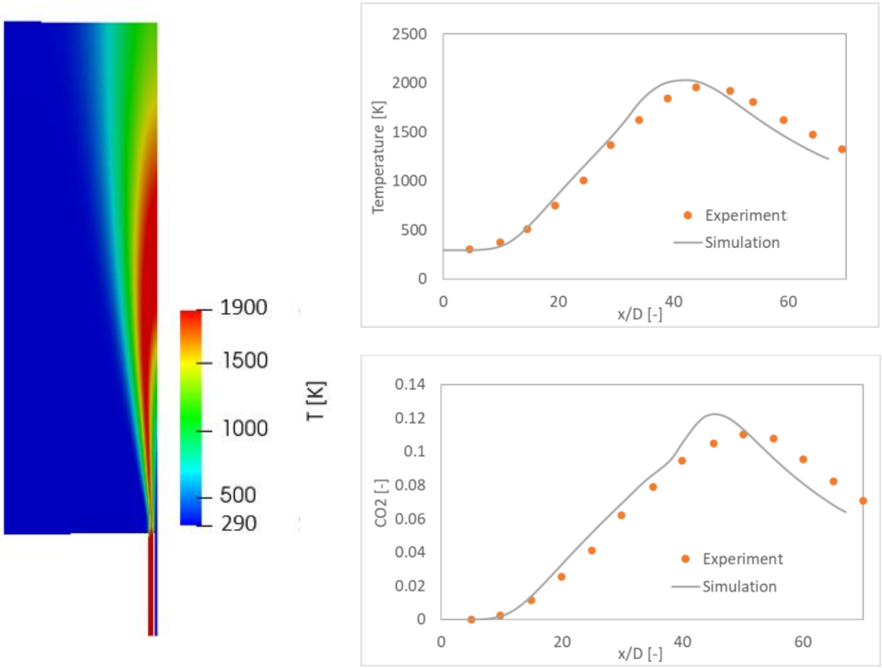
CEMF.ir
Center of Engineering
Multiscale Modeling
Fluid Flow

reactingFoam

One: Sandia D Flame



How to simulate combustion of a flame using OpenFOAM®

Compatible with	OpenFOAM® 7| OpenFOAM® 6 OpenFOAM® v1912

Authors:

Majid Mansouri Borujeni   Hamid Reza Norouzi

Amirkabir University of Technology

Center of Engineering and Multiscale Modeling of Fluid Flow

License Agreement

Extra consideration:

- This document is developed to teach how to use OpenFOAM® software. The document has gone under several reviews to reduce any possible errors, though it may still have some. We will be glad to receive your comments on the content and error reports through this address: h.norouzi@aut.ac.ir

Document history

Revision	Description	Date
Rev1.1	Final version was prepared, tutorial was performed using OpenFOAM 7 and OpenFOAM v1912.	Nov. 4, 2020
Rev1	The first draft was commented.	Sep. 21, 2020
Rev0	The first draft was prepared and run with OpenFOAM 6.	August 16, 2020

Table of Contents

Prerequisites

You need to be familiar with basics of OpenFoam® to start this tutorial.

How to get simulation setup files?

You have two options to get simulation setup files:

- **Tutorial cases:** execute the following command to copy one of the tutorial cases of OpenFoam® to the desktop of your computer. Then, you will need to make the necessary changes to the setup cases based on the instructions given in this tutorial.

  ```
  > cp -r $FOAM_TUTORIALS/combustion/reactingFoam/RAS/SandiaD_LTS Desktop/
  ```

- **Website:** simulation setup files (a compressed file) are uploaded on https://www.cemf.ir alongside this PDF tutorial file.

1. Brief Description of reactingFoam

reactingFoam is a transient solver for simulating compressible, laminar/turbulent reactive systems. This solver uses stoichiometry expressions and kinetic data of reactions to obtain consumption and production rate of species. In addition, this solver supports some thermos-physical models to obtain required properties of the fluid mixture phase.

Special numerical methods are available to obtain reaction rate of fast reactions with high accuracy and to prevent instabilities caused by stiff ode systems. Various implemented thermos-physical and reaction models in this solver make it possible to simulate most of reactive systems in the engineering applications. It is also possible to extend the existing models using C++ programming.

Various combustion models also accompany the solver to simulate turbulent combustion systems. For complex combustion systems, some utilities are provided to easily import kinetic data from other sources.

1.2. Chemistry and Kintetics

The most crucial part of simulating chemical reactions, including combustion, is to determine the system's chemical properties. Therefore, at first, it is needed to specify all the available chemical species and their associated models for computing physical properties. OpenFOAM should know molecular weight, density, enthalpy/internal energy, transport properties of all species. These quantities are obtained based on the thermos-physical models that are selected by user. Each thermos-physical model may require specific data that should be supplied.

For defining species properties, three sets of data should be supplied: molecular weight, thermodynamic properties (EOS and enthalpy/internal energy model), and transport properties of species. The set of reactions is defined in a separate file where the reaction stoichiometry and rate based on the Arrhenius type equation (or other types) is specified for each reaction.

One way to define reaction set and species properties in OpenFOAM is to use available reaction and species data reported in chemkin file format. The reaction paths for combustion of hydrocarbons (here, methane) and related physical properties can be obtained form some sources like Ansys Chemkin package. Here we use GRI 3.0 mechanism. Files of this mechanism is also available at GRI-Mech™ website[1]. By using chemkinToFoam utility, chemkin thermodynamics, transport, and reaction data would be converted into OpenFOAM file format. The procedure will be discussed later.

1.3. Combustion Models

No reaction takes place without contact of reactants. A proper mechanism for mixing of species is required when a reactive system is modeled. Combustion reactions have fast kinetic, so the mixing rate of reactants has significant impact on the reaction rate. Generally, combustion models are the way that interactions between the reactions and the fluid flow i.e. the turbulence and mixing, are treated [Magnussen, B.F. (2005). THE EDDY DISSIPATION CONCEPT A BRIDGE BETWEEN SCIENCE AND TECHNOLOGY]. In simple words, combustion models take mixing into account along with kinetic data of reaction to compute the rate of the combustion. Available combustion models in OpenFOAM are listed in Table 1.

