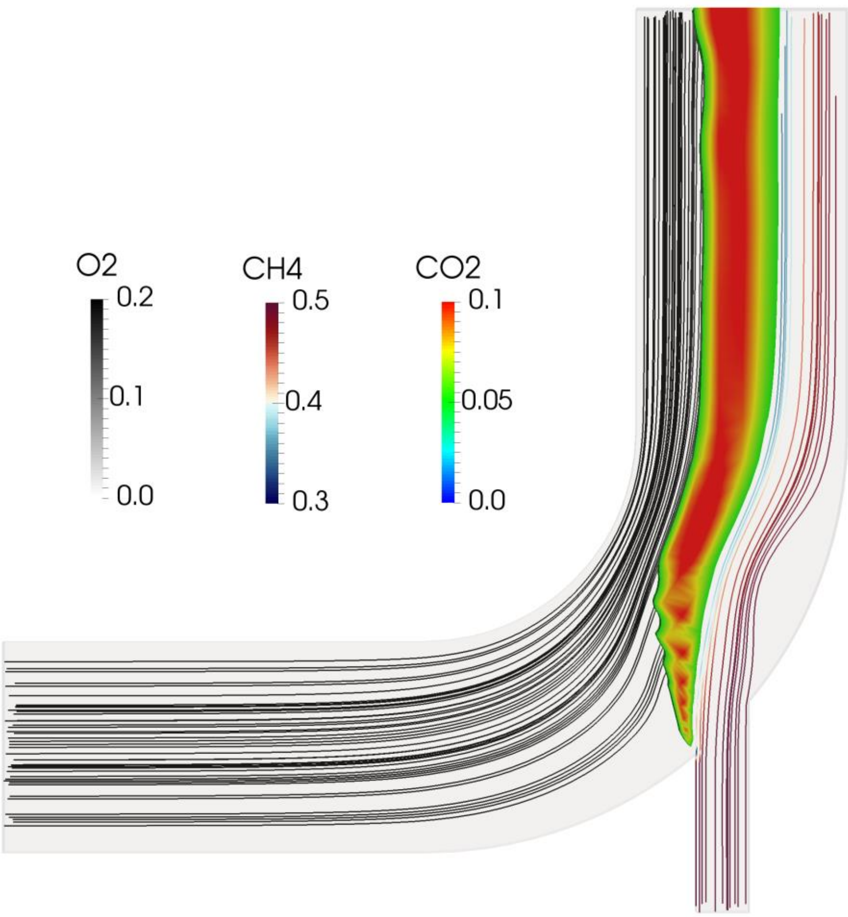
# Tutorial Eleven

# Reaction




4th edition, Jan. 2018

Contributors:

- Bahram Haddadi
- Clemens Gößnitzer
- Sylvia Zibuschka
- Yitong Chen

Compatibility:

- OpenFOAM® 5.0
- OpenFOAM® v1712

Cover picture from:

- Bahram Haddadi



**For more tutorials visit: www.cfd.at**

# Background

There are two common approaches in modeling reactions:

***1. Partially stirred reactor (PaSR) Model***

Partially stirred rector (PaSR) model is used to model thermodynamic and chemical reactions numerically, for example, combustion. In the PaSR approach, a computational cell is split into two different zones: a reacting zone and a non-reacting zone. The reacting zone is modeled as a perfectly stirred reactor (PSR), and all reactants are assumed to be perfectly mixed with each other.

For the reactor, we are interested in three concentrations, 1) mean concentration of key component in the feed, $c_{in}$; 2) mixture concentration in the reacting zone, $c$; 3) concentration at the reactor exit$c_{exit}$.

In the reacting zone, reaction occurs for a duration of $\tau_c$, so the concentration of mixture changes from $c_{in}$ to $c$. In the non-reacting zone, the reacted mixture is getting mixed up with the non-reacted mixture for a duration of $\tau_{mix}$, resulting in the final exit concentration, $c_{exit}$.

A key parameter to be calculated in this model would be the reaction rate, and it is clear that the reaction rate is proportional to the ratio of the chemical reaction time to the total conversion time in the reactor (i.e. sum of reacting and mixing time), $\kappa_k$:

$$\kappa_k = \frac{\tau_c}{\tau_c + \tau_{mix}}$$

***2. Eddy dissipation concept (EDC) Model***

The Eddy Dissipation Concept (EDC) model looks at the interaction between reaction and turbulence, where the overall reaction rate is controlled by turbulent mixing. It is widely used for combustion modeling for a great variety of combustion environments with great success.

