2D Combustor case – liquid fuel injection

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

Abstract.

Lab handout for study of liquid fuel injection inside a combustor-like geometry using a particle tracking approach. Topics covered: injector models, particle properties, lagrangian submodels (impingement, secondary breakup, evaporaton, collision). No combustion nor chemical reactions.

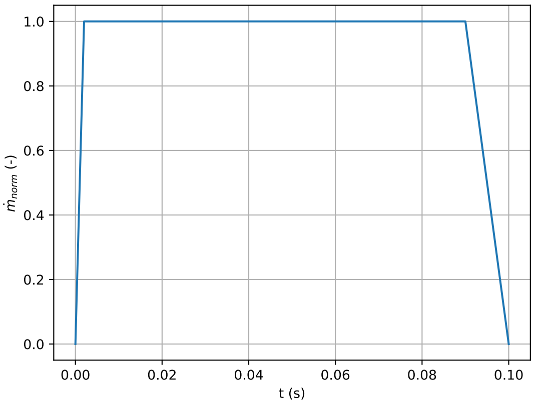
1 Introduction

Problem: In the combustor case, inject fuel and study fuel transport at low temperature. No combustion is to be simulated.

Data of injection are:

- Fuel: n-heptane (C7H16)
- Injected mass: 60 μg
- Injector location: $x = 0.119$, $y = -0.036$
- Injection orientation: 60° from combustor axis
- Injector diameter: 0.19 mm
- Injector cone angle: 10°

Injection law:



t (s)	$\dot{m}_{\text{norm}}$ $(-)$
0	0
0.001	0.5
0.002	1
0.005	1
0.05	1
0.09	1
0.1	0

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

1.1 Structure of lagrangian class

```
typedef SprayCloud
<
    ReactingCloud
    <
        ThermoCloud
        <
            KinematicCloud
            <
                Cloud
                <
                    basicSprayParcel
                >
            >
        >
    >
    > basicSprayCloud;

typedef SprayParcel
<
    ReactingParcel
    <
        ThermoParcel
        <
            KinematicParcel

            <
                particle
            >
        >
    >
    > basicSprayParcel;
```

1.2 Chemkin file format

```
** specs for therm.dat

| Line | Contents                                                                      | Format     | Column   |
|------+-------------------------------------------------------------------------------+------------+----------|
|    1 | THERMO (or THERMO ALL*)                                                       | Free       | Any      |
|------+-------------------------------------------------------------------------------+------------+----------|
|   2# | Temperature ranges for 2 sets of coefficients: lowest T, common T, and highest T  | 3F10.0     | 1 to 30  |
|------+-------------------------------------------------------------------------------+------------+----------|
|    3 | Species name (must start in column 1)                                         | 18A1       | 1 to 18  |
|      | Date (not used in the code)                                                   | 6A1        | 19 to 24 |
|      | Atomic symbols and formula                                                    | 4(2A1,I3)  | 25 to 44 |
|      | Phase of species (S, L, or G for solid, liquid, or gas, respectively)         | A1         | 45       |
|      | Low temperature                                                               | E10.0      | 46 to 55 |
|      | High temperature                                                              | E10.0      | 56 to 65 |
|      | Common temperature (if needed) (blank for default)                            | E8.0       | 66 to 73 |
|      | Atomic symbols and formula (if needed)                                        | 2A1,I3     | 74 to 78 |
|      | The integer 1                                                                 | I1         | 80       |
|------+-------------------------------------------------------------------------------+------------+----------|
|    4 | Coefficients a1-a5 In Eqs. (19) - (21) for upper temperature interval         | 5(E15.0)   | 1 to 75  |
|      | The integer 2                                                                 |            |          |
|------+-------------------------------------------------------------------------------+------------+----------|
|    5 | Coefficients a6, a7 for upper temperature  interval, and a1, a2, and a3 for lower | 5(E15.0)   | 1 to 75  |
|      | The integer 3                                                                 | I1         | 80       |
|------+-------------------------------------------------------------------------------+------------+----------|
|    6 | Coefficients a4, a5, a6, a7  for lower temperature interval                   | 4(E15.0)   | 1 to 60  |
|      | The integer 4                                                                 | I1         | 80       |

** specs for chem.inp
*** elements
1. The first element line must start with the word ELEMENTS (or ELEM).
2. Element or isotope names are either one- or two-character symbols.
3. An isotope name (i.e., a name not on the periodic chart) must be followed
by its atomic weight (in grams per mole) delimited by slashes.
4. Each element or isotope should be declared only once; however, duplicate
element symbols are ignored.
5. An element or isotope name may appear anywhere on the line.
6. Any number of element or isotope names may appear on a line, and more than
one line may be used.
7. Element or isotope names that appear on the same line must be separated by
at least one blank space.
8. An element or isotope name that begins on one line may not continue to the
next line.
9. Any blank spaces between an element or isotope name and the first slash are
ignored and any blank spaces between slashes and an atomic weight are also
```

```
ignored. However, no blank spaces are allowed within an element name or an
atomic weight.
10. There may be more than one ELEMENT statement.
11. All characters following an exclamation mark are comments.

*** species
1. Species data must start with the word SPECIES (or SPEC).
2. Species names are composed of up to 16-character upper- or lower-case
symbols. The names cannot begin with the characters +, =, or a number; an
ionic species name may end with one or more +'s or -'s.
3. Each species should be declared only once; however, duplicated species
symbols will be ignored.
4. Each species that subsequently appears in a reaction must be declared.
5. A species name may appear anywhere on a line.
6. Any number of species names may appear on a line, and more than one line
may be used.
7. Species named on the same line must be separated by at least one blank
space.
8. A species name that begins on one line may not continue to the next line.
9. There may be more than one SPECIES statement.
10. All characters following an exclamation mark are comments.
```

