CFD with OpenSource software

A course at Chalmers University of Technology
Taught by Håkan Nilsson

Project work:

EngineFoam: implementation of a different combustion model and the new Janaf thermo equations

---

Developed for OpenFOAM-2.3.x

Author:
Lorenzo Bartolucci

Peer reviewed by:
Naser Hamedi
Håkan Nilsson

December 9, 2014

Contents

Chapter 1

Introduction

The aim of this tutorial is to show how to implement a new thermodynamic model, a different combustion model in a existing solver and how to run a case involving combustion in a engine dynamic mesh.

In this report the implementation of a different combustion model, in particular the PaSR combustion model, in the engineFoam solver will be shown. It is shown how the new solver can be applied to a tutorial case. Finally, it is described how to implement a new JanfThermo model. But first of all, a brief description of the PaSR combustion model is due. The combustion model used in this work is the *Partially Stirred Reactor Combustion* model, which uses complex chemistry to model ignition and mixed-controlled combustion. For these reasons, it requires high computational time and is suitable for diagnostic purposes. However, it provides very good insight of the combustion process taking place during the ignition and combustion phases, that is what we need.

Chomiak, Golovichev et al. [1], [2] applied detailed kinetics and the Partially Stirred Reactor concept (PaSR) to correctly describe the turbulence/chemistry interaction in Diesel spray combustion.

In the PaSR approach, a computation cell is split into two different zones: in one zone all reactions occur, while in the other one there are no reactions. Thus, the composition changes due to mass exchange with the reacting zone. Furthermore, the reacting zone is treated as a Perfectly Stirred Reactor (PSR), in which the composition is homogeneous. According to the PaSR concept, the concentration of each species i at the exit of reactor can be defined as:

$$c_1^i = k^* c^i + (1 - k^*) C_0^i \tag{1.1}$$

where k^* is the mass fraction of the mixture which reacts.

As shown in Figure 1.1, the model distinguishes between three molar concentrations:

- c_0^i is the averaged concentration in the feed of the cell and may be considered as the initial averaged concentration in the cell.
- c^i is the unknown concentration in the reaction zone on a sub-grid level in the unknown reactive fraction of the cell material.
- c_1^i is the sought for, time-averaged exit concentration. This is also the averaged concentration in the cell.

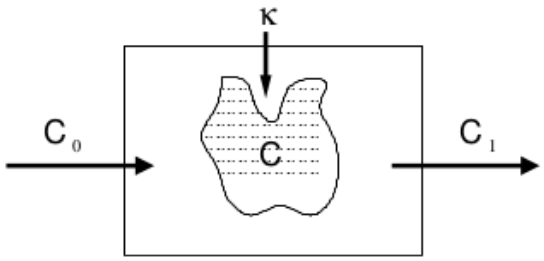

Figure 1.1. Conceptual picture of the Partially Stirred Reactor

According to equation 1.1, c_1^i is a linear interpolation between c^i and c_0^i and the whole combustion process can be split in sub-steps, proceeding in parallel (see Figure 1.2):

I) The initial concentration in the reaction zone changes from c_0^i to c^i;

II) The reactive mixture c^i is mixed by turbulence with c_0^i resulting in the averaged concentration c_1^i.

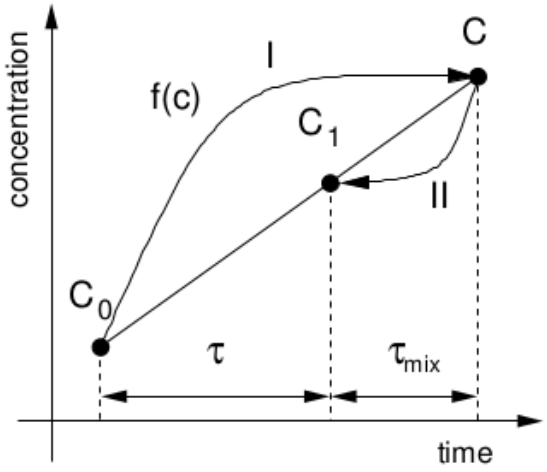

Figure 1.2. The reaction/mixing step procedure

Since c_1^i is the initial value for the next time-step, the time between c_0^i and c_1^i must be the integration step, τ. The turbulence mixes c^i with c_0^i, hence the time difference between c^i and c_1^i must be the characteristic time for turbulence, τ_{mix}. Assuming that the slope of the curve in Figure 1.2 is equal to the reaction rate in the reaction zone, it results in

