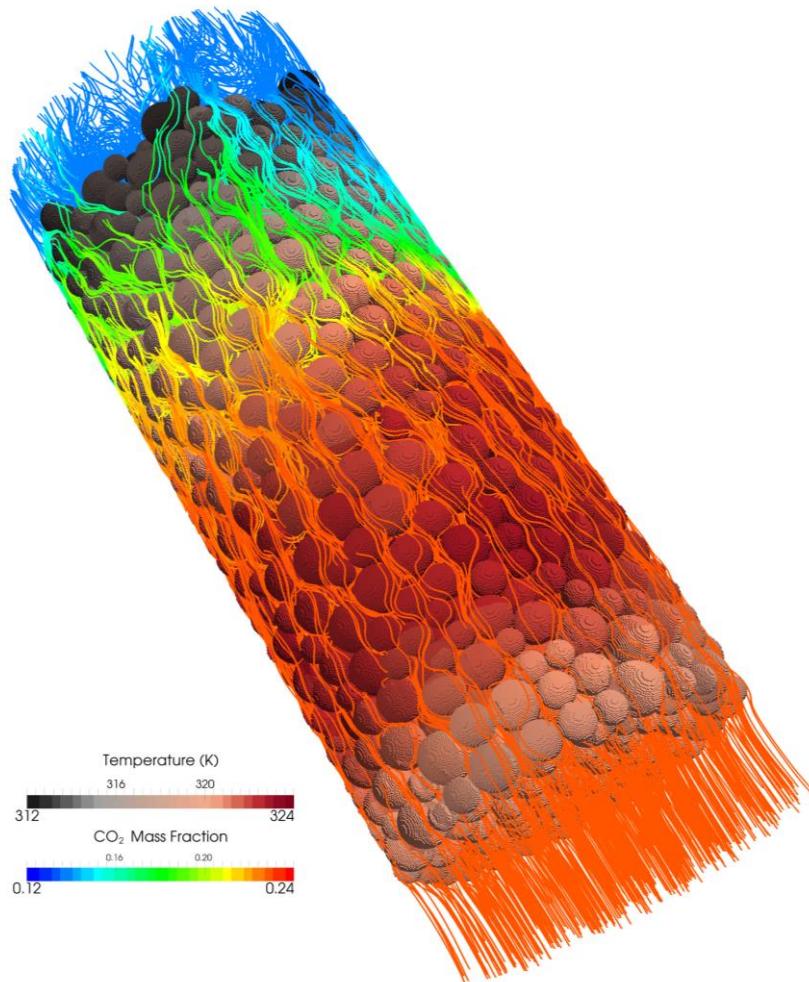




OpenFOAM® Basic Training

Tutorial Ten



3rd edition, Feb. 2015



This offering is not approved or endorsed by ESI® Group, ESI-OpenCFD® or the OpenFOAM® Foundation, the producer of the OpenFOAM® software and owner of the OpenFOAM® trademark.



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Editors and Contributors:

- Bahram Haddadi (TU Wien)
- Christian Jordan (TU Wien)
- Jozsef Nagy (JKU Linz)
- Clemens Gößnitzer (TU Wien)
- Vikram Natarajan (TU Wien)
- Sylvia Zibuschka (TU Wien)
- Michael Harasek (TU Wien)



Vienna University of Technology
Institute of Chemical Engineering



JOHANNES KEPLER
UNIVERSITY LINZ | JKU
Institute of
Polymer Injection Moulding and
Process Automation

VIENNA
SCIENTIFIC
CLUSTER

iPiM

Institute of
Polymer Injection Moulding and
Process Automation

CFD
in collaboration with

Cover picture from:

- Bahram Haddadi, The image presented on the cover page has been prepared using the Vienna Scientific Cluster (VSC).



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

You are free:

- to Share — to copy, distribute and transmit the work
- to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial — You may not use this work for commercial purposes.

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

With the understanding that:

Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;

The author's moral rights;

Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

simpleFoam & scalarTransportFoam – TJunction (Residence Time Distribution)

Simulation

Use the simpleFoam and scalarTransportFoam to simulate the flow through a square cross section T pipe and calculate RTD (Residence Time Distribution) for both inlets using a step function injection:

- Inlet and outlet cross sections: $1 \times 1 \text{ m}^2$
- Gas in the system: air at ambient conditions
- Operating pressure: 10^5 Pa
- Inlet 1: 0.1 m/s
- Inlet 2: 0.2 m/s

Objectives

- Understanding RTD calculation using OpenFOAM®
- Using multiple solver for a simulation

Post processing

Plot the step response function and the RTD curve.

