2D combustor case - non-premixed combustion

Federico Piscaglia [*]
Dept. of Aerospace Science and Technology (DAER), Politecnico di Milano, Italy

Abstract.

Lab handout for study of non-premixed combustion using a simplified chemistry model. Gaseous fuel (methane, CH_4) is injected through secondary inlets. Ignition is triggered by high-temperature of the inlet air.

1 Learning outcome

The software used is the open-source CFD software OpenFOAM®-8 by the OpenFOAM Foundation. In this Lab you will learn how to:

- Set up a solver including chemical reactions
- run a non-premixed combustion case

2 Introduction

2.1 Statement of the problem

In the combustor case, gaseous fuel (CH_4) is injected through the secondary inlet. Chemistry (reaction rate and heat released) is solved by a simplified one-reaction model. Ignition is triggered by the temperature level of the air at the inlet. The oxidizer (air) and the fuel enters into the domain through separate inlet patches, a diffusion (non-premixed) flame forms. Non-premixed combustion occurs in the mixing region according to the selected combustion model. Initial conditions are represented in Fig. 1 and Tab. 2.1 and obtained by running the non-reacting flow for 0.5 s.

Field	inlet	fuel_inlet
U	10	15
T	1000	350
O_2	0.234	0
N_2	0.766	0
CH_4	0	1

Table 1: Boundary conditions in the reacting flow problem

[*]Tel. (+39) 02 2399 8620, E-mail: federico.piscaglia@polimi.it

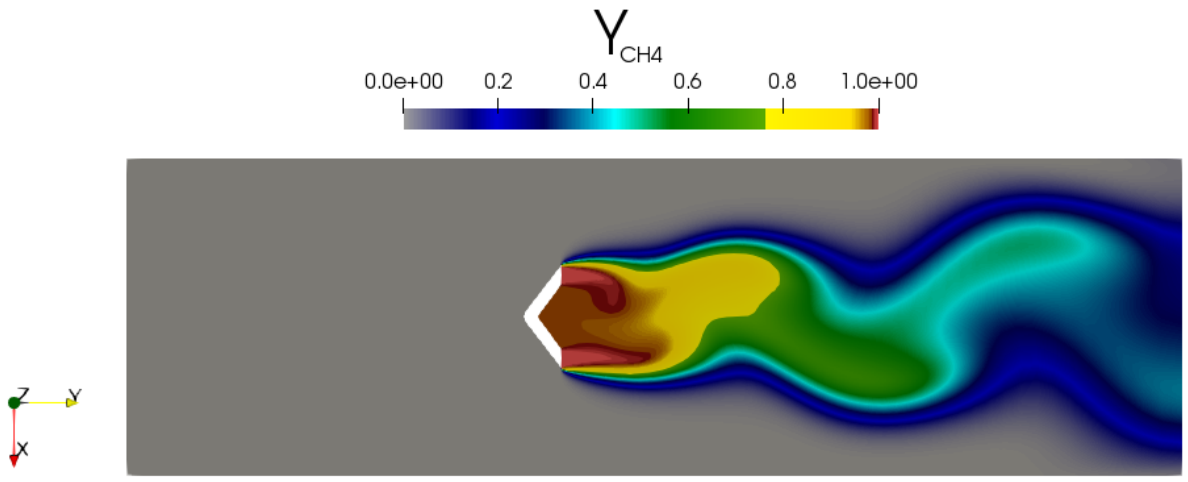


Figure 1: initial conditions

2.2 Outline of the procedure

1. Copy the template
2. Set initial conditions
3. Deactivate lagrangian tracking
4. Select the combustion model
5. Set up the chemistry
6. Setup fluid-dynamics solver
7. Run the case

3 Models for Non-Premixed Combustion in OpenFOAM

Non-premixed combustion modeling in OpenFOAM consists essentially in computing two source terms:

- The reaction rate of the specie $\dot{\omega}_i$
- The released/absorbed heat of reaction $\dot{Q}$

The above terms are the outcome of two mutually interacting phenomena: turbulent mixing and chemical kinetics. Depending on the so called Damköler number:

