The implementation of a PDF combustion model based on the CCM chemistry speedup approach

Shijie Xu

Division of Fluid Mechanics,
Department of Energy Science,
Lund University,
Lund, Sweden.

November 27, 2018

The filtered governing equation of species

$$\frac{\partial \bar{\rho}\tilde{Y}_i}{\partial t} + \frac{\partial \bar{\rho}\tilde{Y}_i\tilde{u}_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\bar{\rho}D_i\frac{\partial \tilde{Y}_i}{\partial x_j}\right) + \frac{\partial}{\partial x_j}\left(\overline{-\rho Y_i'' u_j''}\right) + \bar{\omega}_i$$

Turbulent transport flux

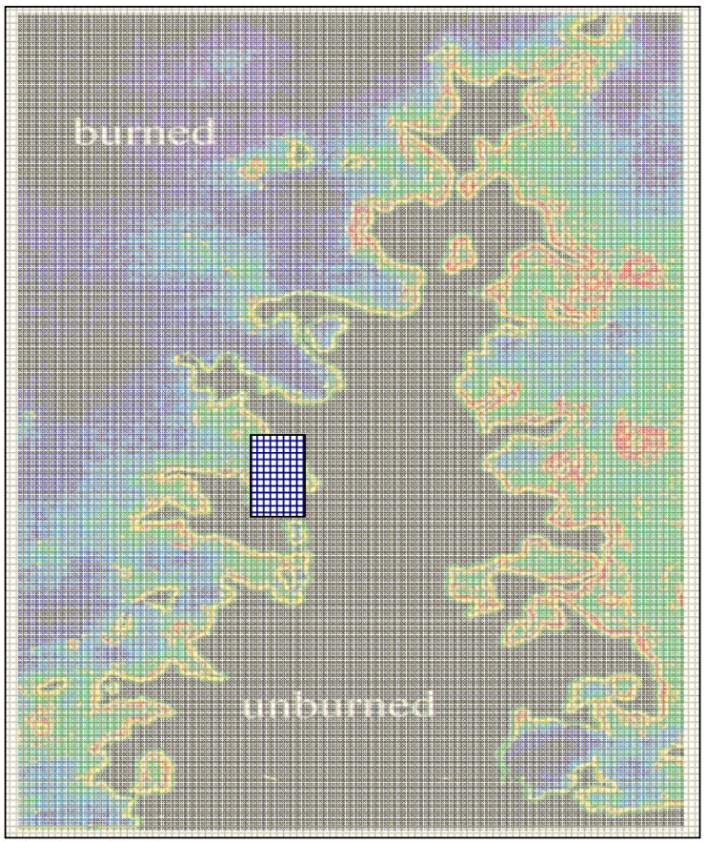
Turbulent reaction rate

Turbulence models
e.g. K-epsilon model

Combustion models

The challenge of turbulence combustion simulation in CFD

$$\overline{\omega\left(T\right)} \neq \omega\left(\overline{T}\right)$$



The concept of stochastic reactor (StoR) combustion model

How to separate a CFD cell into several states with different temperature?

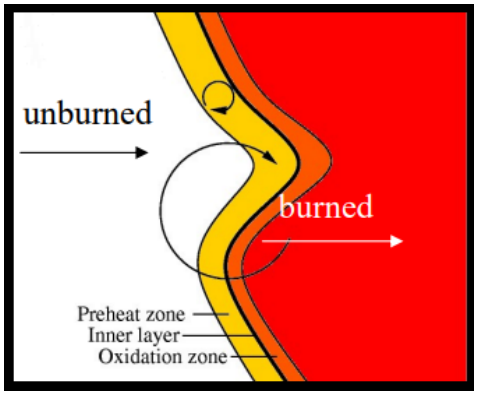


Fig. 1. The temperature states in a CFD cell.

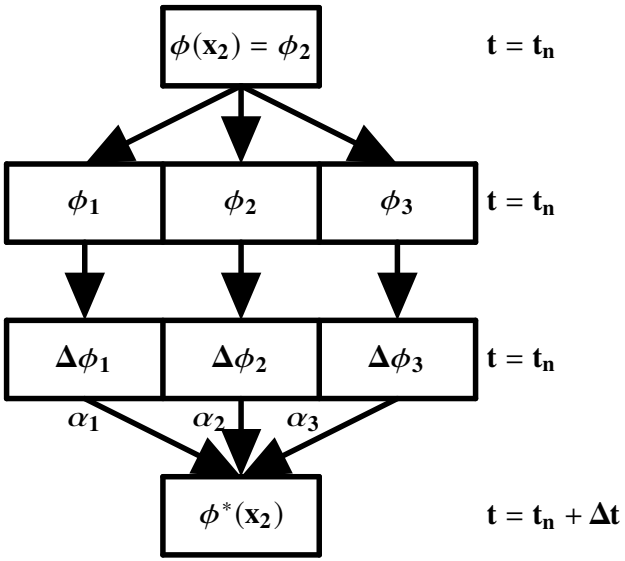


Fig. 2. The concept of stochastic reactor.

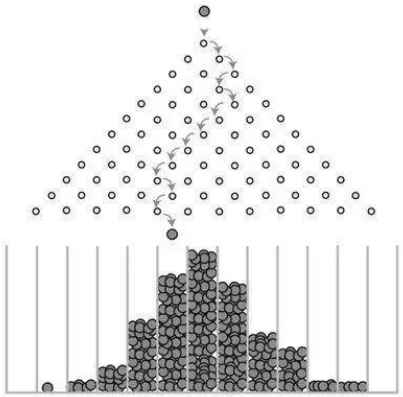


Fig. 3. The probability distribution function(PDF).