$$\frac{c_1^i - c_0^i}{\tau} = \frac{c^i - c_1^i}{\tau_{mix}} = f(c^i); \quad k^* = \frac{\tau}{\tau + \tau_{mix}}. \tag{1.2}$$

Here, $f(c^i)$ is reaction rate of the species i during time-step τ which is modelled with an Arrhenius' equation. To obtain c_1^i, it is now necessary to eliminate c^i in equation 1.1. Using Taylor expansion, the term $f(c^i)$ can be expressed as

$$f(c^i) = f(c_1^i) + \frac{\partial f}{\partial c}\,|_{c=c_1^i}\,(c^i - c_1^i). \tag{1.3}$$

The term $\frac{\partial f}{\partial c}$ is assumed to be the reciprocal of a chemical time scale:

$$\frac{1}{\tau_c} = -\frac{\partial f}{\partial c} \tag{1.4}$$

and is calculated as

$$\frac{\partial f}{\partial c}\,|_{c=c_1^i} = \frac{\dot{\omega}(c_1^i) - \dot{\omega}(c_0^i)}{c_1^i - c_0^i}, \tag{1.5}$$

where $\dot{\omega}(c_1^i)$ and $\dot{\omega}(c_0^i)$ are the reaction rate expressions of species i calculated at the beginning and at the end of the time step. Thus, equation 1.3 becomes

$$f(c^i) = f(c_1^i) - \frac{c^i - c_1^i}{\tau_c}. \tag{1.6}$$

Substituting the expression of τ_c in equation 1.1 the following expression is finally obtained for the sub-grid reaction rate:

$$\frac{c_1^i - c_0^i}{\tau} = \frac{\tau_c}{\tau_c + \tau_{mix}} f_m(c_1^i). \tag{1.7}$$

The reactive fraction k_i becomes equal to

$$k_i = \frac{\tau_c}{\tau_c + \tau_{mix}}. \tag{1.8}$$

Several expressions were proposed for the mixing time τ_{mix} [1], [2]. In this work, it is assumed to be

$$\tau_{mix} = C_{mix}\sqrt{\frac{\mu_{eff}}{\rho\varepsilon}}, \tag{1.9}$$

where C_{mix} needs to be estimated a priori. In order to do this, another definition of τ_{mix} ([5]), based on turbulence scales, has to be used

$$\tau_{mix} = \sqrt{\frac{k}{\varepsilon}(\frac{\nu}{\varepsilon})^{1/2}}. \tag{1.10}$$

Using also the following definitions and assumptions

$$\frac{\mu_t}{\rho} = C_\mu \frac{k}{\varepsilon}; \quad Sc_t = 1; \quad Re_t = \frac{k^2}{\varepsilon\nu}, \tag{1.11}$$

we obtain

$$C_{mix} = \sqrt{\frac{1}{1 + C_\mu Re_t}}. \tag{1.12}$$

So, for:

- ▷ **Laminar Flow** $Re_t = 0 \Longrightarrow C_{mix} = 1$
- ▷ **Typical turbulent flow** $Re_t \simeq 1000 \Longrightarrow C_{mix} \simeq 0.1$
- ▷ **Extremely turbulent flow** $Re_t \to \infty \Longrightarrow C_{mix} \to 0$

Hence typical values for C_{mix} range from 0.001 to 0.3 [3].