It is assumed in the model that most reaction takes place within fine turbulence structures, which are modeled as perfectly-mixed reactors. We need to know the reaction mass fraction and the mass transfer rate between the fine structures and its surrounding fluid.

The mass fraction occupied by the fine structures, $\gamma^*$, is expressed as:

$$\gamma^* = \left\{\frac{u^*}{u'}\right\}^2$$

Where $u^*$ is the mass average fine structure velocity. The fine structures are located in regions with nearly constant turbulent kinetic energy given by $u'^2$.

The mass transfer rate between fine structure and surrounding fluid per unit of fluid and per unit of time is modeled as:

$$\dot{m} = 2 \cdot \frac{u^*}{L^*} \cdot \gamma^*$$

where $L^*$ is the characteristic length of the fine structure.

# reactingFoam – reactingElbow

## Simulation

Use the reactingFoam solver, simulate combustion of $CH_4$ and $O_2$ in a mixing elbow:

- Use the two times finer Hex mesh from Example One
- Domain initially filled with $N_2$
- velocity-inlet-5:
  - Velocity: 1 m/s
  - Mass fractions: 23 % $O_2$, 77 % $N_2$
  - Temperature: 800 K
- velocity-inlet-6:
  - Velocity: 3 m/s
  - Mass fractions: 50 % $CH_4$, 50 % $N_2$
  - Temperature: 293 K
- Operating pressure: $10^5$ Pa
- Operating temperature: 298 K
- Isolated walls

## Objective

- Understanding multi-species and reaction modeling in OpenFOAM®

## Data processing

Evaluate your results in ParaView.

# 1. Pre-processing

## *1.1. Copy tutorial*

Copy the following tutorial to your working directory:

$FOAM_TUTORIALS/combustion/reactingFoam/laminar/counterFlowFlame2D

Copy the GAMBIT® mesh from Tutorial One (two times finer mesh) to the case main directory.

## *1.2. 0 directory*

Update all the files in 0 directory with new boundary conditions, e.g. U:

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

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    wall-4
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    velocity-inlet-5
    {
        type            fixedValue;
        value           uniform (1 0 0);
    }

    velocity-inlet-6
    {
        type            fixedValue;
        value           uniform (0 3 0);
    }

    pressure-outlet-7
    {
        type            zeroGradient;
    }

    wall-8
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    frontAndBackPlanes
    {
        type            empty;
    }
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
```

The reaction taking place in this simulation $CH_4$ combusting with $O_2$ creating $CO_2$ and $H_2O$. $N_2$ is the non-reacting species. The boundary condition and initial value of all species should be defined in the 0 directory. These values are mass fractions (between 0 and 1) and dimension less, e.g. $CH_4$:

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

dimensions      [0 0 0 0 0 0 0];
```

```
internalField   uniform 0.0;

boundaryField
{
    wall-4
    {
        type            zeroGradient;
    }

    velocity-inlet-5
    {
        type            fixedValue;
        value           uniform 0; //no CH4 at this inlet
    }

    velocity-inlet-6
    {
        type            fixedValue;
        value           uniform 0.5; //50% CH4 mass fraction at this inlet
    }

    pressure-outlet-7
    {
        type            zeroGradient;
    }

    wall-8
    {
        type            zeroGradient;
    }

    frontAndBackPlanes
    {
        type            empty;
    }
}

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

*Note: If the file for a species does not exist in the 0 directory, the values from Ydefault will be used for that species.*

### *1.3. constant directory*

In the thermophysicalProperties file the physical properties of the species can be set:

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

inertSpecie N2;

chemistryReader foamChemistryReader;

foamChemistryFile "$FOAM_CASE/constant/reactions";

foamChemistryThermoFile "$FOAM_CASE/constant/thermo.compressibleGas";
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *//
```

The mixture type is set to a reacting mixture for calculating the mixture properties and the heat capacities are calculated using “janaf polynomials”.

$N_2$ is defines as `inertSpecie`. In reaction solvers in OpenFOAM® the inert specie is calculated explicitly using the mass balance equation (to satisfy mass conservation):

$$mass\ fraction\ of\ inert\ specie = 1 - \sum mass\ fraction\ of all\ other\ species$$

The species and the reactions are addressed using `foamChemistryFile`. In this simulation reactions and species are read from reactions file in the constant directory:

```
species
(
    O2
    H2O
    CH4
    CO2
    N2
);

reactions
{
    methaneReaction
    {
        type     irreversibleArrheniusReaction;
        reaction "CH4 + 2O2 = CO2 + 2H2O";
        A        5.2e16;
        beta     0;
        Ta       14906;
    }
}
```

OpenFOAM® v1712: Also the elements are listed in the reactions file and an element balance is performed in the calculations!