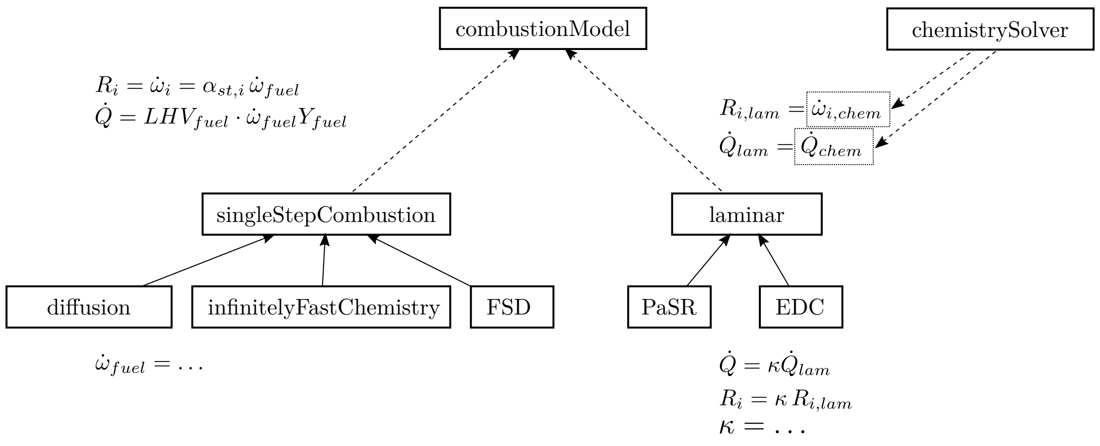
$$\mathrm{Da} = \frac{\text{reaction rate}}{\text{convective mass transport rate}}$$

Two conditions are commonly identified:

- $Da \gg 1$: the chemical reactions are supposed to be much faster than turbulent phenomena, so the reaction rate can be controlled only by mixing (infinitely fast chemistry).

- $Da \ll 1$: if mixing is much faster than chemistry, turbulence does not influence the reaction rate and the terms listed above are determined only by kinetics.

In between those extrema, we have models that accounts the interaction between turbuelence (mixing) and chemistry.



combustionModel
chemistrySolver
$R_i = \dot{\omega}_i = \alpha_{st,i}\,\dot{\omega}_{fuel}$
$\dot{Q} = LHV_{fuel} \cdot \dot{\omega}_{fuel} Y_{fuel}$
$R_{i,lam} = \dot{\omega}_{i,chem}$
$\dot{Q}_{lam} = \dot{Q}_{chem}$
singleStepCombustion
laminar
diffusion
infinitelyFastChemistry
FSD
PaSR
EDC
$\dot{\omega}_{fuel} = \dots$
$\dot{Q} = \kappa \dot{Q}_{lam}$
$R_i = \kappa\, R_{i,lam}$
$\kappa = \dots$

Figure 2: combustion models

3.1 Models with infinitely fast chemistry

In Fig. 2 the aforementioned categories are represented. Models deriving from `singleStepCombustion` do consider infinitely fast chemistry. Combustion is represented by the global reaction only (i.e., there are no intermediate species), like the methane stoichiometric oxidation in air:

$$CH_4 + 2\,O_2 + 7.53\,N_2 \longrightarrow CO_2 + 2\,H_2O + 7.53\,N_2$$

The reaction occurs between the "fuel" specie, that is specified in the input files, and the oxidizer, that defaults to O_2. Each specie production/consumption rate is given by:

$$R_i = \dot{\omega}_f \alpha_{st,i} \tag{1}$$

where $\alpha_{st,i}$ is the mass ratio in stoichiometric conditions between fuel and specie i and $\dot{\omega}_f$ is the fuel consumption ratio. Released heat is then estimated as:

$$\dot{Q} = \mathrm{LHV}_f \dot{\omega}_f Y_f \tag{2}$$

with LHV_f and Y_f as the fuel lower heating value and mass concentration. Different submodels differ then in the way $\dot{\omega}_f$ is computed.