Table 1: Available Combustion Models in OpenFOAM

Model	Description	Model Inlets
laminar	Estimate flames combustion as a laminar flame	-
none	no combustion	-
PaSR	Splitting each cell to the reacting and the mixing zones	Depending on mesh resolution, the Cmix parameter can be used to scale the turbulence mixing time scale.

[1] http://combustion.berkeley.edu/gri-mech/version30/text30.html#thefiles

EDC	Splitting flow to the reacting and non-reacting zones	Only desired version of model must be specified. Available versions are: v1981, v1996, v2005, and v2016.
zoneCombustion	Enable the reactions within the specified list of cell-zones and set to zero elsewhere.	
infinitelyFastChemistry	Mixed is burnt. Mostly used for large eddy simulations because of its low computational cost.	Additional parameter C is used to distribute the heat release rate.in time.

laminar model neglects all turbulent fluctuations and considers it as a laminar flame. infinitelyFastChemistry model is based on the principle of "mixed is burnt", which means it doesn't consider any limitations on reaction rate because of kinetics. It has a lower computational cost than other models and could be used only for non-premixed flames.

PaSR model or "Partially Stirred Reactor" model, assumes that every cell is composed of two different zones: reaction zone and mixing zone. The reaction zone is considered as a perfectly stirred tank reactor. In each cell, the reaction zone is surrounded by the mixing zone. Thus, reaction zone solely exchange masss with the mixing zone.

A simple schematic of this model is shown in Figure 1, where C_0 is concentration of cell inlet, C is the concentration of the reacting mixture in the reaction zone, C_1 is the concentration of cell outlet, and k is the time constant. Due to the mixing between two zones and reaction, the outlet concentration C_1 is obtained [J. Chomiak, Combustion A Study in Theory, Fact and Application, Abacus Press, New York (1990)]:

$$\frac{C_1 - C_0}{\Delta t} = k\, R(C_1)$$

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

where τ_c and τ_{mix} are chemical and mixing characteristic times and $R(C_1)$ is the reaction rate. When the mixing time is small (rapid mixing), the contacnteration C_1 is determined by reaction. On the other hand, when chemical time is small (reaction is slow in comparison to mixing), the mixing – or turbulence – controls the concentration C_1. The mixing time is obtained from [Nordin, Complex Chemistry Modelling of Diesel Spray Combustion, PhD Thesis, Chalmers University of Technology (2001)]:

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

where Cmix is the mixing constant. Cmix is 1 for laminar flows; 0, for extrimly trubulent flows; and between 0.001 and 0.3, for typical turbulent flows.

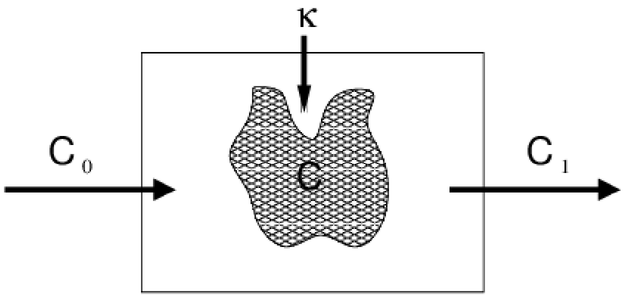


Figure 1, PaSR combustion model[2]

EDC model (Eddy Dissipation Concept) is based on Eddy Dissipation Model. This model splits the fluid into reacting zones (called “fine structures”) and non-reacting zones (called “surroundings”). Fine structures are the regions where the dissipation of turbulence energy takes place. Fine structures have a tube-like shape with the diameter at the order of Kolmogorov’s length-scale. This model was first developed in 1981, but it was later modified. In the first version (v1981), it was assumed that reaction takes place only in the fine structures dut to mass exchange between surrounding and find structures. In the later version (v1996), the mixing effect of fresh gas and fine structures was included. In the 2005-version, the model was modified because of the fact that

[2] Ettorre, D. & Lucchini, Tommaso. (2007). Comparison of Combustion and Pollutant Emission Models for DI Diesel Engines. Combustion. 15. 10.4271/2007-24-0045.

the reactions can also happen outside of the Fine Structures. The goal of the 2016-version was to make the model applicable for non-classic combustion models, like MILD combustions. In OpenFOAM v1981, v1996, v2005, and v2016 versions are available. A simple schematic of this model and its different versions is shown in Figure 2. Circuits show fine structures, while arrows show mass transfer between fine structure and surroundings. [Bösenhofer, M., Wartha, E., Jordan, C., & Harasek, M. (2018). The Eddy Dissipation Concept—Analysis of Different Fine Structure Treatments for Classical Combustion. *Energies*, *11*(7), 1902]

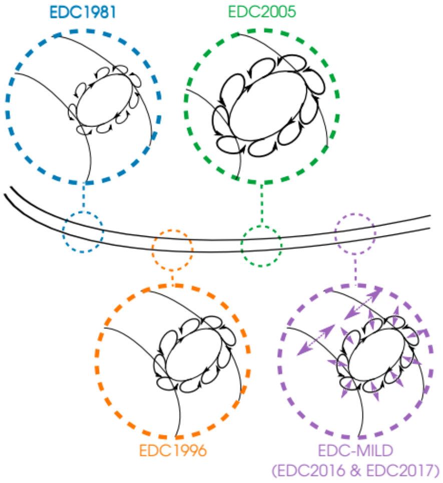


Figure 2, EDC combustion model[3]

2. Problem definition

Combustion occurs in a lot of equipment in different industries. Therefore, it is vital to be able to model combustion for the design and optimization. One of these equipment is flare that is

[3] Bösenhofer, M., Wartha, E., Jordan, C., & Harasek, M. (2018). The Eddy Dissipation Concept—Analysis of Different Fine Structure Treatments for Classical Combustion. *Energies*, *11*(7), 1902.

used most in the energy and petrochemical industries to convert burnable waste gas of the plant into safe and less environmentally-harmful gases.