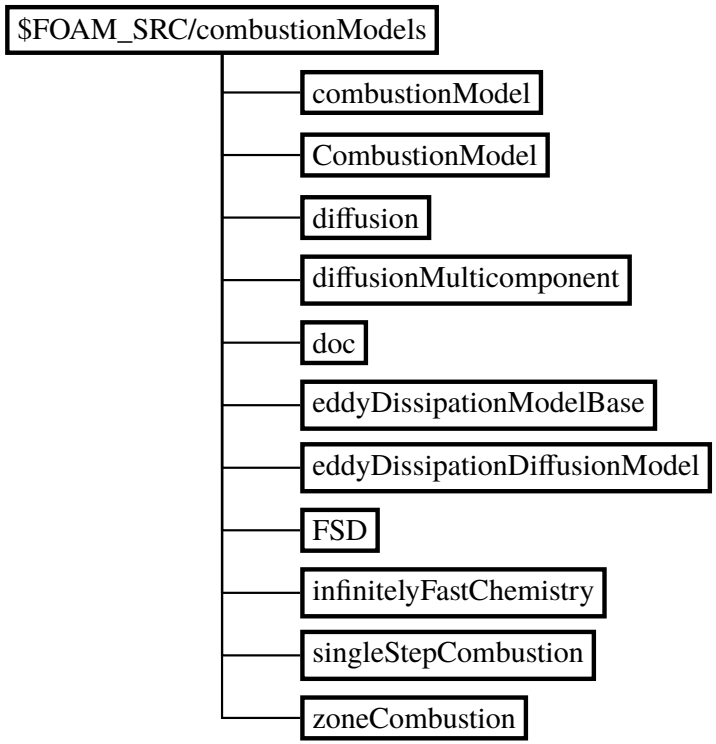
The calculation of weighting coefficient from PDF

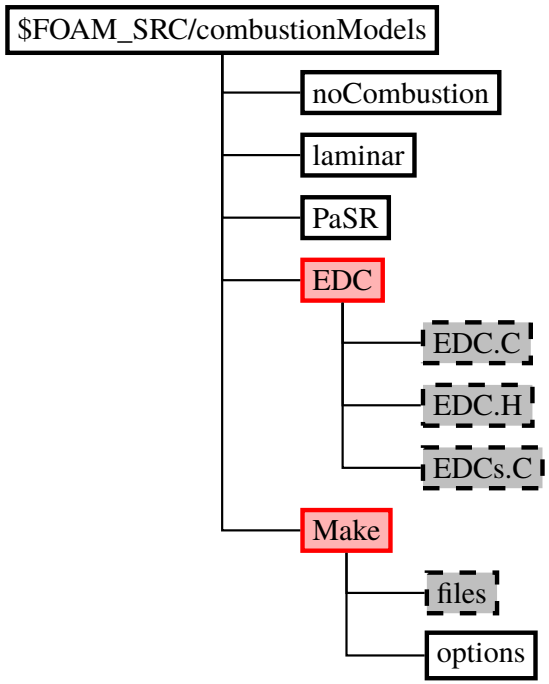
$$\alpha_i = \begin{cases} 0, & \phi_i \notin [\overline{\phi}_{l,t}, \overline{\phi}_{r,t}] \\ \frac{\mathbf{E}_\phi(\phi_1<\phi<\phi_2)-\phi_2\left[F_\phi(\phi_2)-F_\phi(\phi_1)\right]}{\phi_1-\phi_2}, & \phi_i = \overline{\phi}_{l,t} \\ \dfrac{\mathbf{E}_\phi(\phi_i < \phi < \phi_{i+1}) - \phi_{i+1}\left[F_\phi(\phi_{i+1}) - F_\phi(\phi_i)\right]}{\phi_i - \phi_{i+1}} & \\ \quad - \dfrac{\mathbf{E}_\phi(\phi_{i-1} < \phi < \phi_i) - \phi_{i-1}\left[F_\phi(\phi_i) - F_\phi(\phi_{i-1})\right]}{\phi_{i-1} - \phi_i}, & \phi_i \in (\overline{\phi}_{l,t}, \overline{\phi}_{r,t}) \\ -\frac{\mathbf{E}_\phi(\phi_{N-1}<\phi<\phi_N)-\phi_{N-1}\left[F_\phi(\phi_N)-F_\phi(\phi_{N-1})\right]}{\phi_{N-1}-\phi_N}, & \phi_i = \overline{\phi}_{r,t} \end{cases} \tag{1}$$

where $\mathbf{E}$ and $\mathbf{F}$ are the expectation and distribution function for PDF, $\overline{\phi}_{l,t}$ and $\overline{\phi}_{r,t}$ are the left and right truncation value for a specific CFD cell.

The structure of combustion model library

```
OFv1806
cd $WM_PROJECT_DIR/src/combustionModels
tree $WM_PROJECT_DIR/src/combustionModels
grep -r EDC
```





Compile the library in user dictionary without modification

Therefore, we copy the whole dictionary into user project folder $WM_PROJECT_USER_DIR without changing anything.

```
foam
cp -r --parents src/combustionModels $WM_PROJECT_USER_DIR
cd $WM_PROJECT_USER_DIR/src/combustionModels
```

Revise the the final 'libcombustionModels.so' location from $FOAM_LIBBIN to $FOAM_USER_LIBBIN in Make/files, and check whether the changes has been made by using 'tail' command. Compile it after changing.

```
sed -i s/FOAM_LIBBIN/FOAM_USER_LIBBIN/g Make/files
tail Make/files
wmake
```

Create a StoR model dictionary form EDC

It takes about 2 minutes to finish the compilation. When the compilation is finished, we can check the location of new compiled library by using ‘which’ and ‘grep’ command.

```
ldd ‘which sprayFoam‘ | grep libcombustionModels.so
```

Copy the EDC folder and rename it as StoR, rename the files in StoR dictionary.

```
cp -r EDC StoR
mv StoR/EDC.C StoR/StoR.C
mv StoR/EDC.H StoR/StoR.H
mv StoR/EDCs.C StoR/StoRs.C
sed -Ei 's/EDC/StoR/g' StoR/StoR*.*
```

By typing above command, the copy and revision of EDC/EDC.C, EDC/EDC.H and EDC/EDCs.C files are finished, we can look into the files to check by yourself.

