Tutorial 2: Spouted Bed Combustor

In this tutorial we will model a partial combustor with geometry as shown in Figure 1. Char and air are fed into the combustor through a central tube. A second stream of fluidizing air enters the annular region surrounding the tube. Partial combustion of the char occurs in the combustor. The product gases exit from the top of the combustor. MFIX will be used to determine the characteristics of the gas-solids flow and combustion within the spouted bed combustor.

Simulation parameters:

- char properties:
  - diameter, 1000 μm
  - density, 1 g/cm^3
  - composition: fixed carbon, and ash
- gas properties:
  - compressible
  - composition: oxygen, carbon monoxide, carbon dioxide, nitrogen, and soot
- gas/solids inlet:
  - char feed rate, 18 g/sec
  - char composition (by mass): 60% carbon, 40% ash
  - char temperature, 300 K
  - air feed rate, 10 g/sec
  - air composition (by mass): 80% N_2, 20% O_2
  - air temperature, 300 K
  - void fraction, 0.9
- gas inlet:
  - air feed rate, 116 g/sec
  - air composition (by mass): 80% N_2, 20% O_2
  - air temperature, 300 K
  - void fraction, 1.0
- operating pressure: 30 atm
- average void fraction in reactor: 0.5

Since the incoming char is at a low temperature and it is critical to consider char heat-up, the incoming char will be treated as a second solids phase called 'cold-char.' Conversely, the char already in the combustor is called 'hot-char.' When the ash fraction in the cold chare exceeds 0.9, it is assumed to convert into hot-char. A fast pseudo-reaction is specified to convert the cold-char to hot-char. The ash fraction threshold and rate constant for the pseudo-reaction are specified in `mfix.dat` as constants `C(1)` and `C(2)`.

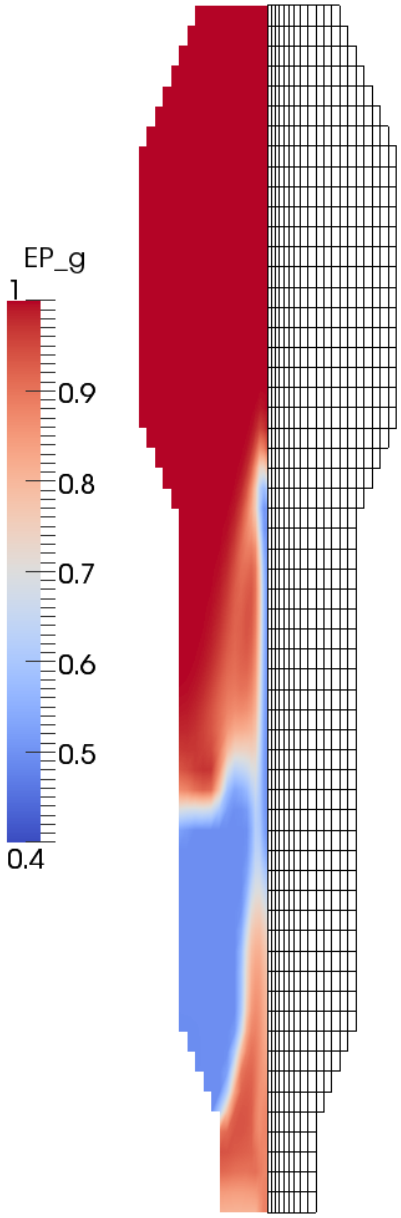

Figure 1. Geometry of the Spouted Bed Combustor

The following chemical reactions are considered:

Carbon combustion: $2C + O_2 \rightarrow 2CO$

CO combustion: $CO + \frac{1}{2}O_2 \rightarrow CO_2$

Char gasification: $C + CO_2 \rightarrow 2CO$

Pseudo-reaction: $\mathrm{Char}_{\mathrm{cold}} \rightarrow \mathrm{Char}_{\mathrm{hot}}$

1 – Provide species information in the data file:

The number, data base name, and an alias for each species are specified in the data file, `mfix.dat`. Note that in this case the same solids phase species database names (`SPECIES_s`) are specified, while as required, the aliases (`SPECIES_ALIAS_s`) are unique. As a result, the two solids phases are comprised of the same species; however any reference to one of the aliases identifies both the species and phase (e.g., , `FC1` is solids phase 1 Fixed Carbon whereas `FC2` is solids phase 2 Fixed Carbon).

```
NMAX_g = 5
SPECIES_g(1) = 'O2'            SPECIES_ALIAS_g(1) = 'O2'
SPECIES_g(2) = 'CO'            SPECIES_ALIAS_g(2) = 'CO'
SPECIES_g(3) = 'CO2'           SPECIES_ALIAS_g(3) = 'CO2'
SPECIES_g(4) = 'N2'            SPECIES_ALIAS_g(4) = 'N2'
SPECIES_g(5) = 'Fixed Carbon'  SPECIES_ALIAS_g(5) = 'Soot'

NMAX_s(1) = 2
SPECIES_s(1,1) = 'Fixed Carbon'  SPECIES_ALIAS_s(1,1) = 'FC1'
SPECIES_s(1,2) = 'Coal Ash'      SPECIES_ALIAS_s(1,2) = 'Ash1'

NMAX_s(2) = 2
SPECIES_s(2,1) = 'Fixed Carbon'  SPECIES_ALIAS_s(2,1) = 'FC2'
SPECIES_s(2,2) = 'Coal Ash'      SPECIES_ALIAS_s(2,2) = 'Ash2'
```

Since pseudo-species 'Fixed Carbon' and 'Coal Ash' are not included in the materials database (`mfix/model/thermochemical/BURCAT.THR`), the thermochemical data must be provided and is specified at the end of the data file.

```
THERMO DATA

Fixed Carbon               WARNING!       0.S   200.000  6000.000  B  12.01100 1
-7.94079328E-01 8.09779756E-03-6.56398654E-06 1.84781958E-09 0.00000000E+00    2
 0.00000000E+00 0.00000000E+00-7.94079328E-01 8.09779756E-03-6.56398654E-06    3
 1.84781958E-09 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    4

Coal Ash                   WARNING!       0.S   200.000  6000.000  B  56.00000 1
 8.70769880E+00 8.45407650E-03 0.00000000E+00 0.00000000E+00 0.00000000E+00    2
 0.00000000E+00 0.00000000E+00 8.70769880E+00 8.45407650E-03 0.00000000E+00    3
 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00    4
```

Additional information on user-defined thermochemcial data is provided in *Section 5.12 Thermochemical Properties* of the MFIX Readme file.

2 – Define chemical reactions in the data file:

Chemical reactions are specified in the data file using the species aliases (see 1).

```
@(RXNS)

  Combustion_s1 { chem_eq = "2FC1 + O2 --> 2CO"}
  Combustion_s2 { chem_eq = "2FC2 + O2 --> 2CO"}

  Char_CO2_s1 { chem_eq = "FC1 + CO2 --> 2CO"}            ! Forward
  Char_CO2_s1r { chem_eq = "2CO + 0.FC1 --> Soot + CO2"}  ! Reverse

  Char_CO2_s2 { chem_eq = "FC2 + CO2 --> 2CO"}            ! Forward
  Char_CO2_s2r { chem_eq = "2CO + 0.FC2 --> Soot + CO2"}  ! Reverse

  CO_Combustion { chem_eq = "CO + 0.5O2 --> CO2" }

  Char_to_Char { chem_eq = 'FC2 --> FC1' }

  Ash_to_Ash { chem_eq = 'Ash2 --> Ash1' }

@(END)
```

The first two constructs, `Combustion_s1 and Combustion_s2`, define char combustion for the two solids phases. Note that reversible reactions must be written as two irreversible reactions. Therefore, the third and forth constructs define solids phase 1 gasification in the forward, `Char_CO2_s1`, and reverse, `Char_CO2_s1r`, directions. Note that the reverse reaction contains fixed carbon with a stoichiometric coefficient of zero, `0.FC1`. As a result, solids phase 1 is treated as a catalyst and is assigned a portion of the heat of reaction. Similarly, solids phase 2 gasification is defined by the fifth and sixth reaction constructs. The seventh reaction construct defines carbon monoxide combustion. The final two constructs define the 'cold-char' to 'hot-char' pseudo-reaction. Additional information on specifying chemical reactions in the data file is provided in the *Chemical Reactions* section of the MFIX Readme.

3 – Define chemical reaction rates `usr_rates.f`:

The reaction rates are specified in the file `usr_rates.f`. Reaction rates are given in g-mol/sec/cm^3 for CGS units and moles/sec/m^3 for SI. The calculated rate values are assigned to the `RATES` array. The array location for a given reaction is identified by the name used to define the reaction in the data file (e.g., `RATES(CO_COMBUSTION)`). Although the species index may be used to access species-specific variables (e.g., molecular weights), to minimize errors and enhance readability, it is recommend that that the species aliases be used instead. For example, in the above sample data file input, carbon monoxide is defined as gas phase species 2 (`SPECIES_g(2) = 'CO'`) with `'CO'` as an alias (`SPECIES_ALIAS_g(2) = 'CO'`). Therefore, the concentration of carbon monoxide in fluid cell `IJK` and with units of moles/cm^3 is given by

```
RO_g(IJK)*X_g(IJK,CO)/MW_g(CO)
```

where `RO_g(IJK)` is the gas density, `X_g(IJK,CO)` is carbon monoxide mass fraction, and `MW_g(CO)` is the molecular weight of carbon monoxide.