Here, we simulate the standard Sandia D flame [webpage: https://tnfworkshop.org/data-archives/pilotedjet/ch4-air/]. The flame is composed of two concentric cylinders; the main jet flow exits from the central cylinder and pilot jet exits from the annulus. A simple schematic of the flame and the surroundings are shown in Figure 3. The main jet is a mixture of Methane and air with a mole proportion 1:3, the exit velocity 49.6 m/s ,and temperature 294 K. The pilot jet is the combustion product with the temperature 1880 K and an the exit velocity 11.4 m/s. Air flows parallel to the main jet with the velocity 0.9 m/s. All the necessary information are shown in Table 2.

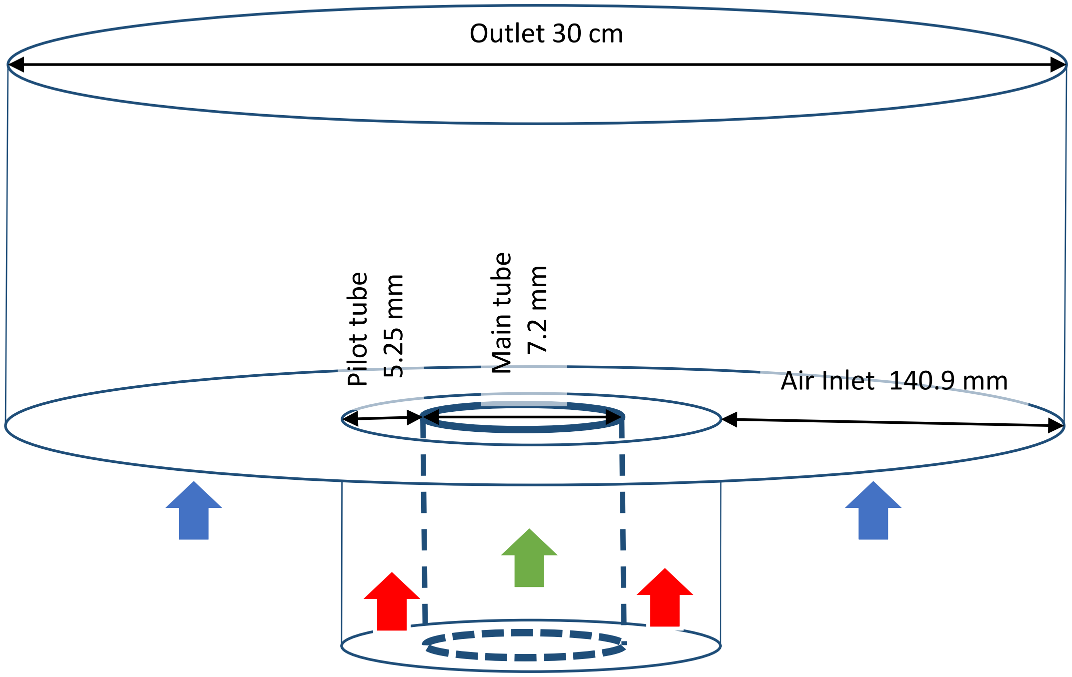


Figure 3, A simple schematic of the flame and its surroundings

The 2005 version of EDC model is used as combustion model here [Magnussen, B.F. (2005). THE EDDY DISSIPATION CONCEPT A BRIDGE BETWEEN SCIENCE AND TECHNOLOGY.]. The simulation is based on k-ε turbulency model and includes a reduced version of GRI 3.0 kinetic mechanism.

Table 2: Physical Properties of Streams

		Streams		
		Main Jet	Pilot Jet	Air Flow
Temperature (K)		294	1880	291
Velocity (m/s)		49.6	11.4	0.9
Mass Compositions	CH_4	0.1561	0	0
	O_2	0.1996	0.054	0.23
	CO_2	0	0.1098	0
	H_2O	0	0.0942	0
	N_2	0.6473	0.7342	0.77

3. Simulation setup

3.1. Creating Geometry and Mesh

The geometry and the mesh properties are defined in system/blockMeshDict. Geometry of flame simulation includes a segment of the main tube, the pilot tube, and the surroundings as shown in Figure 4.