Revise the Make/files to re-compile

Now we need to add a line 'StoR/StoRs.C' into Make/files, to let the compiler knows that a new added model 'StoR' need to be compiled.

```
vi Make/files
```

Update the lnInclude links and compile to get a new 'libcombustionModels.so' library with 'StoR' model.

```
wmakeLnInclude -u .
wmake
```

So far, a new combustion model named 'StoR' has been created, which includes the identical operation with 'EDC' model but different name. Next, we will create a case to run 'sprayFoam' solver with 'StoR' model to test it at distance.

The creation of a case 'aachenBombStoR' with StoR model

Before creating case folder, please check the current path by typing:

```
pwd
```

It should be still in '$WM_PROJECT_USER_DIR/src/combustionModels', since we will come back to revise combustion model, you can choose jump to case path or stay here to run the case in distance.

```
cp -r $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb $FOAM_RUN/aachenBombStoR
cp -r $FOAM_TUTORIALS/combustion/reactingFoam/RAS/SandiaD_LTS/constant/combustionProperties\
 $FOAM_RUN/aachenBombStoR/constant
```

Copy a spray combustion tutorial 'aachenBomb' to '$FOAM_RUN' dictionary, renamed it as 'aachenBombStoR'. And replace the constant/combustionProperties file from 'SandiaD_LTS' case, which employ a 'EDC' combustion model.

The creation of a case 'aachenBombStoR' with StoR model

Replace the keyword 'EDC' with 'StoR' in constant/combustionProperties file. You can check the revised combustionProperties files as well:

```
sed -Ei 's/EDC/StoR/g' $FOAM_RUN/aachenBombStoR/constant/combustionProperties
vi $FOAM_RUN/aachenBombStoR/constant/combustionProperties
```

Create mesh and run the case to see the printed information whether the StoR model works.

```
blockMesh -case $FOAM_RUN/aachenBombStoR
sprayFoam -case $FOAM_RUN/aachenBombStoR
```

If the printed information is like the following outputs and no error printed, it means that StoR model has been added into combustion model.

The creation of a case ‘aachenBombStoR’ with StoR model

```
/*******************   PRINTED INFORMATION   *******************/
...
Creating combustion model

Selecting combustion model StoR
Selecting chemistry solver
{
    solver          ode;
    method          standard;
}
...
```

The modification of StoR model

As mentioned before, there are only four files related with StoR model, where Make/files has been revised. Only three files need to be modified.

```
cp $FOAM_RUN/StoR/StoR* $WM_PROJECT_DIR/src/combustionModels/StoR
```

Let us start with StoR/StoRs.C file. open it by using vi command.This is a template file, define a Enum for EDC version, we do not need it any more, delete it, keep the headers included and namespace Foam dictionary.

```
vi StoR/StoRs.C
```

Then open StoR/StoR.H file to modify the declaration of StoR class and its members. In this file, I only modify the private data and functions.

```
vi StoR/StoR.H
```

The modification of StoR model

- The header #include "laplaceFilter.H" has been added for the calculation of temperature variance, the member variable spaceFilter_ is a laplaceFilter object; the theory of the variance calculation is in Eq.2; A volScalarField value Tsgs_ is created to store the variance of local temperature.

$$\sigma^2 = \widetilde{T'^2} \approx C\left(\widehat{\widetilde{T}^2} - \widehat{\widetilde{T}}^2\right) \tag{2}$$

The modification of StoR model

- The other private data are PDF_Name_, the spanZoneForPDF_, which is used to specific how many stochastic states exists for each cells. The truncationForPDF_ is used to bound the temperature fluctuations of stochastic states. The default value of truncationForPDF_ is unity, means that the temperature of stochastic states are limited in the region of $[\overline{T} - \sigma, \overline{T} + \sigma]$, where σ is the variance of the local temperature.
- scalarField P_i_ and T_i_ are the most important value which is used to specific the posibility and normalized temperature value for all the stochastic states. For example, we choose states $\overline{T} - \sigma$, $\overline{T}$ and $\overline{T} + \sigma$ as three temperature states, then it is posibility could be calculated according to Eq.1 and stored in P_i_.

The modification of StoR model

Recompile the combustion model with new revised StoR model, jump to the 'aachenBombStoR' case and run it again,

```
wmake
blockMesh -case $FOAM_RUN/aachenBombStoR
sprayFoam -case $FOAM_RUN/aachenBombStoR
more log.sprayFoam
```

but now you can see the print information for new developed StoR model, it is generated by construction function when a StoR class object is created.

```
/*********************   PRINTED INFORMATION   **********************/
...
Creating combustion model
Selecting combustion model StoR<psiChemistryCombustion>
...
Selecting ODE solver seulex
    using integrated reaction rate
StoR Combustion Model constructed:
presumed PDF name is NormalDistribution
and Truncations are 2(-1 1)
posibility coeff array are 3(0.24 0.52 0.24)
normalized variances are  3(-1 0 1)
...
```

Finite rate chemistry method

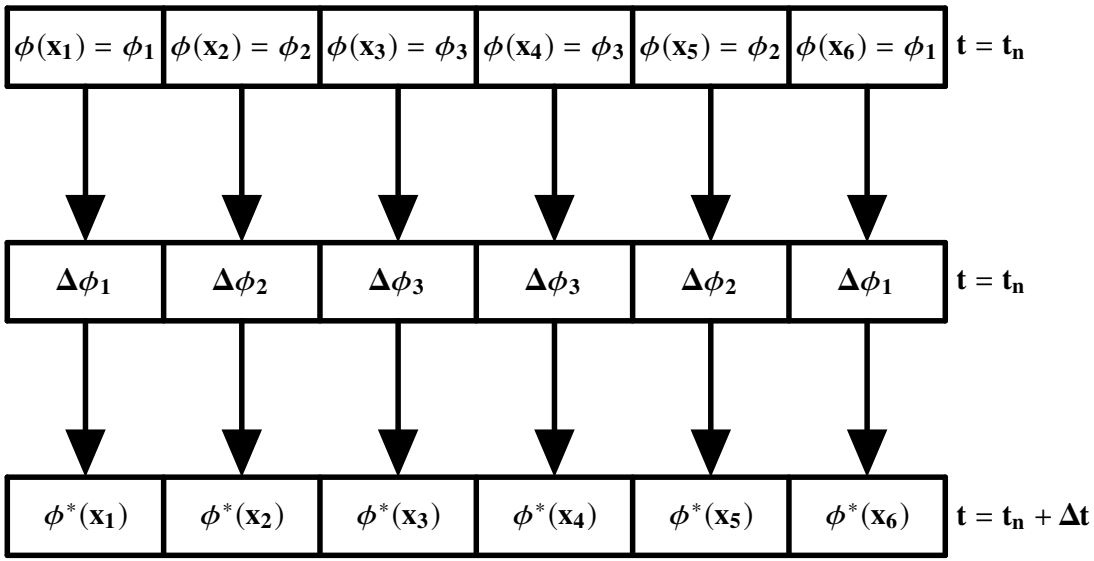


Fig. 4. The concept of finite rate chemistry method without chemistry speedup.