2 Procedure

1. starting point: compressible simulation, low speed and low temperature at $t = 0.05$s.

   i. `$ cp -r combustor2D-RPF combustor2D-spray`
   ii. `$ cd combustor2D-spray`
   iii. `combustor2D-spray $ mv 0 0.bak`
   iv. `combustor2D-spray $ ./Allclean`
   v. `combustor2D-spray $ tar zxvf initialConditions.tar.gz .`
   vi. `combustor2D-spray $ mv initialConditions 0`

2. copy thermophysical properties from tutorial:

   `$ cp $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/constant/thermophysicalProperties constant/.`

   no modification required

3. copy chemkin folder:

   `\$ cp -r \$FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/chemkin .`

   no modification required

4. create BCs for species: O2, N2, Ydefault:

   i. `$ cp $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/0/N2 0/.`
   ii. adjust `N2` internalField, walls and patches values;
   iii. `$ cp -r 0/N2 0/O2` (mind `zero` and `capital O`);
   iv. adjust `O2` internalField, walls and patches values;
   v. `$ cp -r 0/N2 0/Ydefault`
   vi. adjust `Ydefault` internalField, walls and patches values;

   Note: OF uses mass fractions, while it is customary to express concentrations in gas mixtures as volume (i.e. molar) fractions. To convert from molar (=volume) to mass fractions:

$$
\begin{aligned}
x &= \frac{n_i}{n_{\text{tot}}} && (1)\\
&= \frac{m_i/M_w^i}{m_{\text{tot}}/M_w^{\text{mix}}} && (2)\\
&= \frac{m_i}{m_{\text{tot}}} \cdot \frac{M_w^{\text{mix}}}{M_w^i} && (3)\\
&= Y \cdot \frac{M_w^{\text{mix}}}{M_w^i} && (4)
\end{aligned}
$$

where:

$$M_w^{\text{mix}} = \sum_i x_i M_w^i \tag{5}$$

Hence:

$$Y = x \cdot \frac{M_w^i}{M_w^{\text{mix}}} \tag{6}$$

e.g.: If we consider air to be composed 21% in volume by Oxygen (M_w = 32 kg/kmol and 79% (volume) by Nitrogen (M_w = 28 kg/kmol), and assuming M_w^{air} = 28.84 kg/kmol:

$$Y_{O2} = 0.21 \cdot 32/28.84 = 0.233 \tag{7}$$

$$Y_{N2} = 0.79 \cdot 28/28.84 = 0.767 \tag{8}$$

C7H16, CO2 and H2O are 0 at start of simulation. C7H16 comes from fuel evaporation. CO2 and H2O are combustion products. Ydefault is used for initializing the species vector in case not all species defined in 'reactions' are found. This happens, e.g. at time 0. Indeed Ydefault is not written in results folder later on.

5. copy gravity acceleration

```
cp $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/constant/g constant/
```

no modifiration required

6. copy chemistryProperties and deactivate chemistry

```
cp $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/constant/chemistryProperties constant/.
```

```
chemistry       off;
```

7. copy combustionProperties and deactivate combustion

```
combustionModel none;
```

```
active          no;
```

8. in fvSchemes, add:

```
div(phi,Yi_h)   Gauss upwind;
```

9. in fvSolution, add:

```
 "(Yi|O2|N2|H2O).*"
 {
     solver          PBiCGStab;
     preconditioner  DILU;
     tolerance       1e-6;
     relTol          0;
 }
```

10. in controlDict:

```
endTime 0.05;
```

```
writeInterval 0.001;
```

```
maxCo 15;
```

11. copy sprayCloudProperties and change injection properties

```
cp $FOAM_TUTORIALS/lagrangian/sprayFoam/aachenBomb/constant/sprayCloudProperties constant/.

injectionModels
{
    model1
    {
       type            coneInjection;
```

```
            SOI             0; // start-of-injection
            massTotal       6.0e-5;
            parcelBasisType mass;
            injectionMethod disc;
            flowType        flowRateAndDischarge;
            dInner          0;
            dOuter          1.9e-4;
            duration        0.05;
            position        (0.119 -0.036 5e-4);
            direction       (-0.866 0.5 0);
            parcelsPerSecond 20000000;

            flowRateProfile table
            (
                (0      0)
                (0.001 0.5)
                (0.002 1  )
                (0.005 1 )
                (0.05  1)
                (0.09  1)
                (0.1   0)
            );
        }
        //...
    }
```

12. launch the solver:

    ```
    sprayFoam > log.sprayFoam 2>&1 &
    ```