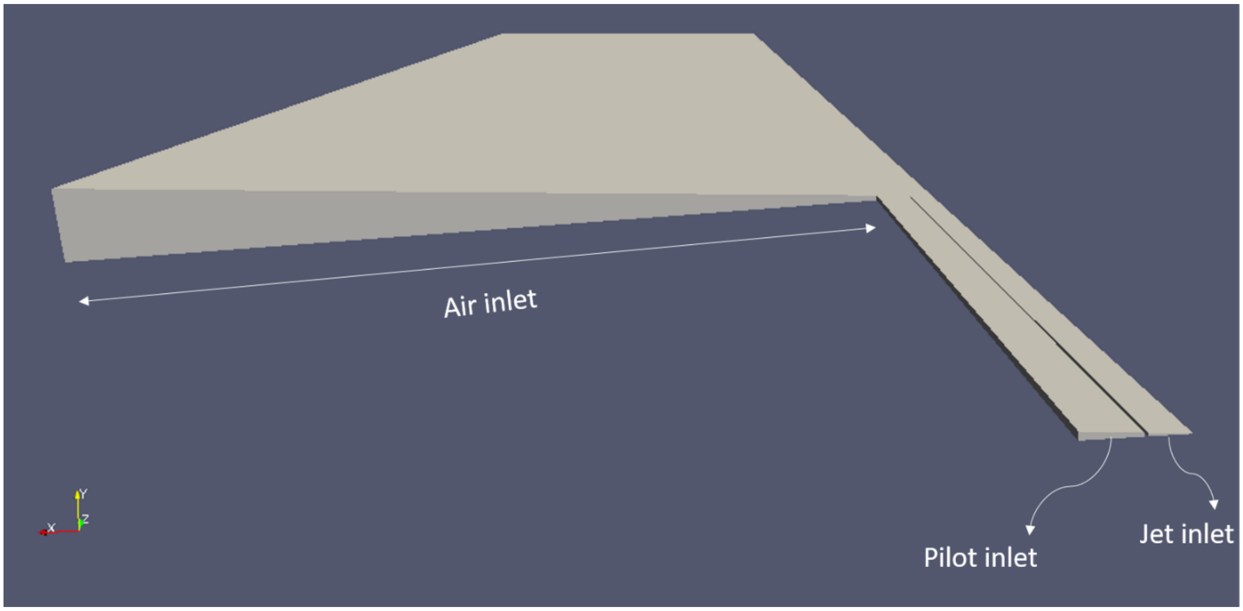


Figure 4, Geometry created by blockMesh

3.2. Reaction and Thermophysical data

The reaction mechanism and kinetics, thermophysical, and transport data are stored in the chemkin file format (grimech30.dat, thermo30.dat and transportProperties), located in chemkin sub-folder. The reaction mechanism used in this tutorial is a simplified version of GRI 3.0 mechanism, which is suitable for simulating the combustion of C1-C3 hydrocarbons. The reaction mechanism includes 36 species and 219 reactions. The original GRI 3.0 mechanism includes 53 species and 325 reactions.

To convert these files into a format that can be interpreted by OpenFOAM, `chemkinToFoam` utility could be used. `chemkinToFoam` needs five inputs: addresses of three chemkin files and the two OpenFOAM files for storing the results. Later you need to specify the full address of these files in constant/thermophysicalProperties so that OpenFOAM finds and reads the required data when the solver is running.

As it is shown below, thermodynamic settings of simulation could be adjusted in the constant/thermophysicalProperties file. Plus, the last two lines specify the address of the files in which the reaction, thermodynamics, and transport data of the species are stored.

constant/thermophysicalProperties

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

inertSpecie N2;

chemistryReader foamChemistryReader;
foamChemistryFile "$FOAM_CASE/constant/reactionsGRI";
foamChemistryThermoFile "$FOAM_CASE/constant/thermo.compressibleGasGRI";
```

3.3. Chemistry Properties

Settings for obtaining reactions rate from kinetic data are located in constant/chemistryProperties. As it is shown, solver and the numerical method of the solver are selected in the `chemistryType` dictionary. By changing value of chemistry to `off`, no reaction takes place in the simulation. Important species could be defined, plus that the initial chemical time step for sub-stepping. Because of the high reaction rate for combustion reactions, it is vital to specify `initialChemicalTimeStep` small enough to prevent possible divergence during simulaiton. For reacting mixtures, the possible cominations of `solver` and `method` are shown in Table 3.

constant/chemistryProperties

```
chemistryType
{
    solver          ode;
    method          TDAC;
}

chemistry       on;

importantSpecies
{
    CO2             ;
    H2O             ;
    CH4             ;
    O2              ;
}

initialChemicalTimeStep 1e-07;

odeCoeffs
{
    solver          seulex;
    absTol          1e-08;
    relTol          0.1;
}

reduction
{
    active          on;
    log             on;
    tolerance       0.0001;
    method          DAC;
    initialSet
    {
        CO              ;
        CH4             ;
        HO2             ;
    }
    automaticSIS    off;
    fuelSpecies
    {
        CH4             1;
    }
}

tabulation
{
    active          on;
    log             on;
    printProportion off;
    printNumRetrieve off;
    tolerance       0.003;
    method          ISAT;
    scaleFactor
    {
        otherSpecies    1;
        Temperature     10000;
        Pressure        1e+15;
        deltaT          1;
    }
    maxNLeafs       5000;
    chPMaxLifeTime  1000;
    maxGrowth       100;
    checkEntireTreeInterval 500;
    maxDepthFactor  2;
    minBalanceThreshold 30;
    MRURetrieve     false;
    maxMRUSize      0;
    growPoints      true;
    maxNumNewDim    10;
}
```

Table 3: Available solvers and methods for reacting mixtures

Solver	Method
ode	standard
	TDAC
EulerImplicit	standard
	TDAC
noChemistrySolver	standard
	TDAC

3.4. Turbulence properties

The turbulence properties of the system are defined in constant/turbulenceProperties. This simulaiton uses on k-epsilon model.

constant/turbulenceProperties

```
simulationType   RAS;

RAS
{
    RASModel         kEpsilon;

    turbulence       on;
    printCoeffs      on;
}
```

3.5. Boundary and initial conditions

Initial and boundary conditions of the filed variables are defined in 0 folder. Here, we give a brief overview of some of the important boundary conditions.

Note

If you copied files from OpenFoam v1912 installation folder, you can find the definition of all the fields in **0.org** folder. Just execute the following command in the case directory:

```
> cp -r 0.orig/  0
```

3.5.1. Velocity field

In the file 0/U, boundary conditions of the velocity field are specified. `noSlip` condition is applied for `wallTube` and `wallOutside` patchs. Fixed value (0, 0, 49.6) is specified for `inletCH4`, (0, 0, 0.9) for `inletAir`, and (0 0 11.4) for `inletPilot`. For `frontAndBack_pos` and `frontAndBack_neg`, boundary condition of `wedge` is applied because of the axisymmetric nature of the geometry. Air flows before the start of the simulation in the domain; therefore `internalField` is set to (0 0 0.9) as the initial condition.

0/U

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

internalField   uniform (0 0 0.9);

boundaryField
{
    wallTube
    {
        type            noSlip;
    }

    outlet
    {
        type            pressureInletOutletVelocity;
        value           $internalField;
    }

    inletPilot
    {
        type            fixedValue;
        value           uniform (0 0 11.4);
    }

    inletAir
    {
        type            fixedValue;
        value           uniform (0 0 0.9);
    }

    wallOutside
    {
        type            zeroGradient;
    }

    inletCH4
    {
        type            fixedValue;
        value           uniform (0 0 49.6);
    }
    frontAndBack_pos
    {
        type            wedge;
    }

    frontAndBack_neg
    {
        type            wedge;
    }
}
```

3.5.2. Temperature field

In the 0/T.orig file, boundary conditions for temperature are set. For all of the walls and outlet, `zeroGradient` condition has been set, while for other fields, a `fixedValue` is set based on the information given in Table 2. For `frontAndBack_pos` and `frontAndBack_neg` fields, `wedge` type is specified similar to velocity field. A fixed value of 300 K is set as the `internalField` the initial temperature.

