

# Solver Development in OpenFOAM

C. Fernandes, L.L. Ferrás, J.M.Nóbrega

i3N/IPC – Institute for Polymers and Composites, University of Minho, Campus de Azurém, Guimarães, Portugal

## Outline

- Modify *simpleFoam* solver to predict temperature evolution
- Compile *utility* to compute also any field average and standard deviation
- Mixing flow Case Study

# Modify *simpleFoam* solver

- Create ***applications*** folder in ***\$WM\_PROJECT\_USER\_DIR*** folder
  - » `mkdir $WM_PROJECT_USER_DIR/applications`
- Copy ***simpleFoam*** solver from ***\$FOAM\_SOLVERS*** folder to *the new* folder
  - » `cp -r $FOAM_SOLVERS/incompressible/simpleFoam $WM_PROJECT_USER_DIR/applications`
- Rename new ***simpleFoam*** folder to ***simpleFoamTemp***
  - » `cd $WM_PROJECT_USER_DIR /applications`
  - » `mv simpleFoam simpleFoamTemp`

# Modify *simpleFoam* solver

- Rename the source code file ***simpleFoam.C***

- » `cd simpleFoamTemp`

- » `mv simpleFoam.C simpleFoamTemp.C`

- Edit ***files*** file that is in ***Make*** folder to read

```
SimpleFoamTemp.C
```

```
EXE = $(FOAM_USER_APPBIN)/simpleFoamTemp
```

- Return to ***simpleFoamTemp*** folder and compile the solver (to be sure everything was done properly)

- » `wclean`

- » `wmake`

# Modify *simpleFoam* solver

- Verify if the new solver was created in **`$FOAM_USER_APPBIN`** folder

» `ls $FOAM_USER_APPBIN`

- New transport equation to solve (viscous dissipation is neglected)

$$\underbrace{\nabla \cdot (\vec{u}T)}_{\text{advection}} - \underbrace{\alpha \nabla^2 T}_{\substack{\text{diffusion} \\ \text{(heat conduction)}}} = 0$$

- We have to insert the new transport equation in the solver, add a new scalar field **T** (for temperature) and create the new scalar field **DT** (that will represent the **thermal diffusivity  $\alpha$** )

# Modify *simpleFoam* solver

- Analyze and edit ***createFields.H*** file
  - » leafpad createFields.H
- Insert the following lines **after velocity vector field definition** to add the new **temperature field** (hint: start by copy and paste the pressure field definition lines)

```
Info<< "Reading field T\n" <<endl;
volScalarField T
(
    IOobject
    (
        "T",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

# Modify *simpleFoam* solver

- Insert the following lines after *singlePhaseTransportModel* definition to define the new **DT** field

```
Info<< "Reading transportProperties\n" << endl;
IOdictionary transportProperties
(
    IOobject
    (
        "transportProperties",
        runtime.constant(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    )
);
dimensionedScalar DT
(
    transportProperties.lookup("DT")
);
```

# Modify *simpleFoam* solver

- Analyze and edit ***simpleFoamTemp.C*** file
  - » `leafpad simpleFoamTemp.C`
- Add the command to insert the resolution of Temperature equation after ***turbulence->correct()***; command
  - » `#include "TEqn.H"`
- Analyze ***UEqn.H*** and ***pEqn.H*** files
- Create ***TEqn.H*** file (hint: start by copying ***UEqn.H***)
  - » `cp UEqn.H TEqn.H`
  - » `leafpad TEqn.H`

# Modify *simpleFoam* solver

- After editing the ***TEqn.H*** file should read (review the new equation to be solved)

