myEngineFoam – Implementation of different combustion model and newJanafThermo model

Lorenzo Bartolucci

Department of Industrial Engineering
University of Rome – Tor Vergata
Italy

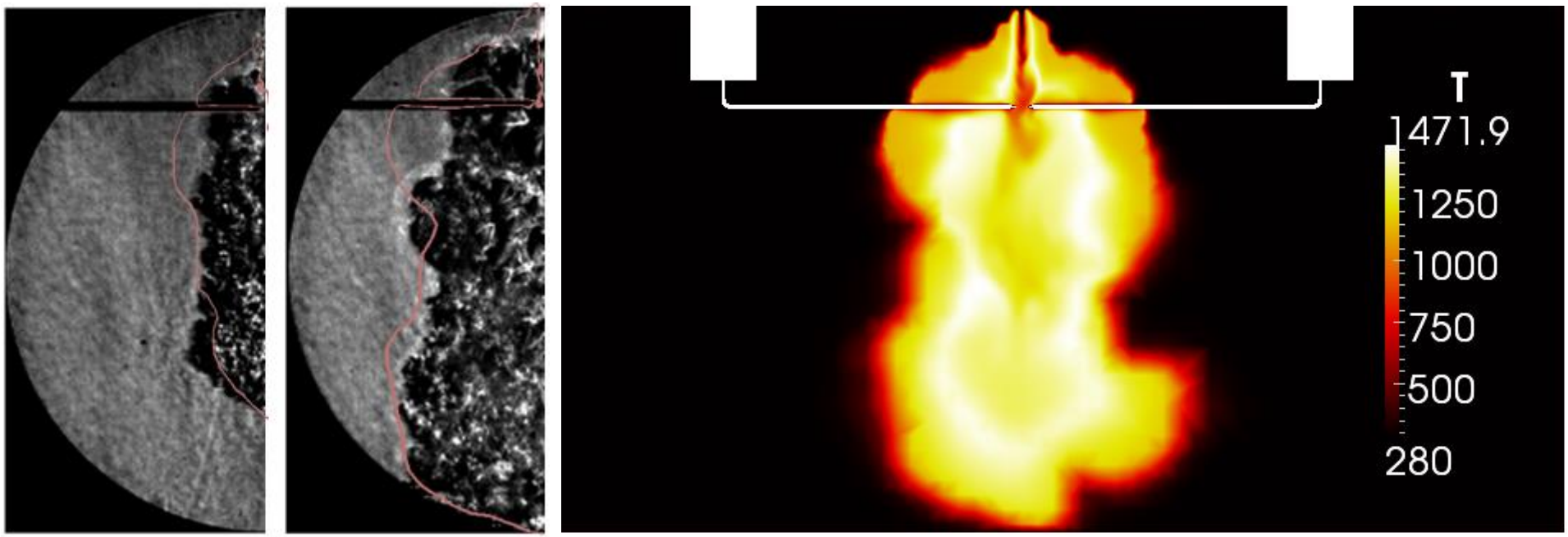
28/11/2014

Outline:

- implementation of PaSR combustion model in engineFoam;
- set a tutorial with the different combustion model;
- implementation of the newJanafThermo model;

Implementation of PaSR combustion model

- Why?



Copy the existing solvers

Copy the engineFoam solver folder into your personal application directory:

```
mkdir   $WM_PROJECT_USER_DIR/applications/solvers/combustion/myEngineFoam   –p

cp   –r   $FOAM_APP/solvers/combustion/engineFoam/*
$WM_PROJECT_USER_DIR/applications/solvers/combustion/myEngineFoam

cd   $WM_PROJECT_USER_DIR/applications/solvers/combustion/myEngineFoam
```

Copy the Yeqn.H, Eeqn.H and createFields.H from reactingFoam solver

```
cp   –r   $FOAM_APP/solvers/combustion/reactingFoam/YEqn.H   .
cp   –r   $FOAM_APP/solvers/combustion/reactingFoam/EEqn.H   .
cp   –r   $FOAM_APP/solvers/combustion/reactingFoam/createFields.H   .
```