Chapter 2

Implement different combustion model in engineFoam

This section shows the steps to follow in order to change the combustion model in the engineFoam solver. After that it will show how to set up a case for the new solver.

2.1 Create the new solver

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 .
```

The engineFoam.C file has to be modified, first of all, changing the name into myEngineFoam.C, and after changing some lines in the file. The includes "psiuReactionThermo.H","laminarFlameSpeed.H" and "ignition.H" have to be deleted since they were needed by the XiCombustionModel but not the PaSR model. The new includes "psiCombustionModel.H" and "multivariateScheme.H" have to be added in order to use the PaSR model [Figure 2.1].

Other changes that have to be done are: delete the "readCombustionProperties.H" include and add the "readGravitationalAcceleration.H" [Figure 2.2]; delete the includes regarding the XiModel equations as: "ftEqn.H", bEqn.H", "EauEqn.H" and "EaEqn.H", add the "YEqn.H" and "EEqn.H" includes and, finally, delete the lines about the ignitionModel, since it will instead be treated as a source of energy through the fvOptions file in the system folder. [Figure 2.3]. The last change is to delete the burned fraction (b) from the logSummary.H file [Figure 2.4]. Now changes in the "options" and "files" have to be implemented. Using the synopsis:

```
\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "engineTime.H"
#include "engineMesh.H"

// modified
//#include "psiuReactionThermo.H"
// with
#include "psiCombustionModel.H"

#include "turbulenceModel.H"

// eliminated
//#include "laminarFlameSpeed.H"
//#include "ignition.H"

#include "Switch.H"
#include "OFstream.H"
#include "mathematicalConstants.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"

// added
#include "multivariateScheme.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Figure 2.1. First changes in myEngineFoam.C file

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    #include "setRootCase.H"

    #include "createEngineTime.H"
    #include "createEngineMesh.H"

    //eliminated
    //#include "readCombustionProperties.H"

    #include "createFields.H"
    #include "createFvOptions.H"
    #include "createRhoUf.H"
    #include "initContinuityErrs.H"
    #include "readEngineTimeControls.H"
    #include "compressibleCourantNo.H"
    #include "setInitialDeltaT.H"
    #include "startSummary.H"

    // added
    #include "readGravitationalAcceleration.H"

    pimpleControl pimple(mesh);

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Figure 2.2. Second changes in myEngineFoam.C file

`git diff [options] [<commit>] [--] [<path>...]`

the differences about the corresponding files are highlighted in the next figures [2.5]-[2.6].

Now it is possible to compile the solver typing wmake.

```
\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "engineTime.H"
#include "engineMesh.H"

// modified
//#include "psiuReactionThermo.H"
// with
#include "psiCombustionModel.H"

#include "turbulenceModel.H"

// eliminated
//#include "laminarFlameSpeed.H"
//#include "ignition.H"

#include "Switch.H"
#include "OFstream.H"
#include "mathematicalConstants.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"

// added
#include "multivariateScheme.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Figure 2.3. Last changes in myEngineFoam.C file

```
Info<< "Mean pressure:" << p.weightedAverage(mesh.V()).value() << endl;
Info<< "Mean temperature:" << thermo.T().weightedAverage(mesh.V()).value()
    << endl;
Info<< "Mean u':"
    << (sqrt((2.0/3.0)*turbulence->k()))().weightedAverage(mesh.V()).value()
    << endl;

logSummaryFile
    << runTime.theta() << tab
    << p.weightedAverage(mesh.V()).value() << tab
    << thermo.T().weightedAverage(mesh.V()).value() << tab
    << (sqrt((2.0/3.0)*turbulence->k()))().weightedAverage(mesh.V()).value()
    << tab
    // deleted
    // << 1 - b.weightedAverage(mesh.V()).value()
    << endl;
```

Figure 2.4. Changes in logSummary.H file