Fig. 4 shows the concept of chemistry fractional step in reaction flow simulation with finite rate chemistry method. As we can see in Fig. 4, the variable in different cells at time t_n have the same value. There is space for solver to improve the computational efficiency.

A chemistry speedup approach–CCM

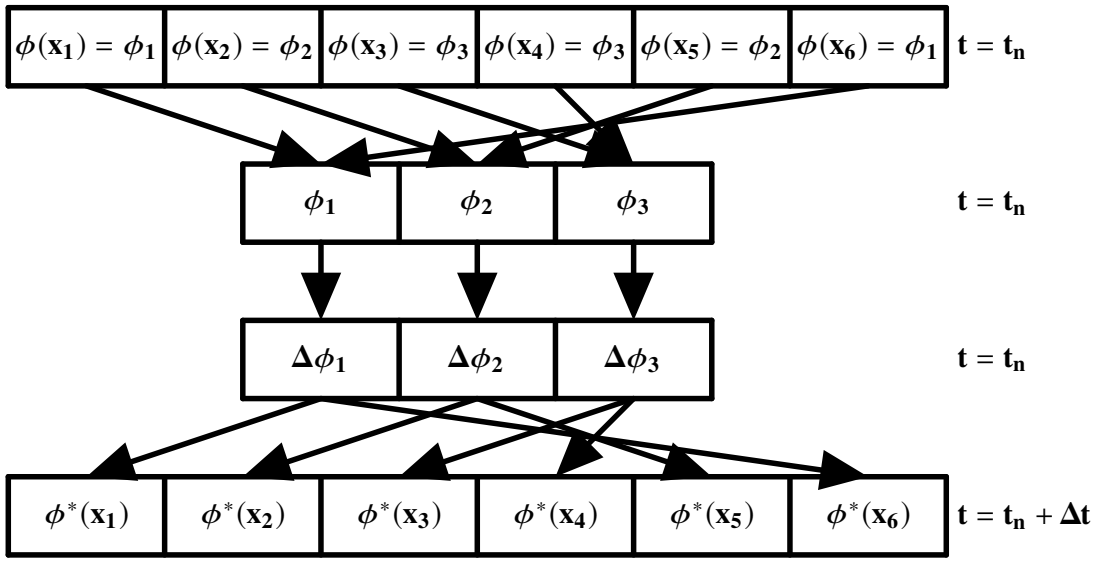
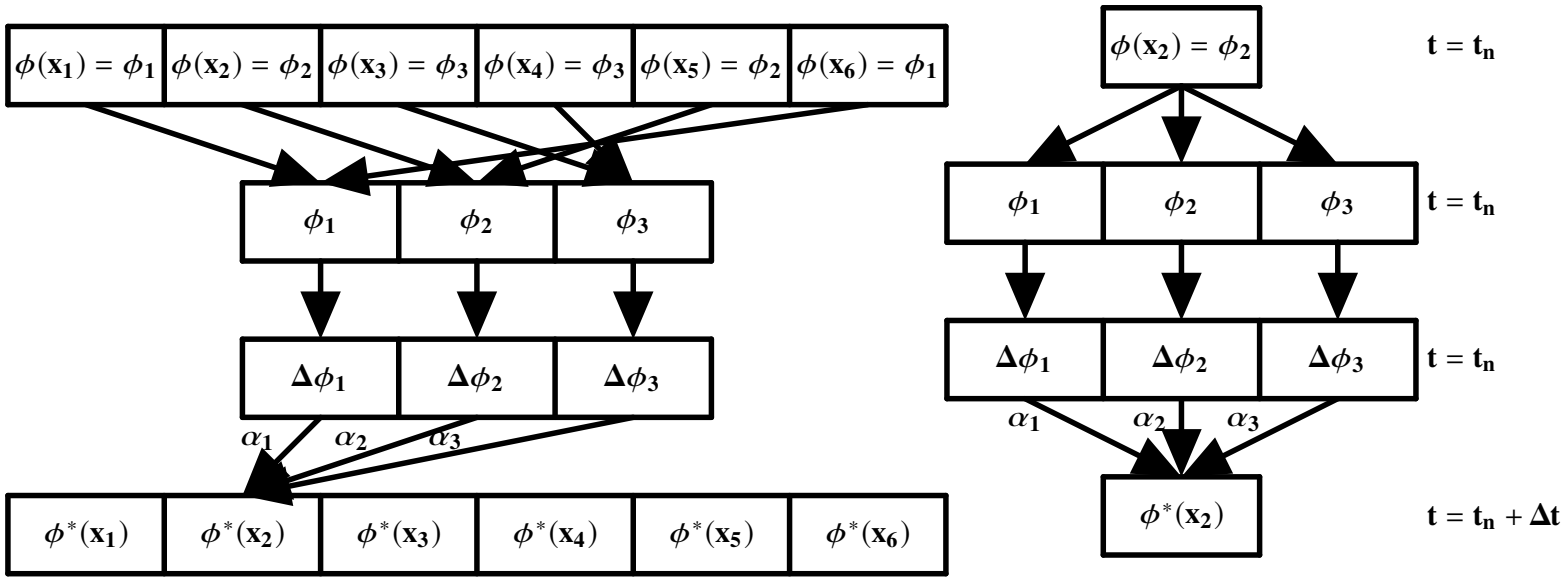


Fig. 5. The concept of the chemistry coordinate mapping approach.