Modify the solver

The engineFoam.C file has to be modified:

```
mv   engineFoam.C   myEngineFoam.C
vi   myEngineFoam.C
```

remove the following includes:

```
#include " psiuReactionThermo.H "
#include " laminarFlameSpeed.H"
#include " ignition.H"
#include " readCombustionProperties.H "
#include " ftEqn.H"
#include " bEqn.H"
#include " EauEqn.H"
#include " EaEqn.H"
#include " EauEqn.H"
```

and add the following:

```
#include " psiCombustionModel.H"
#include " multivariateScheme.H"
#include " readGravitationalAcceleration.H"
#include " YEqn.H"
#include " EEqn.H"
```

Modify the solver

Now changes in the "options" and “files" have to be implemented. In the options file the following lines have to be added:

-I$(LIB_SRC)/thermophysicalModels/specie/lnInclude \

-I$(LIB_SRC)/thermophysicalModels/basic/lnInclude \

-I$(LIB_SRC)/thermophysicalModels/reactionThermo/lnInclude \

-I$(LIB_SRC)/thermophysicalModels/chemistryModel/lnInclude \

-I$(LIB_SRC)/combustionModels/lnInclude \

in EXE INC, and:

-lcombustionModels \

-lspecie \

-lfluidThermophysicalModels \

-lreactionThermophysicalModels \

-lchemistryModel

in the EXE LIBS.

Modify the solver

Change also in Make/files

engineFoam.C into myEngineFoam.C

and

EXE=$(FOAM_APPBIN)/engineFoam into
EXE=$(FOAM_USER_APPBIN)/myEngineFoam

Now it is possible to compile the solver typing wmake**.**

Modify the tutorial case

Now, the tutorial case for engineFoam has to be modified, in order to make it able to use the new combustion model. The first step is to copy the existing tutorial kivaTest in a "run" directory.

```
mkdir $WM_PROJECT_USER_DIR/run/OS_CFD_2014/tutorials/combustion -p
cd $WM_PROJECT_USER_DIR/run/OS_CFD_2014/tutorials/combustion
cp -r $FOAM_TUTORIALS/combustion/engineFoam/kivaTest .
mv kivaTest myEngineFoamTest
cd myEngineFoamTest
```

Type kivaToFoam **in order to convert the otape17 file in a OpenFOAM type mesh.**

After that we have to start to modify the constant folder.

Modify the tutorial case

First of all we change the turbulence treatment, using a LES approach with OneEquationEddy model; in order to do this we have to modify the turbulenceProperties and the RASProperties files with the follow steps:

```
sed  -i  s/RASModel/LESModel/g  constant/turbulenceProperties
mv  constant/RASProperties  constant/LESProperties
vi  LESProperties
```

change the object of the FoamFile from RASProperties to LESProperties, the RASModel into LESModel and the kEpsilon into oneEqEddy. Addition of the follow lines is also need:

```
delta vanDriest;
vanDriestCoeffs {
    delta cubeRootVol;
    cubeRootVolCoeffs {
        deltaCoeff 1;
    }
    Aplus 26;
    Cdelta 0.158;
}
```

Modify the tutorial case

Now we have to copy all files needed to set properly the combustion characteristic and the thermodynamic species from the reactingFoam tutorial (that yet uses the PaSR combustion model).

```
cd constant/
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/chemistryProperties .
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/combustionProperties .
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/reactions .
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/thermo.compressibleGas .
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/thermophysicalProperties .
cd ..
```

Modify the tutorial case

We have to modify the boundary and initial conditions now.

mv ./-180/alphat ./-180/alphaSgs

vi ./-180/alphaSgs

change al alphatWallFunction boundary condition in zeroGradient boundary condition.

mv ./-180/mut ./-180/muSgs

vi ./-180/muSgs

change al mutWallFunction boundary condition in zeroGradient boundary condition.