0/T.orig *

```
dimensions      [0 0 0 1 0 0 0];

internalField   uniform 300;

boundaryField
{
    inletCH4
    {
        type            fixedValue;
        value           uniform 294;
    }

    wallOutside
    {
        type            zeroGradient;
    }

    wallTube
    {
        type            zeroGradient;
    }

    inletPilot
    {
        type            fixedValue;
        value           uniform 1880;
    }

    inletAir
    {
        type            fixedValue;
        value           uniform 291;
    }

    outlet
    {
        type            zeroGradient;
    }

    frontAndBack_pos
    {
        type            wedge;
    }

    frontAndBack_neg
    {
        type            wedge;
    }
}
```

* For OpenFoam v1912, it is 0/T.

3.5.3. Creating initial main jet

`setFields` utility could be used to change the initial condition of a variables in a selected sub-domain. In system/setFieldsDict, the value of the fileds `T`, `N2`, `O2` and `CH4` are set to 300, 0.77, 0.23 and 0 for all the cells (whole domain), respectively. Then, the cells palced in a box whose boundaries are defined in `boxToCell` dictionray are selected and the value of the fields `N2`, `O2` and `CH4` are changed to 0.1561, 0.1966 and 0.6473 in these selected cells, respectively. This box encompases the main jet tube.

system/setFieldsDict

```
defaultFieldValues
(
    volScalarFieldValue T 300
    volScalarFieldValue N2 0.77
    volScalarFieldValue O2 0.23
    volScalarFieldValue CH4 0
);

regions
(
    boxToCell
    {
        box (0 -10 -100) (0.0036 10 0);
        fieldValues
        (
            volScalarFieldValue CH4 0.1561
            volScalarFieldValue O2 0.1966
            volScalarFieldValue N2 0.6473
        );
        }
);
```

4. Running the simulation

You need to execute the following commands in the case directory:

```
> chemkinToFoam chemkin/grimech30.dat chemkin/thermo30.dat chemkin/transportProperties
constant/reactionsGRI constant/thermo.compressibleGasGRI
```

```
> blockMesh
```

```
> setFields
```

Note

If you copied files from OpenFoam v1912 installation folder, you must execute the following command (if have not done so up to here) before `setFields` command:

```
> cp -r 0.orig/  0
```

Up to here, you are ready to run the solver (reactingFoam) and perform the main simulation. The main simulation consists of two steps: the first step is to allow the flow fields to develop before the start of combustion, and the second step is to run the solver with combustion to obtain a steady simulation.

First step:

You need to set the start and end time (`endTime`) of simulation to 0 and 1500 in system/controlDict. Change the write interval (`writeInterval`) to 1500, too. Then turn off the chemistry by setting `chemistry` to `off` in constant/chemisteryProperties. The purpose of these changes is to allow the flow field to develop before starting the main simulation. Then run the application by the following command.

```
> reactingFoam
```

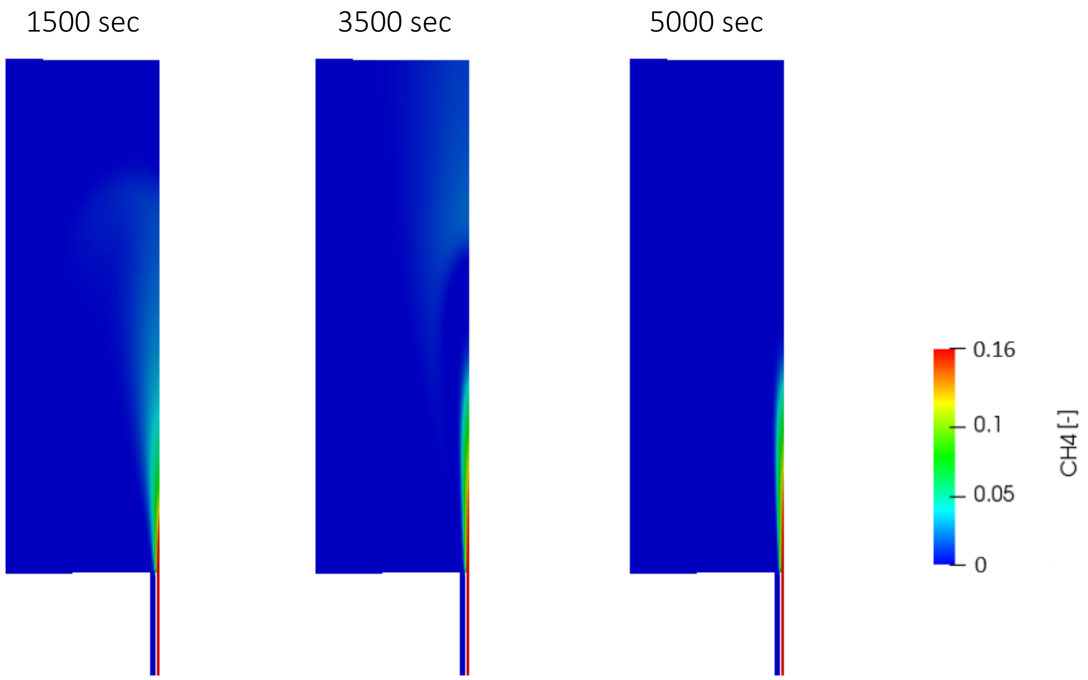
Second step:

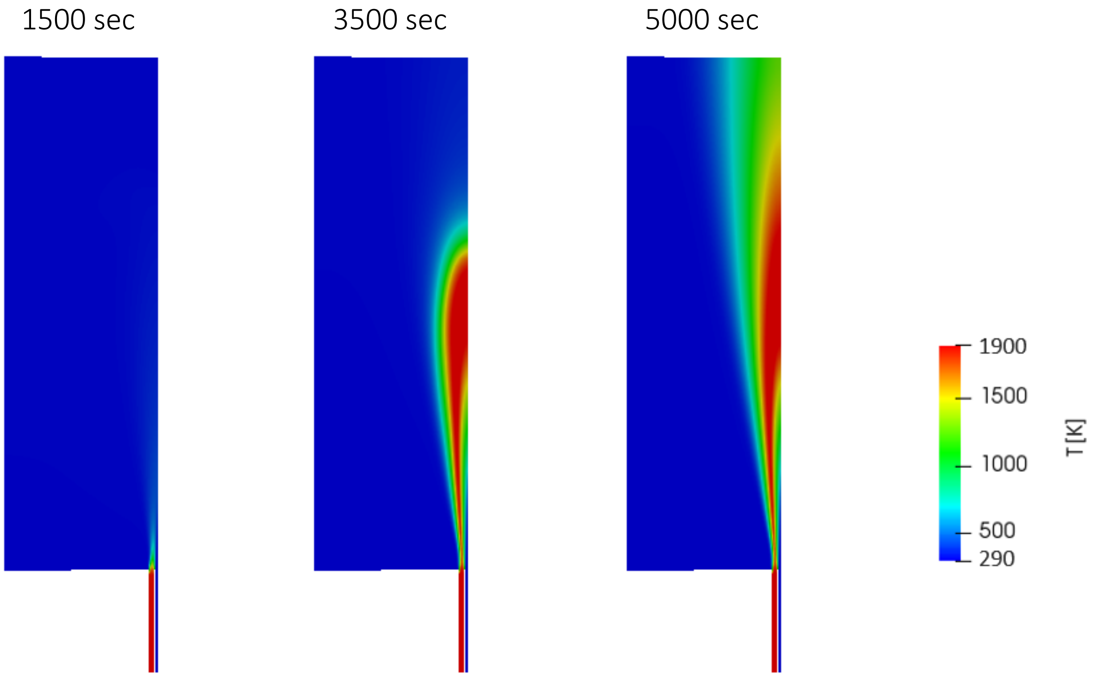
After the end of the simulation in the first step, turn the chemistry on in constant/chemisteryProperties file. Set `startTime` and `endTime` to 1500 and 5000 in system/controlDict, respectively. Set the `writeInterval` to 100 and run the solver again by the following command:

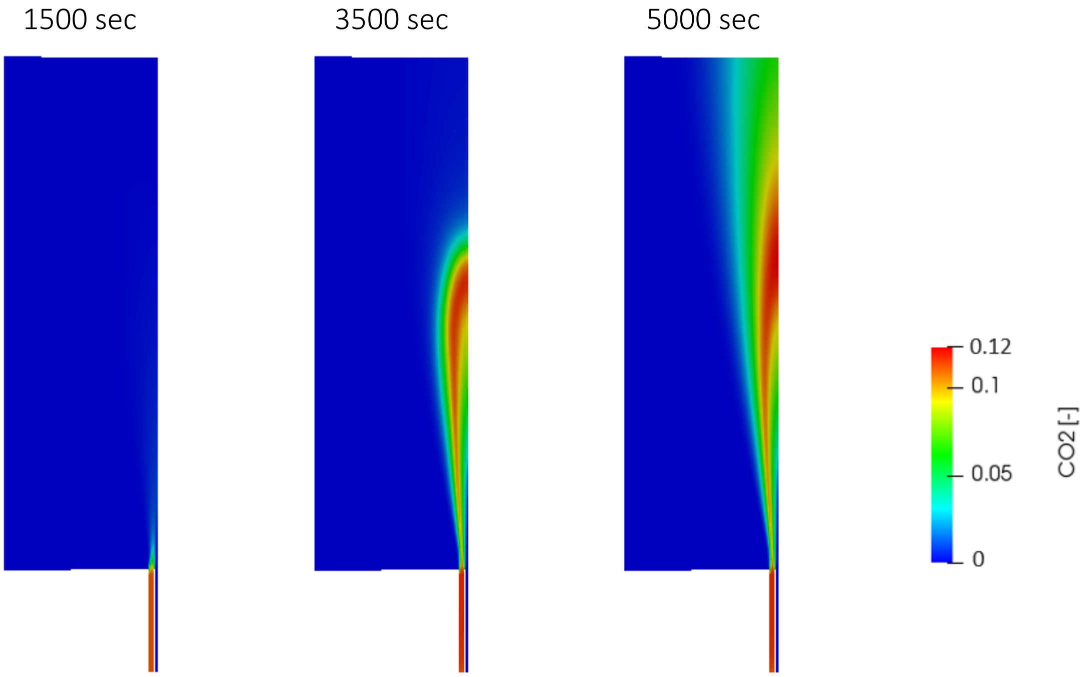
```
> reactingFoam
```

5. Results

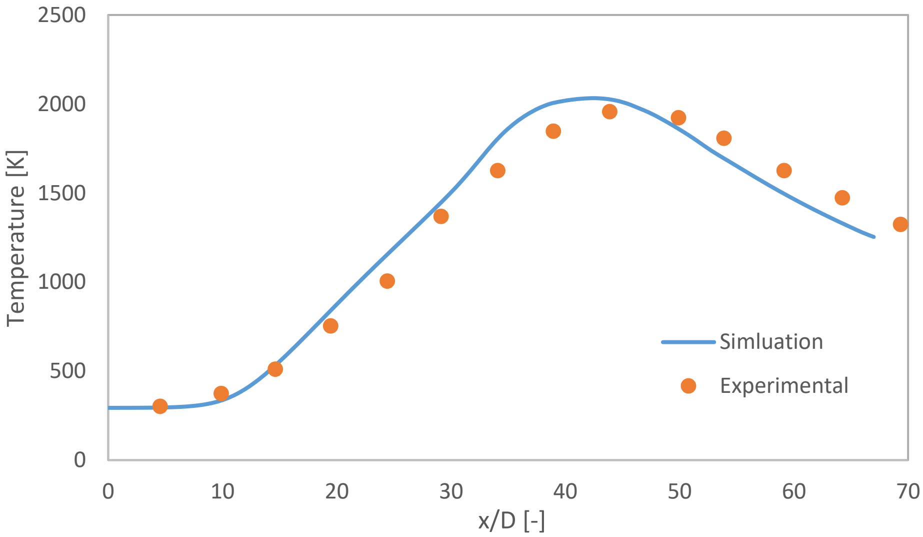
Some snapshots of the burner simulation are presented here.