Fig.5 illustrates the ideal of CCM method, as shown in the second line, the composition variables in physical space at time step t_n are mapped into a thermodynamic space according to its value.

The combination of StoR model and CCM approach



This figure shows the ideal of the combination of StoR and CCM. In this sketch, from line three to line four, the composition variables state $\mathbf{\Phi}^*$after the chemistry fractional step are updated by all of the chemical results in phase space at time t_n with a set of coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$.

Compile the library in user dictionary without modification

The CCM approach is used to speedup the chemistry calculation, where chemistry library is located at $FOAM_SRC/thermophysicalModels/chemistryModel.

```
OFv1806
foam
cp -r --parents src/thermophysicalModels/chemistryModel $WM_PROJECT_USER_DIR
cd $WM_PROJECT_USER_DIR/src/thermophysicalModels/chemistryModel
```

Revise the the final 'libchemistryModel.so' location from $FOAM_LIBBIN to $FOAM_USER_LIBBIN in Make/files, and check whether the changes has been made by using 'tail' command. Compile it after changing.

```
sed -i s/FOAM_LIBBIN/FOAM_USER_LIBBIN/g Make/files
tail Make/files
wmake
```

Create a CCM chemistry model dictionary form StandardChemistryModel

Check the location of new compiled library by using ‘which’ and ‘grep’ command.

```
ldd ‘which sprayFoam‘ | grep libchemistryModel.so
```

Copy the StandardChemistryModel folder and rename it as CCMChemistryModel, rename the files in CCMChemistryModel dictionary.

```
cp -r chemistryModel/StandardChemistryModel chemistryModel/CCMChemistryModel
mv chemistryModel/CCMChemistryModel/StandardChemistryModel.C chemistryModel/CCMChemistryModel/CCMChemistryModel.C
mv chemistryModel/CCMChemistryModel/StandardChemistryModel.H chemistryModel/CCMChemistryModel/CCMChemistryModel.H
mv chemistryModel/CCMChemistryModel/StandardChemistryModelI.H chemistryModel/CCMChemistryModel/CCMChemistryModelI.H
sed -Ei 's/StandardChemistryModel/CCMChemistryModel/g' chemistryModel/CCMChemistryModel/CCMChemistryModel*.*
sed -Ei 's/standard/CCM/g' chemistryModel/CCMChemistryModel/CCMChemistryModel*.*
```

By typing above command, the copy and revision of StandardChemistryModel is finished, we can look into the files to check by yourself. It is worth mention that you still need to, there is a keyword named ‘standard’, convert it to ‘CCM’ as well, please do not miss that.

Find out all of the related files and replace them

It is harder than combustion model, because there is a mutual reference between CCM class and chemistry solver class. We can use grep command to find out the related files:

```
grep -r TDAC
```

```
/*******************   FILES NEED TO BE MODIFIED   *******************/
chemistryModel/BasicChemistryModel/BasicChemistryModels.C
chemistryModel/basicChemistryModel/basicChemistryModelTemplates.C
chemistrySolver/chemistrySolver/makeChemistrySolverTypes.H

chemistryModel/CCMChemistryModel/CCMChemistryModel.C
chemistryModel/CCMChemistryModel/CCMChemistryModel.H
chemistryModel/CCMChemistryModel/CCMChemistryModelI.H
/*************************************************************************/
```

```
cp $FOAM_RUN/CCM/BasicChemistryModels.C chemistryModel/BasicChemistryModel
cp $FOAM_RUN/CCM/basicChemistryModelTemplates.C chemistryModel/basicChemistryModel
cp $FOAM_RUN/CCM/makeChemistrySolverTypes.H chemistrySolver/chemistrySolver
```

Find out all of the related files and replace them

```
cp $FOAM_RUN/CCM/CCMChemistryModel.C chemistryModel/CCMChemistryModel
cp $FOAM_RUN/CCM/CCMChemistryModel.H chemistryModel/CCMChemistryModel
cp $FOAM_RUN/CCM/CCMChemistryModelI.H chemistryModel/CCMChemistryModel
```

Replace the related files one by one, then update the lnInclude links and compile to get a new 'libchemistryModel.so' library with 'CCM' model.

```
touch chemistryModel/makeChemistryModel.H
wmakeLnInclude -u .
wmake
```

So far, a new chemistry speedup approach named 'CCM' has been created. Next, we will create a case to run 'sprayFoam' solver with 'CCM' approach to test it.

The creation of a case ‘aachenBombCCM’ with CCM approach

```
cp -r $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb $FOAM_RUN/aachenBombCCM
cp -r $FOAM_TUTORIALS/combustion/reactingFoam/laminar/counterFlowFlame2D/constant/combustionProperties\
 $FOAM_RUN/aachenBombCCM/constant
```

Copy a spray combustion tutorial ‘aachenBomb’ to ‘$FOAM_RUN’ dictionary, renamed it as ‘aachenBombCCM’. And replace the constant/combustionProperties file from ‘counterFlowFlame2D’ case, which employ a ‘laminar’ combustion model. Add an additional configuration file for CCM named ‘CCMProperties’ into ‘constant’ folder:

```
cp $FOAM_RUN/CCM/CCMProperties $FOAM_RUN/aachenBombCCM/constant
```

The creation of a case 'aachenBombCCM' with CCM approach

Add a line 'CCM on;' in constant/chemistryProperties/chemistryType dictionary. It should be like that:

```
/***************   constant/chemistryProperties   ***************/
...
chemistryType
{
    chemistrySolver   ode;
    chemistryThermo   psi;
    CCM         on;
}
chemistry       on;
initialChemicalTimeStep 1e-07;
...
// ************************************************************* //
```

Create mesh and run the case to see the printed information whether the CCM approach works.

```
blockMesh -case $FOAM_RUN/aachenBombCCM
sprayFoam -case $FOAM_RUN/aachenBombCCM
```