3.1.1 infinitelyFastChemistry

Base assumption is "mixed is burnt": the reaction rate is given by the consumption of the limiting reactant in the cell:

$$\dot{\omega}_f = \rho \frac{1}{C\ \Delta t} \min \left[Y_f, \frac{Y_{O2}}{\alpha_{st}} \right] \tag{3}$$

3.1.2 diffusion

Diffusion of reactants (fuel and oxidizer) is considered by taking the inner product of the concentration gradients.

$$\dot{\omega}_f = C\,\mu_{eff}\,|\nabla Y_f \cdot \nabla Y_{O2}|\,\mathrm{pos}_0\left(Y_f\right)\mathrm{pos}_0\left(Y_{O2}\right) \tag{4}$$

where μ_{eff} is the effective viscosity computed by the turbulence model. $\mathrm{pos}_0\left(x\right)$ is a function that is equal to 1 if $x \geq 0$ and 0 otherwise.

3.1.3 FSD

Flame Surface Density. It solves transport equation for the flame surface density and the mixture fraction z. Used along with LES modeling. Not considered in this lab.

3.2 Models with finite-rate chemistry

3.2.1 laminar

It does not consider turbulent mixing. Reaction rate and heat of reaction are computed by the chemistry solver:

$$R_i = \dot{\omega}_{i,chem} \tag{5}$$
$$\dot{Q} = \dot{Q}_{chem} \tag{6}$$

3.2.2 PaSR

(Partially Stirred Reactor). The mixing time scale is computed as:

$$\tau_k = C_{mix}\sqrt{\frac{\mu_{eff}}{\rho\varepsilon}} \tag{7}$$

and it is used to compute the reaction limiter κ:

$$\kappa = \begin{cases} \dfrac{\tau_{chem}}{\tau_{chem}+\tau_k} & \text{if} \quad \tau_k > 0 \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

Then:

$$R_i = \kappa\,\dot{\omega}_{i,chem} \tag{9}$$
$$\dot{Q} = \kappa\,\dot{Q}_{chem} \tag{10}$$

3.2.3 EDC

Eddy Dissipation Concept. The limiter on reaction rate due to mixing timescales κ (Eq. (11)) is evaluated as a function of the scalar dissipation rate γ_L (Eq. (12)):

$$\kappa = \max\left[\min\left(\frac{\gamma_L^{e1}}{1-\gamma_L^{e2}}, 1\right), 0\right] \tag{11}$$

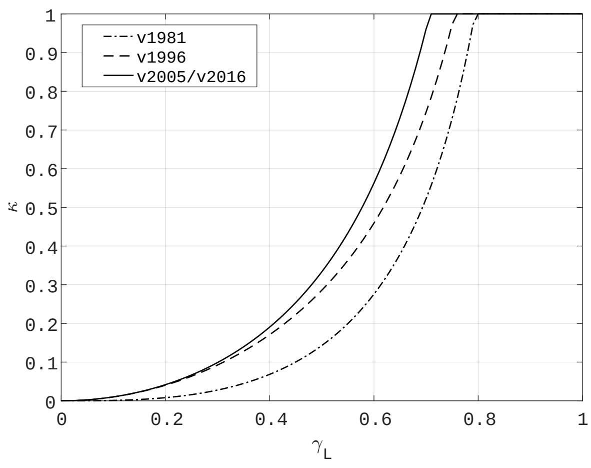


Figure 3: EDC model constant κ

$$\gamma_L = C_\gamma \left(\frac{\nu\varepsilon}{k^2}\right)^{0.25} \tag{12}$$

Exponents $e1$ and $e2$ in Eq. (11) and coefficient C_γ in Eq. (12) depend on the model version. Four options are available and selectable via the “version” keyword in the model subdictionary:

version	$e1$	$e2$	C_γ
v1981	3	3	2.1377
v1996	2	3	2.1377
v2005	2	2	2.1377
v2016 **(default)**	2	2	$\max\left[\min\left(0.5\sqrt{\mathrm{Da}(\mathrm{Re}_t+1)}, 5\right), 0.4082\right]$

The curve $\kappa(\gamma_l)$ is represented in Fig. 3.

Per-specie reaction rate and the released heat are evaluated as in the previous model (Eqs. (9) and (10))

$$R_i = \kappa\,\dot{\omega}_{i,chem} \tag{9}$$
$$\dot{Q} = \kappa\,\dot{Q}_{chem} \tag{10}$$

3.3 Specifying chemical scheme

Chemical reactions to be solved in case of finite-rate chemistry models can be specified using two formats: “chemkin” or “foamChemistryReader”.