```
diff --git a/home/barto/OpenFOAM/OpenFOAM-2.3.x/applications/solvers/combustion/engineFoam/Make/options
b/OpenFOAM/barto-2.3.x/applications/solvers/combustion/myEngineFoam/Make/options
index fcbc3a4..cadf3a0 100644
--- a/home/barto/OpenFOAM/OpenFOAM-2.3.x/applications/solvers/combustion/engineFoam/Make/options
+++ b/OpenFOAM/barto-2.3.x/applications/solvers/combustion/myEngineFoam/Make/options
@@ -1,5 +1,4 @@
 EXE_INC = \
-    -I$(FOAM_SOLVERS)/combustion/XiFoam \
     -I$(LIB_SRC)/finiteVolume/lnInclude \
     -I$(LIB_SRC)/fvOptions/lnInclude \
     -I$(LIB_SRC)/sampling/lnInclude \
@@ -9,7 +8,9 @@ EXE_INC = \
     -I$(LIB_SRC)/thermophysicalModels/basic/lnInclude \
     -I$(LIB_SRC)/thermophysicalModels/reactionThermo/lnInclude \
     -I$(LIB_SRC)/turbulenceModels/compressible/turbulenceModel \
-    -I$(LIB_SRC)/thermophysicalModels/laminarFlameSpeed/lnInclude
+    -I$(LIB_SRC)/thermophysicalModels/chemistryModel/lnInclude\
+    -I$(LIB_SRC)/ODE/lnInclude \
+    -I$(LIB_SRC)/combustionModels/lnInclude

 EXE_LIBS = \
     -lfiniteVolume \
@@ -20,7 +21,9 @@ EXE_LIBS = \
     -lcompressibleRASModels \
     -lcompressibleLESModels \
     -lspecie \
+    -lODE \
+    -lcombustionModels \
     -lfluidThermophysicalModels \
     -lreactionThermophysicalModels \
-    -llaminarFlameSpeedModels
+    -lchemistryModel
```

Figure 2.5. Differences between engineFoam/Make/options and myEngineFoam/-Make/options files

```
diff --git a/home/barto/OpenFOAM/OpenFOAM-2.3.x/applications/solvers/combustion/engineFoam/Make/files b/
OpenFOAM/barto-2.3.x/applications/solvers/combustion/myEngineFoam/Make/files
index 131a3be..2db00f5 100644
--- a/home/barto/OpenFOAM/OpenFOAM-2.3.x/applications/solvers/combustion/engineFoam/Make/files
+++ b/OpenFOAM/barto-2.3.x/applications/solvers/combustion/myEngineFoam/Make/files
@@ -1,3 +1,3 @@
-engineFoam.C
+myEngineFoam.C

-EXE = $(FOAM_APPBIN)/engineFoam
+EXE = $(FOAM_USER_APPBIN)/myEngineFoam
```

Figure 2.6. Differences between engineFoam/Make/files and myEngineFoam/Make/-files files

2.2 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
```

After that we have to start to modify the constant folder. 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 RAS-Model 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;
}
```

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

```
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/chemistryPr
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/combustionP
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/reactions c
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/thermo.comp
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/constant/thermophysi
```

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.

```
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/CH4 ./-180/
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/O2 ./-180/
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/N2 ./-180/
cp $FOAM_TUTORIALS/combustion/reactingFoam/ras/counterFlowFlame2D/0/Ydefault ./-180/
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
rm ./-180/b ./-180/epsilon ./-180/ft ./-180/fu ./-180/Su ./-180/Tu ./-180/Xi
```

We have to change the value of the specie mass fraction in order to make it correct for a simulation.