If the printed information is like the following outputs and no error printed, it means that CCM works.

```
/*******************   PRINTED INFORMATION   *******************/
...
Selecting combustion model laminar<psiChemistryCombustion>
Selecting chemistry type
{
    chemistrySolver ode;
    chemistryThermo psi;
    CCM             on;
}
...
CCMChemistryModel: Number of species = 5 and reactions = 1
Reading CHEMKIN thermo data in new file format
 J_H_Ox= 0.0024 J_H_fu= 0.16
 J_C_Ox= 0.025 J_C_fu= 0.84
 J_O_Ox= 0.233 J_O_fu= 0
...
0 min(MZ)= 0 max(MZ)= 20 ZoneSpan= 0.01 Nu. Zone in CCMMesh 2001
1 min(MZ)= 300 max(MZ)= 3000 ZoneSpan= 5 Nu. Zone in CCMMesh 541
2 min(MZ)= 0 max(MZ)= 1 ZoneSpan= 0.025 Nu. Zone in CCMMesh 41
Selecting ODE solver seulex
...
```

The introduction of a spray and combustion case

In this chapter, we will dig into the ‘aachenBomb’ tutorial case, explain the case geometry and injector configuration, chemical mechanism for a typical sprayFoam combustion case.

The case ‘aachenBomb’ in OpenFOAMv1806 is located at $FOAM_TOTORIALS/lagrangian/sprayFoam/aachenBomb.

The geometry of the ‘aachenBomb’ is a cuboid, as shown in Fig. 6. The injector is located on the top center of the geometry, which is specified in the file constant/sprayCloudProperties:

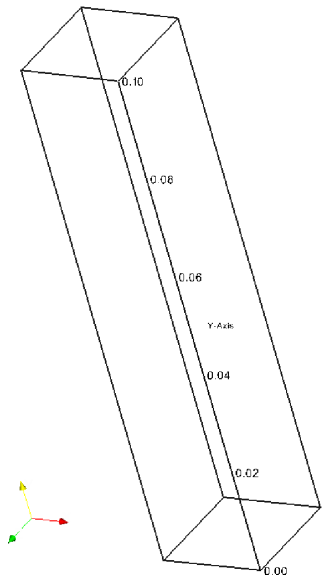


Fig. 6. Geometry of the aachenBomb tutorial case.

The introduction of a spray and combustion case

After injection and evaporation, liquid fuel will come into gas phase and react with oxygen, the reaction of the gas phase are defined in constant/thermalphysicalProperties.

```
/*******************   thermophysicalProperties   *******************/
...
CHEMKINFile     "$FOAM_CASE/chemkin/chem.inp";
CHEMKINThermoFile "$FOAM_CASE/chemkin/therm.dat";
CHEMKINTransportFile "$FOAM_CASE/chemkin/transportProperties";
...
inertSpecie     N2;
liquids
{
    C7H16;
}
...
// ************************************************************************ //
```

where we can found that the location of CHEMKINFile, CHEMKINThermoFile and CHEMKINTransportFile are specified, the solver will call for the functions in thermal class to read thermophysicalProperties and find out the chemkin file location, read the reactions, species and its thermal properties form those three files.

The introduction of a spray and combustion case

Let's see the CHEMKINFile, it is located in "$FOAM_CASE/chemkin/chem.inp".

```
/*******************   $FOAM_CASE/chemkin/chem.inp   *******************/
ELEMENTS
 H   O    C   N   AR
END
SPECIE
C7H16 O2 N2 CO2 H2O
END
REACTIONS
 C7H16 + 11O2            => 7CO2 + 8H2O        5.00E+8  0.0   15780.0! 1
FORD / C7H16 0.25 /
FORD / O2 1.5 /
END
// ********************************************************************* //
```

ELEMENTS keyword is the list of total elements in reaction solver system, SPECIE list is the reaction species such as fuel oxygen products and intermidiate products. REACTIONS define the total reaction in the system, each reaction consist of reactants products and three coefficients which is corresponding to the Arrhenius theory. Some of the reaction contain the reverse reaction. In this tutorial, a one step reaction is employed.

Case modification

Let us source the OpenFOAM at first, copy and rename the tutorial case we introduce above.

```
foam
cp -r $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb\
$FOAM_RUN/aachenBombCCMStoR
cd $FOAM_RUN/aachenBombCCMStoR
```

Come into the folder and revise the combustionProperties file as follows, in this revised file, combustion model is set as StoR, active is set as true, means that the chemical reaction calculation is on, this is achieved by a judgment statement if(this->active()) as we can see in StoR functions, such as correct() and R().

```
vi constant/combustionProperties
```

Case modification

```
/****************   constant/combustionProperties   ****************/
...
combustionModel  StoR;
active  true;
StoRCoeffs
{
    //For Presumed PDF
    temperaturePDF          NormalDistribution;
    //From (-1 \sigma, +1 \sigma),1 stands for 1*standard variance.
    truncationForPDF        1;
    //Similarity model for getting variance of T
    //it is the similarity coefficients, default 1.
    deviationSimilarCoeff   1;
}
// ************************************************************** //
```

StoRCoeffs dictionary is used to set the parameters for StoR combustion model, such as PDF distribution and variance coefficient. Specifically, temperaturePDF is the name of presumed PDF, truncationForPDF is the truncation boundary for the PDF function, for example, if we set it as 3, means that the temperature of generated stochastic reactor will located in the region $[\overline{T} - 3\sigma, \overline{T} + 3\sigma]$, where σ is the variance of the local temperature, $\overline{T}$ is the averaged value within the cell. Parameter deviationSimilarCoeff is a scale coefficient used to calculate the variance of temperature. As shown in Eq. 2.