The species in this simulation are $O_2$, $H_2O$, $CH_4$, $CO_2$ and $N_2$. They are defined in the `species` sub-dictionary. In the `reactions` sub-dictionary, reactions are specified. The reaction of methane combustion is defined and it is of `type` irreversible Arrhenius reaction, `irreversibleArrheniusReaction`.

In the Tutorial Two it was explained that the coefficients for calculating gas mixture properties are defined in the `mixture` sub-dictionary because it was a homogeneous mixture. But in this example the mixture is not homogenous so coefficients for calculating properties of each species are needed separately to calculate mixture properties based on each cell composition. The coefficients of each species are defined in the foamChemistryThermoFile, which reads the file *thermos.compressibleGas* from the constant directory (this step is outlined in the thermophysicalProperties file). For example, the `O2` coefficients for each model are shown below:

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

O2
{
    specie
    {
        molWeight       31.9988;
    }
    thermodynamics
    {
```

```
        Tlow            200;
        Thigh           5000;
        Tcommon         1000;
        highCpCoeffs    ( 3.69758 0.00061352 -1.25884e-07 1.77528e-11 -
                          1.13644e-15 -1233.93 3.18917 );
        lowCpCoeffs     ( 3.21294 0.00112749 -5.75615e-07 1.31388e-09 -
                          8.76855e-13 -1005.25 6.03474 );
    }
    transport
    {
        As              1.67212e-06;
        Ts              170.672;
    }
}
…

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

OpenFOAM® v1712: Number of elements in the specie is listed in this file!

In the `thermodynamics` sub-dictionary the janaf polynomial model coefficients for calculating the heat capacity can be found and in `transport` the sutherland model coefficients for viscosity are stored.

### *1.4. system directory*

By setting the `adjustTimeStep` to `yes` in the controlDict, the solver automatically ignores `deltaT`, and calculates the `deltaT` based on the maximum Courant number `maxCo` defined for it. Change the `endTime` to `120` (approximately one time the volumetric residence time based on velocity-inlet-5) and `writeTimeInterval` to `10`, to write every 10 s to case directory.

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

application     reactingFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         120;

deltaT          1e-6;

writeControl    adjustableRunTime;

writeInterval   10;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep  yes;

