocess* directory contains (i) the OpenFoam files required to run a case, stored in *CV2cyclefiles* (these files are those that change for each simulation phases of each cycle) and *CV2foamfiles* folders, (ii) the meshes source files generated with *gmsh* application (not ready yet), (iii) the python script necessary to create the data files *CV2inoutBC-???.csv* containing the inlet and outlet mass flow rates used to run a case and (iv) the script creating the runcase directory *CV2-run-All*.
- The *postprocess* directory contains some scripts (python, gnuplot, bash) to automatically store data or create figures (not ready yet : to clean).
- The *CV2-run-All* directory is created using the script gen-runcase of the preprocess directory. This is the OpenFoam runcase directory and it contains all the necessary files to run the simulation.

2 CV2 brief description

CV2 is a constant volume combustion chamber that processes cyclically: air intake, fuel injection, ignition, combustion and exhaust. Its behavior is similar to a piston engine but without moving parts, see references [1], [2], [3]. The computational volume is limited to the combustion chamber up to the intake and exhaust valves. The inlet and outlet mass flows rates has been experimentally determined from pressure signals and are imposed as boundary conditions. The corresponding data are stored in the files *CV2inoutBC-???.csv*. There are five physical phases: intake, mixing, combustion, isochoric cooling, exhaust. But only two phases are defined numerically: phase A is intake to mixing and phase B mixing to exhaust. The duration of one cycle is 0.125s. The boundary conditions have to be changed for each numerical phase of each cycle. Accordingly, the

initial and boundary conditions corresponding to each numerical phase of each cycle are stored in the subdirectory *CV2cyclefiles* of the *preprocess* directory. The files are named with the initial time of the corresponding phase.

- Phase A of cycle 0 : 0
- Phase B of cycle 0 : 0.03
- Phase A of cycle 1 : 0.125
- Phase B of cycle 1 : 0.155
- Etc.

The first cycle 0 is not reactive. The modeling parameters specific to CV2 simulations have been obtained from the measured pressure signal (stored in *python-scripts/data-exp.csv*) and are stored in the file *CV2param???* of the directory *CV2cyclefiles*.

3 compiling and running the first case

3.1 Setting environments variables on Thor

- **module load codes/openfoam/4.1**
- **source /sw/codes/OpenFOAM-4.1/OpenFOAM-4.1/etc/bashrc**

3.2 Compiling CV2 specific source files

- in *solver/applications/CV2Foam250620* : **wmake**
- in *solver/src* : **wmake**

3.3 creating the run case directory *CV2-run-All*

in *preprocess* : **./gen-runcase**

3.4 runnig the case

in *CV2-run-All* : **./Allrun**

3.5 post-treatment

in *postprocess* : **./plot.plt**
This gnuplot script plot the averaged pressure versus time obtained from the simulation and from experimental data. The X server must be set correctly.

4 solver

4.1 brief description

The CV2 solver is based on the sonicFoam solver where tree scalars have been added:

- *scal1* is the reactive scalar (equivalent to the fuel mass fraction)
- *scal2* is the fuel-inlet tracer (equivalent to the mixture fraction or mass fraction of fuel conditioned in fresh gas)
- *scal3* is the initial-condition tracer. It corresponds to the residual burned gas mass fraction.

This last scalar is required for cycles that follows a reactive cycle. In these cycles the domain is initially filled with the residual burned gases of the previous cycle that may affect combustion. However, the initial spatial distribution of concentration is neglected: the residual burned gases are considered homogeneous in the initial whole domain. These three scalars must be reset at the end of each reactive cycle.

The transport equations for *scal2* and *scal3* (non-reactive scalar) : *scal2Eqn.H*, *scal3Eqn.H* do not require additional source terms. However, in *scal1Eqn.H* two additional source terms are used:

- the chemical source term representative of the flame propagation mechanism.
- the chemical source term representative of the spark ignition.

These two terms are also included in the energy equation (sensible internal energy, see ref[3]).

4.2 solver modification

To correct or improve the solver :

- copy the solver-version directory in a new one: *CV2Foam+date*
- Change the name of the source c++ file that is in the main directory: *CV2Foam+date.C*
- Change the name of the source c++ file in the configuration make file (*Make/files*) : *CV2Foam+date.C*
- Change the name of the executable file in the configuration make file : *CV2Foam+date*
- Use the command **wmake** in the solver-version directory. This command must be used after each modification of the solver.

5 pre-process

Not ready yet : to develop

6 Run case

6.1 Main running script description

The running script *Allrun.pbs* (for Thor cluster) execute the following commands:

- **topoSet** (to defined the location of the spark ignition in the mesh from */system/topoSetDict*)
- **changeDictionary** (to defined the boundary conditions of the phase from */system/changeDictionaryDict*)
- **decomposePar** (to decompose the domain from */system/decomposeParDict*)) *CV2Foam+date.C*
- `mpiexec_mpt -np $NCPU $OF_APP -parallel > logfile 2>&1` (to run in parallel)
- **reconstructPar -latestTime** (to reconstruct the latest time required for the initial condition of the next phase)
- **mkdir**... (to store the results in a specific directory)

Then, the script prepare the files to run the next phase by copying the boundary conditions, the parameter file from the *preprocess* directory. The new simulation time is also directly modified in the /system/controlDict file with the following command:

```
sed -i "s/${STARTTIME}/${ENDTIME}/" $FILE
```

All these command lines (except *toposet*) are used for each simulated phase.

6.2 the *constant* folder

Not ready yet : to develop

6.3 the *system* folder

Not ready yet : to develop

6.4 the *0* folder

Not ready yet : to develop

7 post-process

Not ready yet : to develop

8 Changing the mesh of the simulation

8.1 create the new mesh with GMSH

In the preprocess/mesh directory change the refine-paremeters 'a' 'b' 'x' 'y' of the file "CV2-mesh-ori.geo". Then, create the mesh using : ./gmsh -3 CV2-mesh-ori.geo -o CV2-mesh-ori.msh

8.2 folders preparation

The running case directory corresponding to the desired cycle obtained with the original mesh is required to set initial conditions for the new mesh.

- prepare the running case with the original mesh.
- prepare a running case for the new mesh with "constant" and "system" folders and "Allrun.*" files of the original mesh case.
- Suppress constant/polyMesh directory of this new case.

8.3 Mesh conversion

- copy the new mesh in the new case directory
- convert to foam mesh using : gmshToFoam newMesh.msh
- save the original changeDictionaryDict file : mv system/changeDictionaryDict system/changeDictionaryDict-ori
- copy the changeDictionaryDict-newmesh file from preprocess directory : cp ../preprocess/mesh/changeDictionaryDict-newmesh system/changeDictionaryDict
- change the file /constant/polyMesh/boundary executing the command : changeDictionary

8.4 Create initial condition

- create an empty initial condition folder, for example : mkdir 0.28
- In /system/controlDict set the corresponding startTime at 0.28
- Generate the initial conditions from those of the orignial mesh case : mapFields ../original-mesh-case/ -consistent
- set the ignition zones : topoSet

- Put the right boundary conditions : mv system/changeDictionaryDict-ori system/changeDictionaryDict
- Execute the command to put the appropriate boundary conditions : changeDictionary

You can run the case

9 References

Not ready yet : to develop
- [1] Thèse Quentin
- [2] Article Quentin
- [3] Article Num