Case modification

```
vi constant/chemistryProperties
```

After revising the combustionProperties, lets come to modify the chemistryProperties file, it is located in constant/chemistryProperties and used to specify the chemistry model, we only need to add a line in chemistryType dictionary to turn CCM on, the other coefficients will be read in constant/CCMProperties files.

```
/***************   constant/chemistryProperties   ***************/
...

chemistryType
{
    chemistrySolver   ode;
    chemistryThermo   psi;
    CCM         on;     //add a line to choose CCM chemistry approach
}

chemistry       on;

initialChemicalTimeStep 1e-07;
...
```

Case modification

```
cp $FOAM_RUN/CCMProperties $FOAM_RUN/aachenBombCCMStoR/constant
vi constant/CCMProperties
```

Let's have a look at constant/CCMProperties file, it will be loaded at the construction of CCM class object or calling readCCMproperties() function when we turn CCM on in chemistryProperties.

```
/******************   constant/CCMProperties   ******************/
...
babaCCM                                on;
ratioOxygenToCarbonElementInFuel       0; //fuel:C7H16;
chemicallyFrozenT                      chemicallyFrozenT [0 0 0 1 0] 600.;
maxFlammabilityLimit                   maxFlammabilityLimit [0 0 0 0 0] 20;
minFlammabilityLimit                   minFlammabilityLimit [0 0 0 0 0] 1e-6;
...
min:max:SpanZoneJe  0   20  0.01;
min:max:SpanZoneT   300 3000 5;
min:max:SpanZoneXi  0   1   0.025;
...
```

Case modification

After modifying those combustionProperties, chemistryProperties and CCMProperties, the case setup is done. But for convenience, we create two batch files to run the case automatically.

```
touch Allrun
vi Allrun
```

Create two batch files and edit them by using ‘touch’ and ‘vi’ command.

```
/*************************     Allrun     *************************/

#!/bin/sh
cd ${0%/*} || exit 1                        # Run from this directory
. $WM_PROJECT_DIR/bin/tools/RunFunctions    # Tutorial run functions

runApplication blockMesh
runApplication $(getApplication)

#------------------------------------------------------------------
```

Case modification

```
touch Allclean
vi Allclean
```

The same with batch file Allclean:

```
/*************************   Allclean   **************************/

#!/bin/sh
cd ${0%/*} || exit 1                          # Run from this directory
. $WM_PROJECT_DIR/bin/tools/CleanFunctions  # Tutorial clean functions

cleanCase

#------------------------------------------------------------------
```

Case execution

Allocate execution rights to the batch files by using command ‘chmod’, check the privilege of Allclean and Allrun files:

```
chmod 755 All*
ll All*
```

You can get the following results, means that batch files is executable now.

```
/*********************   PRINTED INFORMATION   **********************/
-rwxr-xr-x 1 ... ... 243 nov 24 12:02 Allclean*
-rwxr-xr-x 1 ... ... 289 nov 24 12:02 Allrun*
```

Run the case and wait for completion, see the usage of StoR combustion model CCM chemistry approach, and find out the total execution time.

```
./Allrun
more $FOAM_RUN/aachenBombCCMStoR/log.sprayFoam
tail $FOAM_RUN/aachenBombCCMStoR/log.sprayFoam
```

Case execution

In order to compare the results and computational efficiency under the usage of combustion model and chemistry speedup approach, we can run a default tutorial case 'aachenBomb' without any modification as a contrast. The command is as follows:

```
cp -r $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb\
 $FOAM_RUN/aachenBomb
cd $FOAM_RUN/aachenBomb
cp $FOAM_RUN/aachenBombCCMStoR/All* $FOAM_RUN/aachenBomb
./Allclean
./Allrun
```

Computational efficiency

Until now, four cases have been tested: the default 'aachenBomb' case with partial stirred reactor combustion model (PaSR), the StoR model case without chemistry speedup approach, CCM with laminar combustion model (WSR), CCM with stochastic reactor combustion model (StoR). For each of the cases, we extract the execution time, table. 1 shows the detailed case information.

Case	Case Name	combustion model	Execution Time(s)
1	aachenBomb	PaSR	38836
2	aachenBombStoR	StoR	
3	aachenBombCCM	WSR	10810
4	aachenBombCCMStoR	StoR	25804

Table 1. Initial conditions for the dieselFoam tutorial

Temperature variance

Fig.7 is the temperature variance distribution calculated by StoR combustion model at ignition time. The larger variance is, the strong turbulence interaction are.

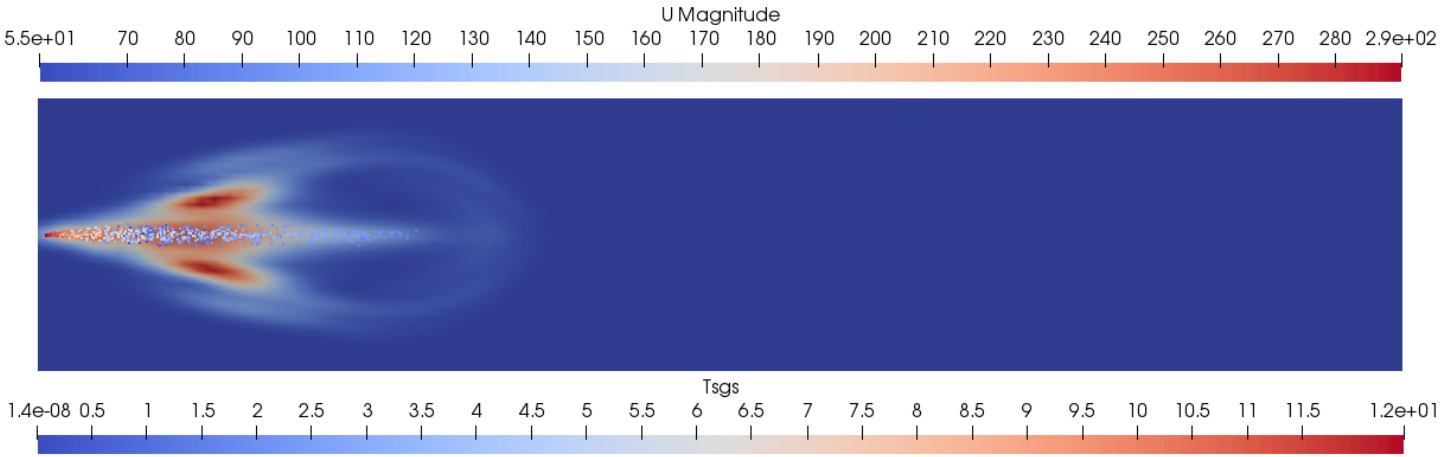


Fig. 7. The temperature variance at ignition time.