3.3.1 Chemkin

Definitions are included in two files in the “chemkin” subfolder:

- `chemkin.inp` containing the reactions and their kinetic coefficients

- `therm.dat` containining thermodynamic properties for all elements and species.

Chemkin files have a fixed line format and are hardly human-readable. More information can be found at http://akrmys.com/public/chemkin/CKm_inp.html.en.

To use the chemkin reader the following lines have to be put in the "thermophysicalProperties" dictionary:

```
chemistryReader chemkinReader;

CHEMKINFile      "$FOAM_CASE/chemkin/chem.inp2";
CHEMKINThermoFile "$FOAM_CASE/chemkin/therm.dat";
CHEMKINTransportFile "$FOAM_CASE/chemkin/transportProperties";

newFormat        yes;
```

3.3.2 OpenFOAM chemistry reader

Thermodynamic properties and reactions can also be specified in the familiar OpenFOAM dictionary format. OpenFOAM®-8 is able to read this familiar format in "thermophysicalProperties" provided you insert the transformed (from CHEMKIN to OpenFOAM®) properties:

```
  #include "$FOAM_CASE/constant/thermo.mixture";
```

File: reactions

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

}
```

File: thermo.mixture

```
species 5 (O2 H2O CO2 N2 CH4);

CH4
{
    specie
    {
        molWeight       16.0428;
    }
    thermodynamics
    {
        Tlow            200;
        Thigh           6000;
        Tcommon         1000;
        highCpCoeffs    ( 1.63543 0.0100844 -3.36924e-06 5.34973e-10 -3.15528e-14 -10005.6 9.9937 );
    {
        As              1.67212e-06;
        Ts              170.672;
    }
}

O2
{
  // ...
}

// all species and elements...
```

3.3.3 Conversion chemkin to OpenFOAM

Since the CHEMKIN format is a widely accepted standard, the mechanism input files are often obtained from a third party. To have them in foamChemistryReader format, it is possible to exploit the "chemkinToFoam" converter.

```
chemkinToFoam <CK file> <CK thermodynamics file> <CK transport file> \
              <OF chemistry file> <OF thermodynamics file>
```

ex

```
$ chemkinToFoam chemkin/chem.inp chemkin/therm.dat chemkin/transportProperties \
                constant/reactions constant/thermo.mixture
```

3.4 Chemistry solver

The chemistry solver has the task of solving the ODE system defining the kinetic mechanism involved in combustion. In a real combustion problem he global reaction can be decomposed into several elementary reactions involving elements, radicals and intermediate species:

$$\sum_{i=1}^{N} \nu'_{ij} M_i = \sum_{i=1}^{N} \nu''_{ij} M_i \tag{13}$$

where M_i is the chemical symbol for i, ν'_{ij} and ν''_{ij} the stoichiometric coefficients.

The rate of each reaction is written as:

$$\dot{\omega}_i = k_i Y_i Y_j \tag{14}$$

where k_i is the constant for the i-th reaction, that is usually modeled using an Arrhenius-like expression

$$k_i = A_i T^{n_i} \exp\left(-\frac{T_{a,i}}{T}\right) \tag{15}$$

The solution of Eqs. (13), (14) and (15) requires an ODE solver like Implicit Euler, Runge-Kutta or more advanced schemes.

The stiffness of the resulting system requires very small timestep. To avoid slowing down the entire simulation, subcycling is employed. Chemistry is solved using a δt that is a fraction of the global one, and multiple chemistry steps are computed for each fluid-dynamic Δt. ODE solver properties are set in the file "constant/chemistryProperties".

4 Setup the case

4.1 Prepare

1. Copy the setup of the lagrangian case:

   ```
   cp -r combustor2D-spray combustor2D-PaSR
   ```

2. Extract initial and boundary conditions archive (see tab. 2.1):

   ```
   combustor2D-PaSR$ tar xzf initialConditions-cmb.tar.gz
   ```

4.2 Deactivate lagrangian tracking

In file "constant/sprayCloudProperties":

```
    active    no;
```

In the subdictionary defining the injector properties

```
  submodels
  {
    injectionModels
    {
      model1
      {
        //...
        massTotal    0;
        massFlowRate 0;
        // ...
      }
      //...
    }
    // ...
  }
```

4.3 Select combustion model

Mind that the call of the combustion model has been simplified compared to previous OpenFOAM® versions, now the simple definition of the model (*ex:* `PaSR`) is enough and advised.