Modify the tutorial case

```
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/CH4
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/O2
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/N2
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/Ydefault
sed -i s/fuel/piston/g ./-180/CH4 ./-180/O2 ./-180/N2 ./-180/Ydefault
sed -i s/air/liner/g ./-180/CH4 ./-180/O2 ./-180/N2 ./-180/Ydefault
sed -i s/outlet/cylinderHead/g ./-180/CH4 ./-180/O2 ./-180/N2 ./-180/Ydefault
sed -i s/fixedValue/zeroGradient/g ./-180/CH4 ./-180/O2 ./-180/N2 ./-180/Ydefault
vi ./-180/CH4 ./-180/O2 ./-180/N2 ./-180/Ydefault
rm ./-180/b ./-180/epsilon ./-180/ft ./-180/fu ./-180/Su ./-180/Tu
```

Modify the tutorial case

```
sed -i s/"(phi,epsilon)"/"(phi,Yi_h)"/g system/fvSchemes
sed -i s/Xi/Yi/g system/fvSolution
```

Delete all variables and divergences are not needed for the running of the new solver

Check quickly the settings of the case and the ./Allrun file

```
vi system/controlDict.1st system/controlDict.2nd system/controlDict.3rd
vi ./Allrun
```

Implementation of the newJanafThermo model

- **why?**

For some application the temperature range up to 6000 K is not sufficient. In an application containing plasma (e.g. arc welding) the temperature range could be up to 30000 K. Therefore the newer janaf model is needed which handles a larger temperature range up to 30000 K. The temperature dependent values of Cp, H and S are calculated with 3 polynomials using 9 coefficients.

Cp(T) = R(a[1]/T^2 + a[2]/T+ a[3] + a[4]T + a[5]T^2 + a[6]T^3 + a[7]T^4)
H(T) = R(-a[1]/T + a[2]lnT + a[3]T + (a[4]T^2)/2 + (a[5]T^3)/3 + (a[6]T^4)/4 + (a[7]T^5)/5 + b[1])
S(T) = R(-a[1]/(2*T^2) - a[2]/T + a[3] lnT + a[4]T + (a[5]T^2)/2 + (a[6]T^3)/3 + (a[7]T^4)/4 + b[2])

You can download an example for the new janaf table here:
http://www.galcit.caltech.edu/EDL/public/thermo/thermo.inp

For temperatures above 10000 K the Sutherland formula is not a good approximation anymore. The beginning ionization leads to an decreasing of the laminar viscosity. Therefor a transport model based on polynomials should be used instead

Getting start

In order to implement the new Janaf Thermo method into OpenFOAM, several modifications have to be done.

First of all we have to copy the thermophysicalModels and combustionModels folders from the $FOAM_SRC to the $WM_PROJECT_USER_DIR/src to be sure that our modifications cannot compromise the released libraries.

```
cp -r $FOAM_SRC/thermophysicalModels/ $WM_PROJECT_USER_DIR/src/
cp -r $FOAM_SRC/combustionModels/ $WM_PROJECT_USER_DIR/src/
```

Now we have to modify all files that include the janafThermo model and to define the newJanafThermo model.

To make this issue easier, we can find where the current janafThermo model is present and included.

```
find $WM_PROJECT_USER_DIR/src/ -name janaf*
grep -r janaf $WM_PROJECT_USER_DIR/src/
```

newJanafThermo

As it is possible to see the janafThermo model is included in several files, in particular in the thermophysicalModels and combustionModels folder, and we now try to explain the reasons for each include, but before to do this, the newJanafModel has to be defined.

```
cd $WM_PROJECT_USER_DIR/src/thermophysicalModels/specie/thermo/
cp -r janaf/ newJanaf/
mv newJanaf/janafThermo.C newJanaf/newJanafThermo.C
mv newJanaf/janafThermo.H newJanaf/newJanafThermo.H
mv newJanaf/janafThermoI.H newJanaf/newJanafThermoI.H
sed -i s/janaf/newJanaf/g newJanaf/newJanaf*
sed -i s/"nCoeffs_ = 7"/"nCoeffs_ = 9"/g newJanafThermo.H
vi newJanafThermoI.H
```

In this file we have to modify the expression of cp, h and s.