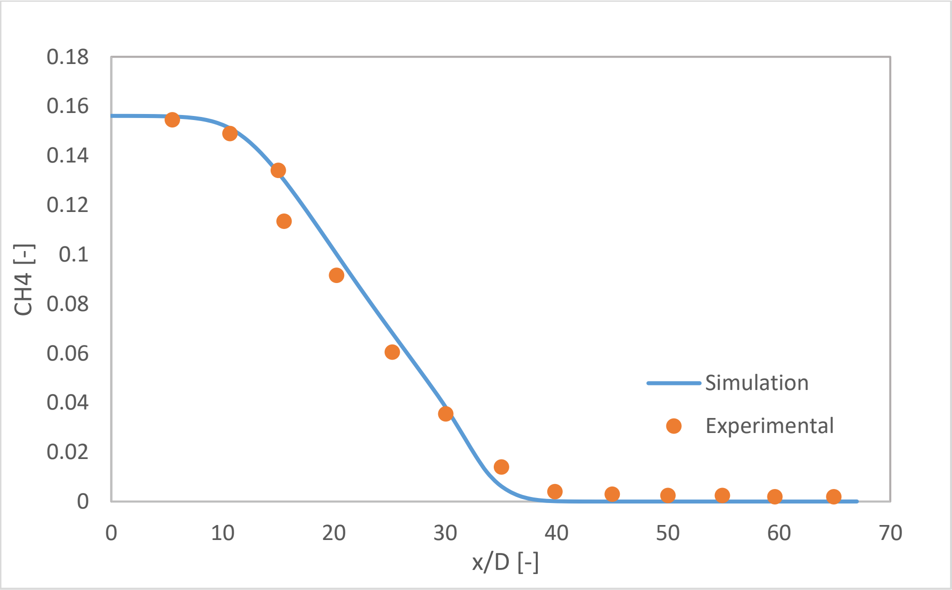



1500 sec
3500 sec
5000 sec
1900
1500
1000
500
290
T[K]

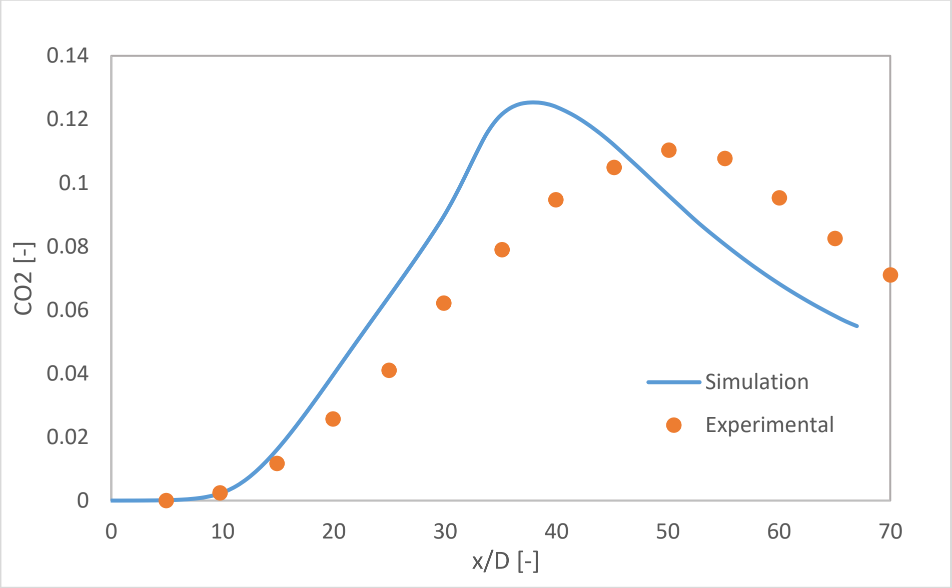

1500 sec
3500 sec
5000 sec
0.12
0.1
0.05
0
CO2 [-]

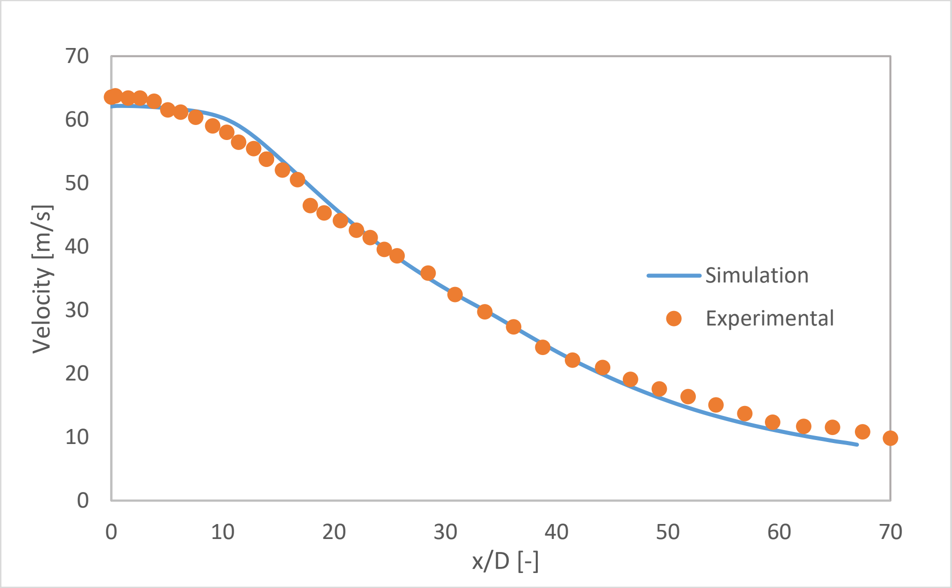
The following graphs compare simulation and experimental results. Data are gathered from central axis of the flame. x is the distance from the center of the main jet outlet and D is diameter of the main jet tube (7.2 mm). The experimental data are available at the TNF Workshop website[4].