```
sed -i s/engineFoam/myEngineFoam/g system/controlDict
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. The controlDict, fvOptions and Allrun files are attached in the follow in order to show how to set the running of the case.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvOptions;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

energySource1
{
    type            scalarSemiImplicitSource;
    active          false;
    timeStart       -180;
    duration        10;
    selectionMode   points;
    points
    (
        (0.03 0 0.091)
    );

    scalarSemiImplicitSourceCoeffs
    {
        volumeMode      absolute;
        injectionRateSuSp
        {
            h            (120 0);
        }
    }
}
```

Figure 2.7. The fvOptions file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict.1st;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     myEngineFoam;

startFrom       startTime;

startTime       -180;

stopAt          endTime;

endTime         -50;

deltaT          0.25;

writeControl    runTime;

writeInterval   5;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep  no;

maxCo           0.2;

maxDeltaT       1;
```

Figure 2.8. The controlDict.1st file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict.2nd;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     myEngineFoam;

startFrom       startTime;

startTime       -50;

stopAt          endTime;

endTime         -42;

deltaT          0.025;

writeControl    runTime;

writeInterval   1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;



timePrecision   6;

runTimeModifiable true;

adjustTimeStep  no;

maxCo           0.2;

maxDeltaT       1;
```

Figure 2.9. The controlDict.2nd file

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.3.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict.3rd;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     engineFoam;

startFrom       startTime;

startTime       -42;

stopAt          endTime;

endTime         60;

deltaT          0.025;

writeControl    runTime;

writeInterval   5;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;



timePrecision   6;

runTimeModifiable true;

adjustTimeStep  no;

maxCo           0.2;

maxDeltaT       1;
```

Figure 2.10. The controlDict.3rd file

```
#!/bin/sh
cd ${0%/*} || exit 1    # run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# Get application name
application=`getApplication`

runApplication kivaToFoam -file otape17

cp system/controlDict.1st system/controlDict
runApplication $application
mv log.$application log.$application.1

cp system/controlDict.2nd system/controlDict
sed -i s/false/true/g system/fvOptions
runApplication $application
mv log.$application log.$application.2

cp system/controlDict.3rd system/controlDict
sed -i s/true/false/g system/fvOptions
runApplication $application
mv log.$application log.$application.3
# ------------------------------------------------------------------------ end-of-file
```

Figure 2.11. The Allrun file

Chapter 3

Implementation of new Janaf Thermo method

In order to implement the new Janaf Thermo method into OpenFOAM, several modification 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 can not compromise the released libraries. But first to start with the implementation the reasons of the newJanafThermo model have to been briefly explained. 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.

$$\begin{cases} cp(T) = R(a[1]/T^2 + a[2]/T + a[3] + a[4]T + a[5]T^2 + a[6]T^3 + a[7]T^4) \\ H(T) = R(-a[1]/T + a[2]lnT + a[3]T + (a[4]T^2)/2 + (a[5]T^3)/3 + (a[6]T^4)/4 + (a[7]T^5)/5 + b[1]) \\ S(T) = R(-a[1]/(2T^2) - a[2]/T + a[3]lnT + a[4]T + (a[5]T^2)/2 + (a[6]T^3)/3 + (a[7]T^4)/4 + b[2]) \end{cases}$$

It's possible to 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. Therefore a transport model based on polynomials should be used instead.
Now let's go through the implementation.

```
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/
```

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.

```
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, etc

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.

```
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 so we can remove these from the thermophysicalModels folder.
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.

```
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. 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 "
);
```

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 among the possibilities.

Bibliography

[1] V. Golovichev, N.Nordin, R.Jarnicki and J.Chomiak, *3-D Diesel Spray Simulations Using a New Detailed CHemistry Turbulent Combustion Model*, SAE Paper, 2000-01-1891, 2000

[2] J.Husaka, N.Horie, Y.Daisho, V.Golovichev and S.Nakayama, *Numerical Simulation Accounting for the Finite-Rate Elementary Chemical Reactions for Computing Diesel Combustion Precess*, SAE Paper, 2005-24-051, 2005

[3] Nordin, *Complex Chemistry Modelling of Diesel Spray Combustion*, PhD Thesis, 2001

[4] OpenFOAM, www.openfoam.org

[5] J.Chomiak and A.Karlsson, *Flame Liftoff in Diesel Sprays*. Proceedings of Twenty-Sixth Combustion Symposium (International), pages pp.2557-2564, 1996