Now we have to modify all files that include the janafThermo model, in order to highlight which files have to be modify in the detail, we repeat the grep command for all folder present in the two main folders thermophysicalModels and combustionModels.

newJanafThermo

cd $WM_PROJECT_USER_DIR/src/thermophysicalModels

grep –r janaf specie

vi specie/include/thermophysicsTypes.H

#include " newJanafThermo.H "

Create a duplicate of all thermophysicsTypes which includes janafThermo model and rename these.

vi specie/include/reactionTypes.H

vi specie/reaction/reactions/makeReaction.H

vi specie/reaction/reactions/makeReactions.C

grep –r janaf radiatonModels

vi radiationModels/submodels/sootModel/mixtureFractionSoot/mixtureFractionSoots.C

#include " thermophysicsTypes.H " ← HERE!!

gasHThermoPhysics → gasHNewThermoPhysics ...

newJanafThermo

grep –r janaf basic

vi basic/rhoThermo/rhoThermos.C

vi basic/psiThermo/psiThermos.C

#include " newJanafThermo.H "

Create a duplicate of all Thermo which includes janafThermo model and rename these.

grep –r janaf reactionThermo

vi reactionThermo/chemistryReaders/chemistryReader/makeChemistryReaders.C

vi reactionThermo/chemistryReaders/chemkinReader/makeChemkinReader.C

vi reactionThermo/psiReactionThermo/psiReactionThermos.C

vi reactionThermo/psiuReactionThermo/psiuReactionThermos.C

vi reactionThermo/rhoReactionThermo/rhoReactionThermos.C

#include " newJanafThermo.H "

Create a duplicate of all constructors which includes janafThermo model and rename these.

newJanafThermo

```
grep –r janaf chemistryModel
vi chemistryModel/chemistrySolver/chemistrySolver/makeChemistrySolvers.C
vi chemistryModel/chemistryModel/rhoChemistryModel/rhoChemistryModels.C
vi chemistryModel/chemistryModel/psiChemistryModel/psiChemistryModels.C

grep –r janaf solidChemistryModel
vi solidChemistryModel/solidChemistrySolver/makeSolidChemistrySolvers.C
vi solidChemistryModel/basicSolidChemistryModel/basicSolidChemistryModels.C

grep –r janaf radiationModels
vi radiationModels/submodels/sootModel/mixtureFractionSoot/mixtureFractionSoot.C
```

In all other folders janafThermo model is not included.

Now we have to modify all Make/files and Make/options files in order to make unique the name of the libraries and different from the released onces. After that we have not yet finish. The combustionModels include the janafThermo model as well, so we have to modify also that folder.

newJanafThermo

```
cd $WM_PROJECT_USER_DIR/src/combustionModels

grep –r janaf .
vi FSD/FSDs.C
vi infinitelyFastChemistry/infinitelyFastChemistrys.C
vi diffusion/diffusions.C
```

After changing Make/files and Make/options for combustionModels folder we are ready to compile the libraries with the newJanafThermo model implemented.

myEngineNewJanafFoam

In order to use the new libraries we have to add the sourcing of these in the our tutorial controlDict file.

Add the following line in all controlDict files:

```
libs (
        " libmyCombustionModels.so "
        " libmySpecie.so "
        " libmyFluidThermophysicalModels.so "
        " libmyReactionThermophysicalModels.so "
        " libmyChemistryModel.so "
    );
```

myEngineNewJanafFoam

Since it will be very long in time, to test the correct implementation we can put some dummy in the thermophysicalProperties instead of janaf:

```
thermoType
{
        type                hePsiThermo;
        mixture             reactingMixture;
        transport           sutherland;
        thermo              asdasdasd;
        energy              sensibleEnthalpy;
        equationOfState     perfectGas;
        specie              specie;
}
```

In the output we can see the newJanafThermo model amoung the possibilities.

Thanks for your attention

Remember that the science is called such because it tries, it looks ...
some times with success ...
but to get to that point ...
many attempts have to be made without ever get discouraged!
G.M.