maxCo           0.4;
```

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

## 2. Running simulation

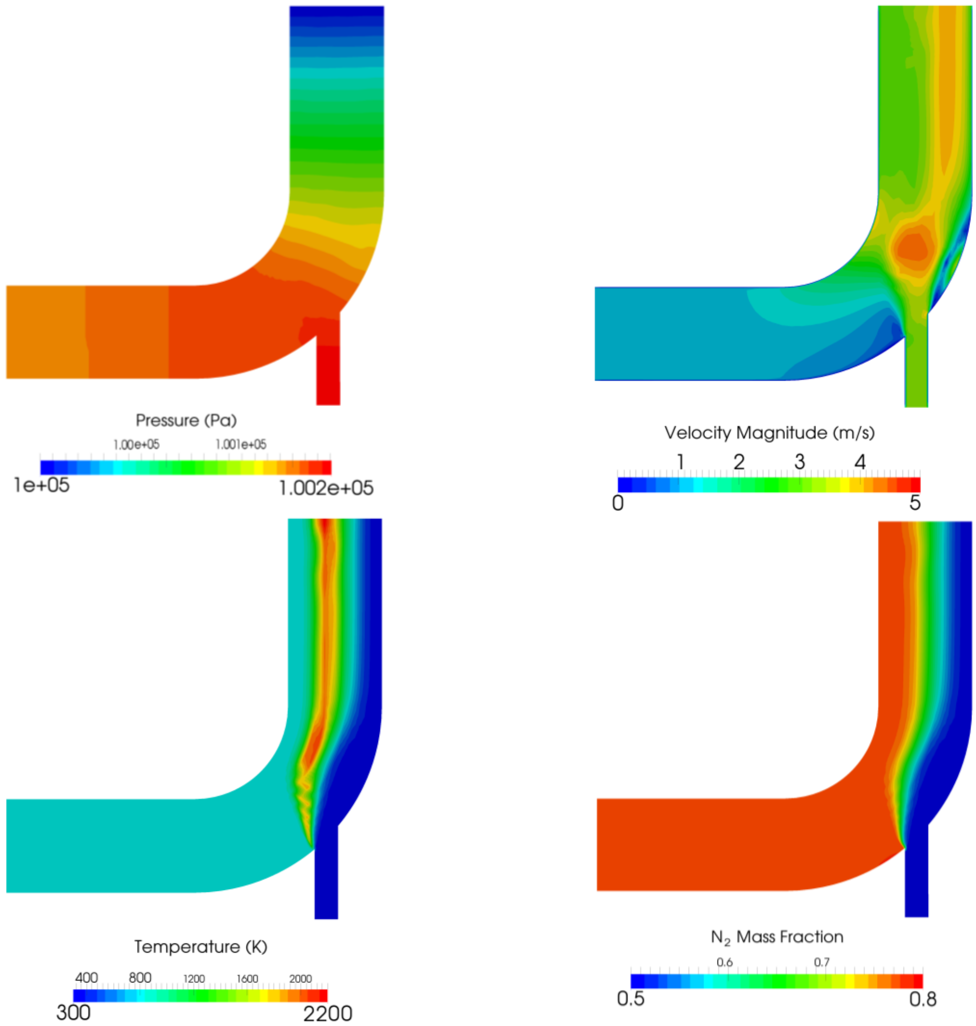
```
>fluentMeshToFoam fineHex.msh
```

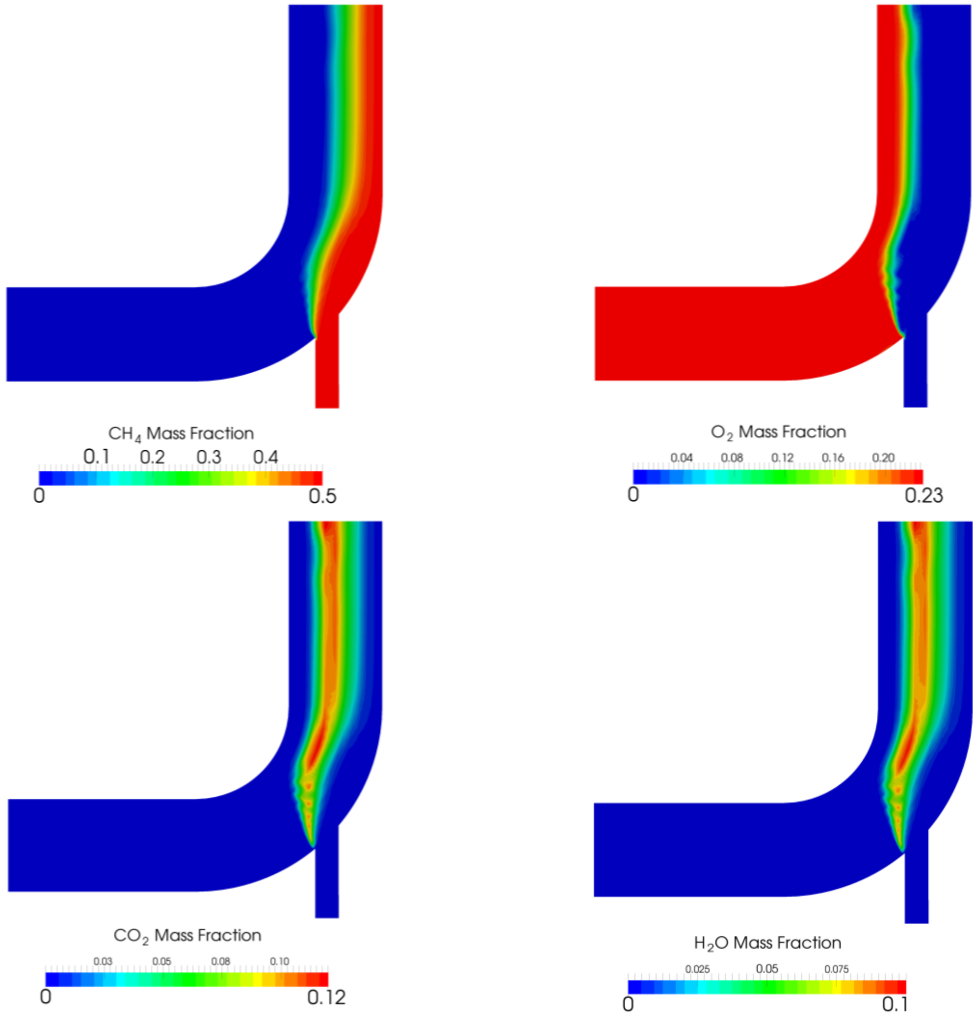
After converting the mesh, check the boundary file in the constant/polyMesh directory and change the `type` and `inGroups` of boundary `frontAndBackPlanes` from `wall` to `empty` (it is a 2D simulation).

```
>reactingFoam
>foamToVTK
```

## 3. Post-processing

The simulation results are as follows:






Simulation results after 120 s