```
tmp<fvScalarMatrix> TEqn
(
    fvm::div(phi, T)
    - fvm::laplacian(DT, T)
);
TEqn().relax();
TEqn().solve();
```

- Compile the new solver
  - » *wclean*
  - » *wmake*

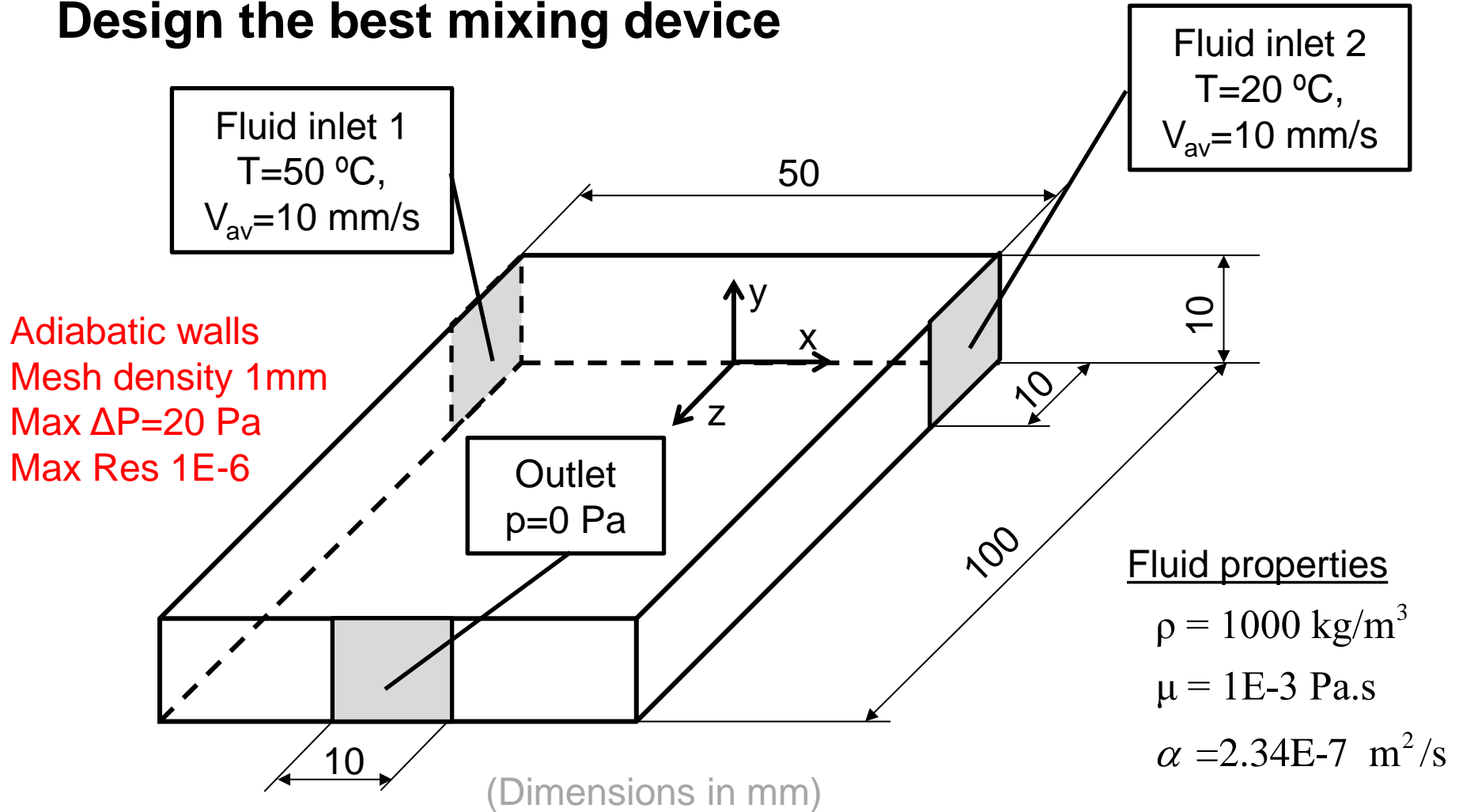
# New utility

- Study *patchAverageStd* utility located in **A4** folder
- Compile *patchAverageStd* utility

$$\sigma_T = \sqrt{\frac{\sum_{i=1}^{nfaces} (T_i - \bar{T})^2 \times A_i}{A_T}}$$

# Mixing Flow Case Study

Design the best mixing device



# Mixing Flow Case Study

## To-do:

- Use simpleFoam/pitzdaily tutorial as base study;
- Define laminar flow (**constant/RASProperties**)
- Insert DT definition in **transportProperties**
- Create new **blockMeshDict** file (adapt file from A4 folder)
- Generate mesh and check it (**checkMesh** and **paraFoam**)
- Remove utility calculations from **controlDict**
- Define missing discretization schemes (try to run code to identify what is missing)
- Define System of Equation Solver for Energy Equation (copy pressure solver)

# Mixing Flow Case Study

- For the previous problem study the effect of the Thermal diffusivity on the mixing efficiency