```
combustionModel PaSR;
active          yes;
semiImplicit no;

PaSRCoeffs
{
    Cmix                1.0;
}
```

4.4 Set up chemistry

1. In order to select OpenFOAM® chemistry reader in "constant/thermophysicalProperties" it is now sufficient to:

   ```
   #include "$FOAM_CASE/constant/thermo.mixture";
   ```

2. This means that the user must have the converted OpenFOAM® format files based on the chemkin ones:

   ```
   combustor2D-PaSR$ cd chemkin
   chemkin$ chemkinToFoam chem.inp therm.dat transportProperties
                                  ../constant/reactions ../constant/thermo.mixture
   ```

3. In "constant/thermo.mixture" add the OpenFOAM® header (optional), plus add CH_4 in the reacting mixture (and delete C_7H_{16})

   ```
   // header,

   species 5 (CH4 O2 N2 CO2 H2O)

   CH4
   {
       specie
       {
           molWeight       16.0428;
       }
       thermodynamics
       {
           Tlow            200;
           Thigh           6000;
           Tcommon         1000;
           highCpCoeffs    ( 1.63543 0.0100844 -3.36924e-06 5.34973e-10 -3.15528e-14 -10005.6 9.9937 );
           lowCpCoeffs     ( 5.14988 -0.013671 4.91801e-05 -4.84744e-08 1.66694e-11 -10246.6 -4.64132 );
       }
       transport
       {
           As              1.67212e-06;
           Ts              170.672;
       }
       elements
       {
           C               1;
           H               4;
       }
   }
   // O2, N2, CO2, H2O
   ```

4. Add combustion reaction in "constant/reactions" and delete the "elements" list

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

5. Copy the chemistry solver file

   ```
   $ cp -r $FOAM_TUTORIALS/combustion/chemFoam/h2/constant/chemistryProperties constant/.
   ```

6. Set up chemistry solver. In file “constant/chemistryProperties”

```
chemistryType
{
    chemistrySolver   EulerImplicit;
    chemistryThermo   psi;
}

chemistry on;

initialChemicalTimeStep 1e-08;

EulerImplicitCoeffs
{
    cTauChem        1;
    equilibriumRateLimiter on;
}

#include "$FOAM_CASE/constant/reactions";
```

4.5 Set up fluid-dynamic solver

In “system/fvSolution”

- Use a low-CFL PIMPLE setup:
  - 2 inner correctors
  - 5 outer correctors
  - turbulence solved on all iters (**turbulenceOnFinalIterOnly false;**)
  - consistent
  - no residual control
- Select high relaxation factors:
  - $\alpha_p = 1$
  - $\alpha_\rho = 0.9$
  - $\alpha_{U,Ufinal} = 0.9$
  - $\alpha_{h,hFinal} = 0.9$
  - $\alpha_{k,\varepsilon} = 0.9$

In “system/controlDict”

- end time = 0.1 s
- $\Delta t = 5 \cdot 10^{-7}$ s
- write interval: 0.001 s
- $\text{CFL}_{max} = 0.9$

5 Run the solver

```
$ reactingFoam > log.reactingFoam &
```

6 Hands-on

Run with infinitely-fast chemistry and compare results:

1. In “constant/chemistryProperties” deactivate the chemistry solver
   chemistry off;
2. In “constant/combustionProperties” use:

```
combustionModel infinitelyFastChemistry;

infinitelyFastChemistryCoeffs
{
    C                 1.0;
    semiImplicit      yes;
}
```

3. Since this model does not “see” chemistryProperties, then the reaction must be added in **“constant/combustionProperties”**. Therefore, you also have to copy the methane reaction:

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

4. In “constant/thermophysicalProperties” add fuel in the main of the file:

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

inertSpecie     N2;

#include "$FOAM_CASE/constant/thermo.mixture";

fuel CH4;

liquids
{}

solids
{}
```