Step by step simulation

Copy tutorial

Copy the following tutorial to your working directory as a base case:

```
~/OpenFOAM/OpenFOAM-2.3.0/tutorials/incompressible/simpleFoam
```

/pitzDaily

0 directory

Update p, U, nut, nuTilda, k and epsilon files with the new boundary conditions, e.g.
U:

constant directory

Edit the `blockMeshDict` in the `polyMesh` directory as following for creating an appropriate geometry.

```

(3 3 1) // 12
(3 0 1) // 13
(4 0 1) // 14
(4 3 1) // 15
(7 3 1) // 16
(7 4 1) // 17
(4 4 1) // 18
(3 4 1) // 19

);

blocks
(
    hex (0 1 2 9 10 11 12 19) (10 30 10) simpleGrading (1 1 1)
    hex (9 2 5 8 19 12 15 18) (10 10 10) simpleGrading (1 1 1)
    hex (8 5 6 7 18 15 16 17) (10 30 10) simpleGrading (1 1 1)
    hex (2 3 4 5 12 13 14 15) (30 10 10) simpleGrading (1 1 1)
);
edges
(
);
patches
(
    patch inlet_one
    (
        (0 10 11 1)
    )
    patch inlet_two
    (
        (7 6 16 17)
    )
    patch outlet
    (
        (4 3 13 14)
    )
    wall walls
    (
        (0 1 2 9)
        (2 5 8 9)
        (5 6 7 8)
        (2 3 4 5)
        (10 19 12 11)
        (19 18 15 12)
        (18 17 16 15)
        (15 14 13 12)
        (0 9 19 10)
        (9 8 18 19)
        (8 7 17 18)
        (2 1 11 12)
        (3 2 12 13)
        (5 4 14 15)
        (6 5 15 16)
    )
);
mergePatchPairs
(
);
// ****

```

Check RASProperties file for the turbulence model (kEpsilon).

```

// ****
RASModel      kEpsilon;
turbulence     on;
printCoeffs   on;

// ****

```

Running simulation

```
>blockMesh
```

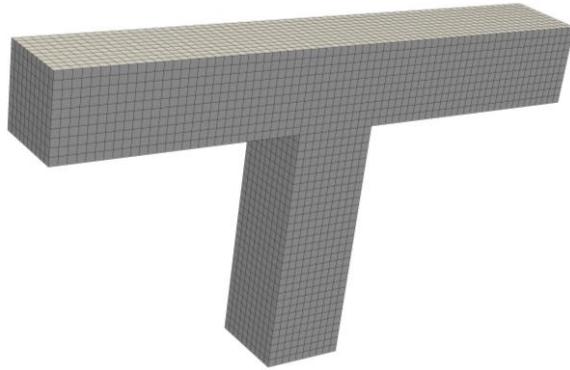


Figure 10.1 mesh created using blockMesh

```
>simpleFoam
```

Wait for simulation to converge. After convergency check the results to be sure the solution is converged (?).

```
>foamToVTK
```

The simulation results are as follows:

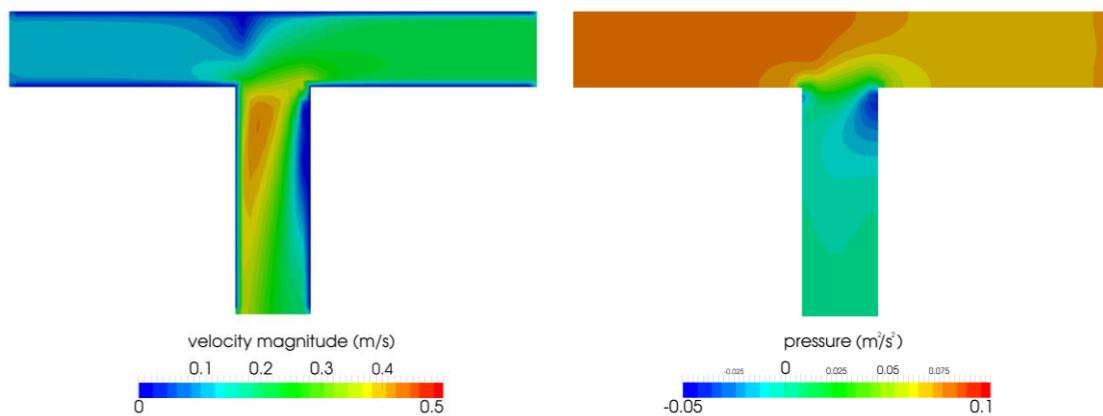


Figure 10.2 Simulation results after convergence (114 iterations)