4 – Writing User-Define Subroutines:

In addition to `usr_rates.f`, user-defined routines and variables are specified in `usr0.f`, `usr1.f`, `usr_init_namelist.f`, `usrnlst.inc`, and `usr_mod.f`.

`usr0.f` – User-defined input data are checked for error and a constant, `f_EP_A`, is calculated based on the user-defined input. This routine is called before entering the time-loop in MFIX.

`usr1.f` – This routine is used to calculate the Sherwood number for the solids phases. The Sherwood number is later used by `usr_rates.f` in calculating char combustion film layer resistance. This routine is called from the time loop.

`usr_init_namelist.f` – Two user-defined namelist variables, `PAFC` and `PAA`, are initialized as `UNDEFINED`. In MFIX `UNDEFINED` is a parameter used to indicate undefined double precision numbers. By initializing values this way, the user is able to determine if values were supplied in the data file.

`usrnlst.inc` – This file contains user-defined variables, user-defined common blocks, and a namelist section.

`usr_mod.f` – This routine is used for user-defined variables accessible to all program blocks that invoke '`use usr`.'

5 – Additional comments on the data file:

- The energy balance calculations is solved. (`ENERGY_EQ`)
- The species balance calculations are solved for all three phases. (`SPECIES_EQ`)
- User-defined routines `usr0.f` and `usr1.f` are invoked. (`CALL_USR`)
- Two user-defined constants, `C(1)` and `C(2)`, are defined. These constants are used in subroutine `usr_rates` and are assigned names (optional), with which to label them in the `*.OUT` file.
- In the radial direction, a nonuniform grid is used to get a better resolution near the jet. The first four radial grids are 0.5 cm wide and the next four grids expand to approach 1 cm. The last 11 grids have a size of 1 cm. The axial grid size is 2.5 cm. (*see Figure 1*)
- For simplicity, the entire domain is divided into two sections (bed and freeboard) and uniform initial conditions are specified in each section.
  - The initial condition regions encompass non-fluid cells (walls). Although this approach does not follow 'best practice,' it is not problematic as the initial conditions are only applied to the fluid cells.
  - The specified initial conditions are not representative of the flow that will develop. Namely, the specified void fraction is too low for the central jet region. As a result, the solids will clear out of that region, as the MFIX calculations progress.

- There are two flow inlets and one outlet. Note that mass flows are specified instead of the axial velocity component. The composition of gas and solids at the inlet are specified. Only the pressure is specified at the outlet.
- The reactor geometry is specified with 17 obstacles. We have started the obstacles' specification as boundary condition number 20, so that any future additions or deletions of inlets or outlets can be accommodated without having to change the boundary condition indices for the obstacles.
- An internal surface is specified to simulate the central tube at the bottom of the reactor.
- The keyword `nRR=9` is included to write out nine reaction rate fields. The nine fields are populated in the `usr_rates` routine.
- The restart file, `*.RES`, is written every 0.05 seconds.
- `*.SPx` files 1-7, and 10 are written every 0.05 seconds.
- `*.SPx` files 8 and 9 are written every 100.0 seconds.
- Diagnostics from the code are printed out every 100th time step.

5 – Running MFIX:

The FORTRAN files discussed above and the data data file are stored in a run directory. Detained instructions on building the MFIX executable are given in the MFIX Readme. The command `make_mfix` should be ran to create/update the mfix.exe executable if:

- the number, order, or alias of any species is changed in the data file
- the number, order, or name of any chemical reaction is changed in the data file
- any change is made to chemical reaction rates in `usr_rates.f`

The script file `make_mfix` will copy the modified user-defined files in the run directory to the mfix/model directory and copy the MFIX executable to the run directory after compiling.

Before starting a production run, it is strongly advised to first run MFIX with `TSTOP=0`. Correct any input errors until MFIX successfully reads the input data file and exits. MFIX will write all the output files. Review the `*.OUT` file to verify that the inputs were read correctly. Verify that the cell flags are assigned correctly. Note how the wall cells have been assigned to approximate the geometry of the reactor. Also note how the impermeable internal surface has been specified at the east face of the cells `i=3` and `j=2-6`. Once the simulation details are verified, MFIX can be run with `TSTOP` set to the desired run time.