This figure shows that the temperature variance along the liquid particles is not negligible. StoR and PaSR combustion can take turbulence interaction into account, in theory, it can predict ignition time more correct.

Chemistry active zones in CCM

Fig.8 shows the active CCM zones in computational domain at ignition time. The cell with same zone number share identical CCM zone.

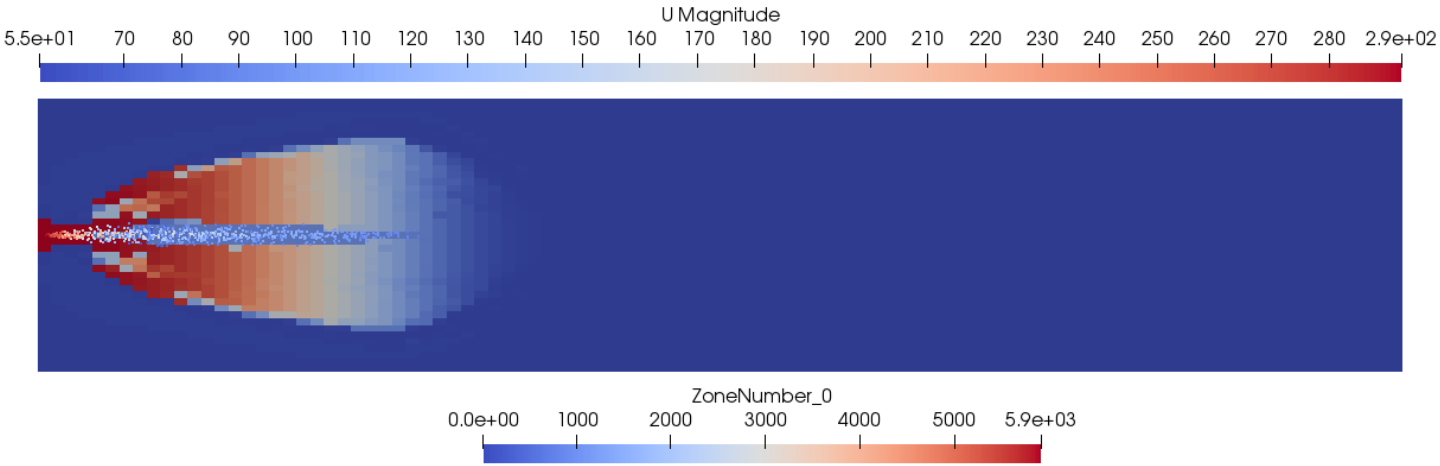


Fig. 8. The active zone number in CCM map for the aachenBombCCM case.

As can be seen in Fig.8, all of the blue color share one CCM zone which will be calculate only once. The maximum zone number is 5900, while the total cell is 160000, this will speed up the reaction calculation more than 30 times.

Temperature distribution at ignition time

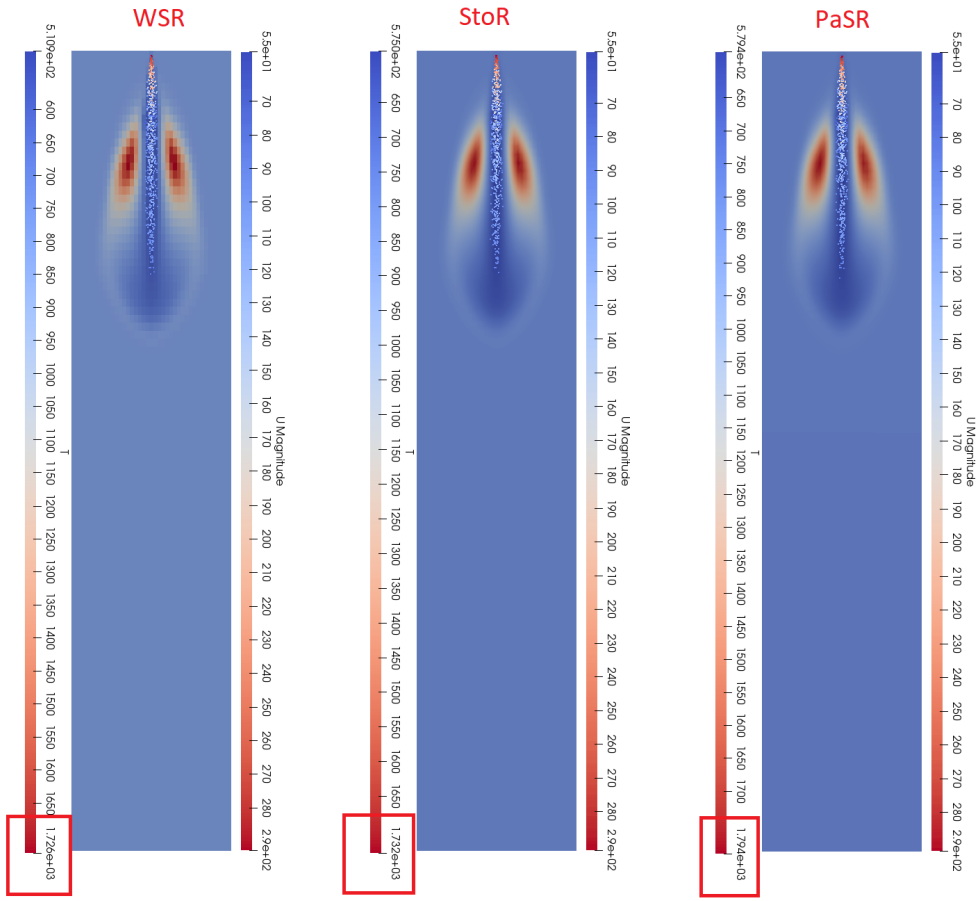


Fig. 9. Temperature distribution with WSR, StoR, PaSR combusion model.

Acknowledgements

- Course Lecturer: Prof. Håkan Nilsson
- Technical Support: Mohammad Hossein Arabnejad Khanouki
- Supervisors: Prof. Xue-song Bai and Prof. Mehdi Jangi