RTD calculation

Copy tutorial

Copy following tutorial to your working directory:

```
~/OpenFOAM/OpenFOAM-2.3.0/tutorials/basic/scalarTransportFoam
```

```
/pitzDaily
```

0 directory

Delete the U file and replace it with the calculated velocity field from the first part of the tutorial (use the latest time step velocity field from previous part of simulation to

calculate RTD for this geometry). There is no need to modify or change it. The solver will use this field to calculate the scalar transportation.

Update T (T will be used as an inert scalar in this simulation) file boundary conditions to match new simulation boundaries, to calculate RTD of the `inlet_one` set the `internalField` value to 0, T value for `inlet_one` to 1.0 and T value for `inlet_two` to 0.

constant directory

Replace the `blockMeshDict` file with the one from the first part of tutorial.

system directory

In the `controlDict` file change the `endTime` from 0.1 to 120 (approximately two times ideal resistance time) and also `deltaT` from 0.0001 to 0.1 (Courant number approximately 0.4).

Running simulation

```
>blockMesh
>scalarTransportFoam
>foamToVTK
```

Simulation results

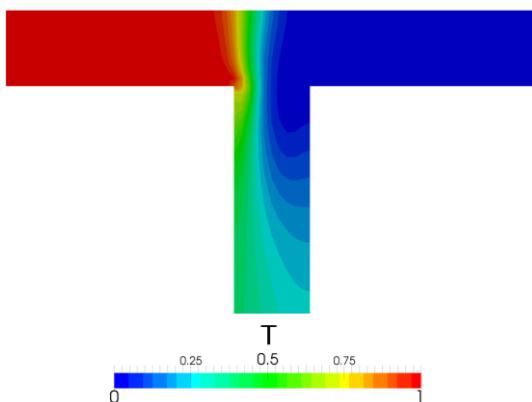


Figure 10.3 Contour plots scalar T at 120 s

Calculating RTD

To calculate RTD the average T value at the outlets should be calculated first. The “integrate variables function” of ParaView can be used for this purpose.

```
>foamToVTK
```

Load the outlet VTK file into paraview using following path:

File > Open > VTK > outlet > outlet_.vtk > OK > Apply

Select T from variables menu, and then integrate the variables on the outlet:

Filters > Data Analysis > Integrate Variables > Apply

The values given in the opened window are integrated values in this specific time step. By changing the time step values for different time steps are displayed. As mentioned before, the average value of the property is needed. Therefore, these values should be divided by outlet area to get average values ($1\text{m} \times 1\text{m}$).

The same procedure should be followed for calculating RTD of `inlet_two`, except T value for `inlet_one` should be 0 and for `inlet_two` it should be 1.0.