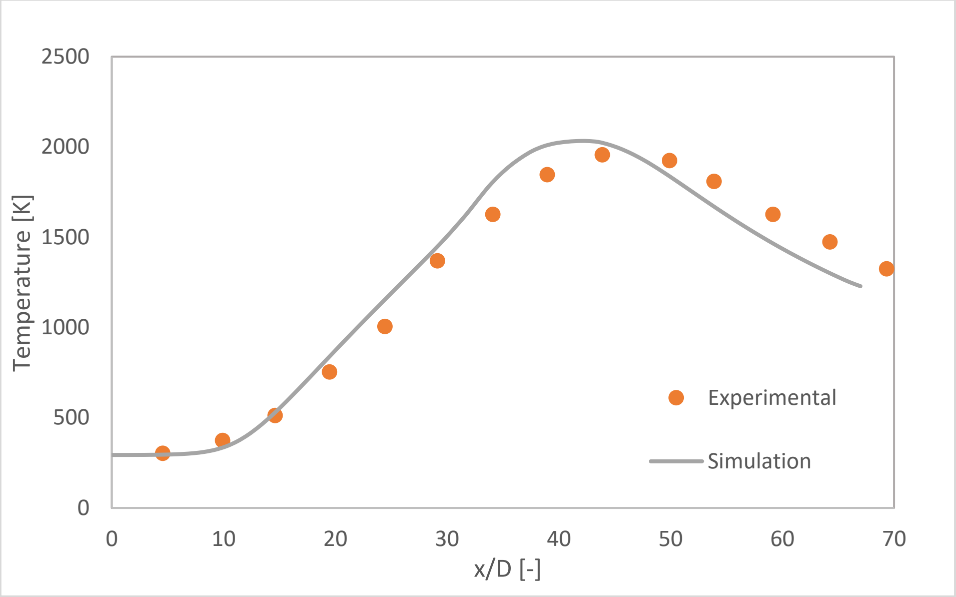
[4] https://tnfworkshop.org/data-archives/pilotedjet/ch4-air/

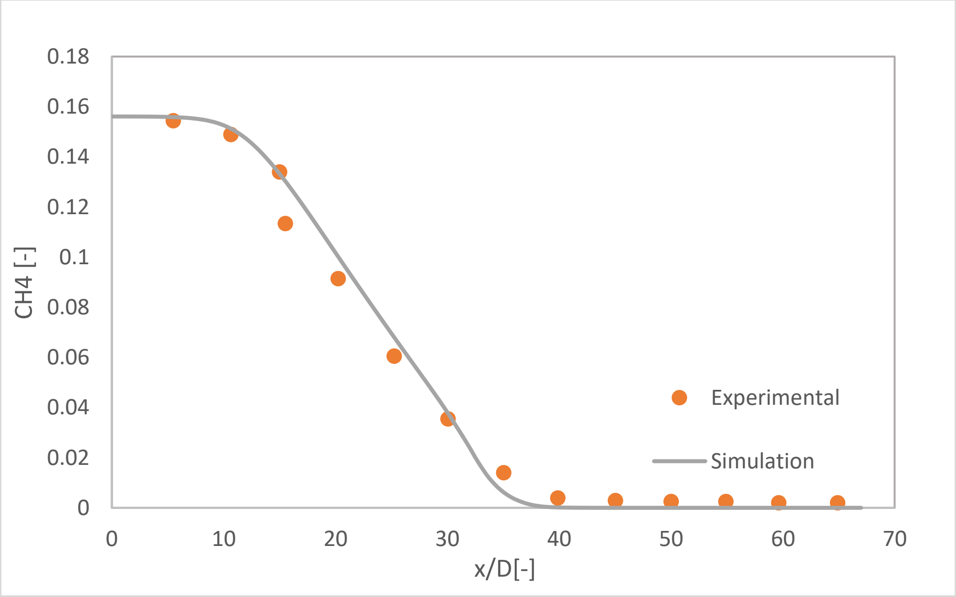


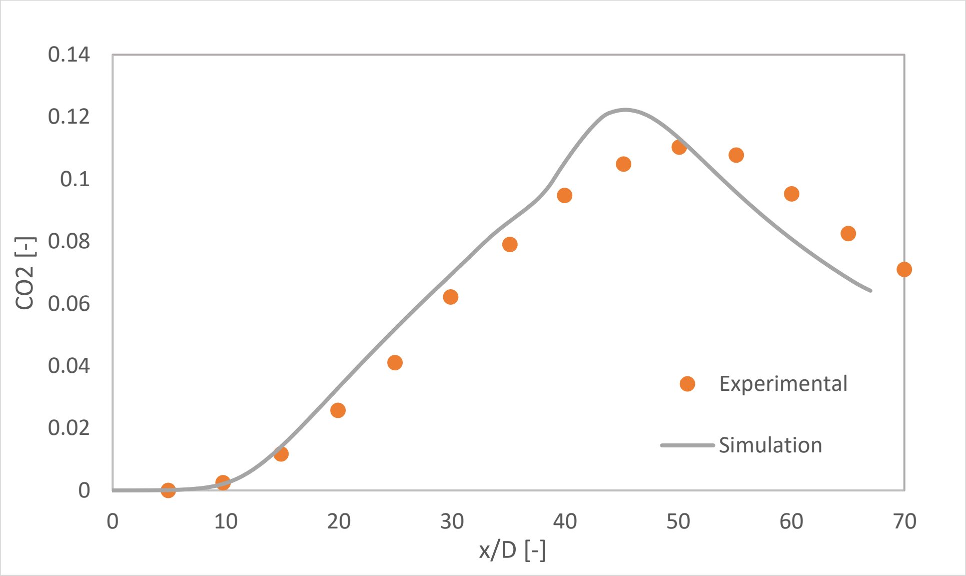


As it's shown in the above graphs, the implemented combustion path mechanism (reduced GRI 3.0) can well predict the profile of various important variables along the flame axis expect for CO_2 concentration. The large deviations between CO_2 mass concentration in simulation and experimental data is due to implementation of reduced cumbusion path mechanism.

The same simulation is done by applying the original GRI 3.0 mechanism and following profiles are obtained. As you see, the simulation can well predict the CO_2 mass concentration too.






0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
CO2 [-]
0
10
20
30
40
50
60
70
x/D [-]
● Experimental
Simulation