Calculating RTD

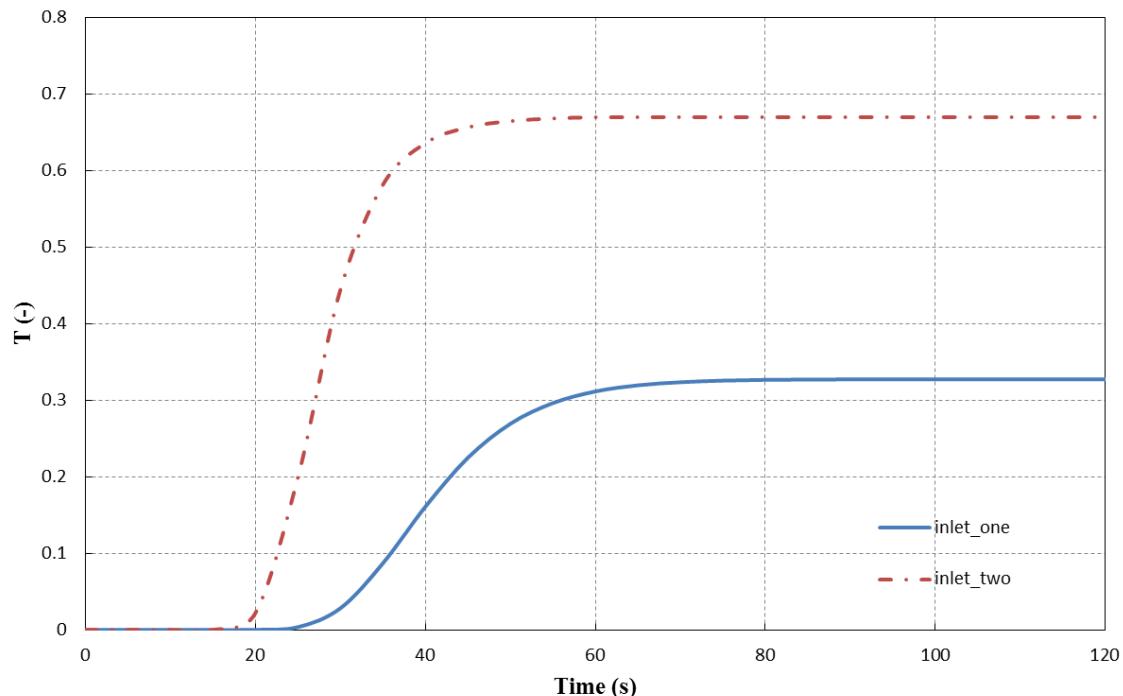


Figure 10.4 Average value of T on the outlet for two inlets versus time

The average value of T for each outlet approaches a certain constant value, which is the ratio of that scalar mass inlet to the whole mass inlet.

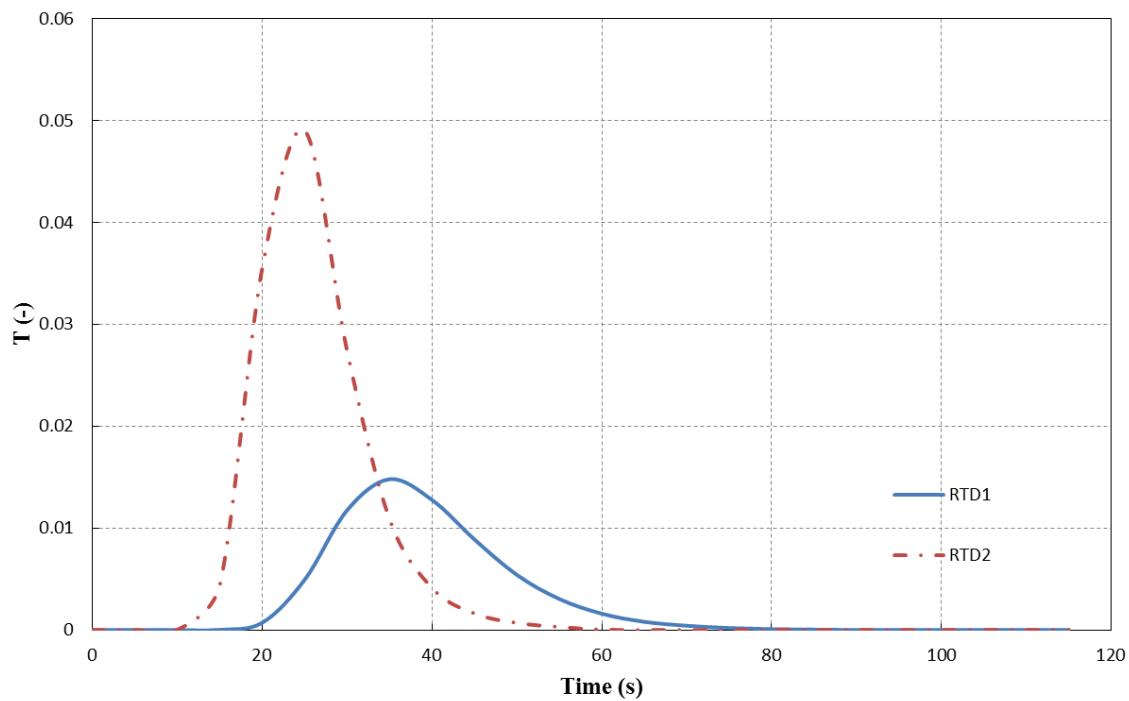


Figure 10.5 RTD of two inlets