OpenFOAM

An Open source alternative to commercial CFD Package

Instruction Manual for Solving Heat Transfer over solid rod using OpenFOAM

Prepared By:

CFD Team

FOSSEE

Indian Institute of Technology, Bombay

December 2017

Objective

To carry out CFD simulation using OpenFOAM, to obtain the axial temperature profile in a solid rod having constant temperature at both the ends and heat source. The wall of the rod is maintained at constant heat loss. The source causes uniform generation of heat in the rod.

Geometry and Conditions

Straight horizontal pipe

Radius of the rod (r) = 1 cm

Length of the rod (L) = 30 cm

Rod is aligned along the x-axis

Material of the rod : Copper.

Temperature at x=0 : 370 K

Temperature at x=L : 310 K

Wall is kept at constant heat loss of 400 Watts.

Volumetric rate of heat generation (q) = $1 * 10^7 W/m^3$

Step 1

1. Press **Ctrl**+**Alt**+**t** to open the terminal
2. Now you are in the home directory of your user.
3. Create a folder for our case.

   ```
   mkdir heattransfer
   ```

4. To open the case folder type

   ```
   cd heattransfer
   ```

Step-2

1. Before starting to problem we need to select our solver according to the flow conditions.
2. For this case since we are dealing with heat conduction over a rod we can choose **laplacianFoam** or **scalarTransportFoam** solver.
3. We choose **scalarTransportFoam** solver.

4. To create the case directory of our problem, we make use of the tutorial case for **scalarTransportFoam** solver, which is opened by typing the following command in the terminal

```
cd $FOAM_TUTORIALS
cd basic/scalarTransportFoam/pitzdaily
```

5. Now type **ls** command in the terminal to display the contents inside the folder.

6. To copy the files **0, constant and system** folders to our case folder **heattransfer** type the following command.

```
cp -r 0 constant system /home/test<UserID>/heattransfer
```

Solver Modification

1. There are couple of ways to add the heat generation term which is specified in problem statement, one is using fvOptions and other is adding the source term in the equation in the solver.

2. In this tutorial we do the second one, which is modifying the solver by adding the source term in to the equation.

3. For this simulation, the **scalarTransportFoam** is used. In this section it is explained how to modify the scalarTransportFoam to add the heat source.

4. In general **scalarTransportFoam** is used to solve the **Convection-Diffusion** equation, but it can be made to solve Conduction problems by defining velocity field as zero.

5. Now Adding Heat Source in the solver.

6. Open the solver in the source directory. For this typing the following command in the terminal.

```
cd ..
cd ..
cd opt/openfoam5/applications/solvers/basic
```

7. Copy the **scalarTransportFoam** folder and paste it in the directory where you kept the case folder **heattransfer**.

```
cp -r scalarTransportFoam/ /home/test<UserID>/heattransfer

cd
cd heattransfer
```

8. Rename the folder as htFoam

```
mv scalarTransportFoam htFoam
```

9. The general Convection Diffusion equation is given by

$$\frac{\partial T}{\partial t} - \nabla.(\alpha \nabla T) + \nabla.(\vec{u}T) = Q \tag{1}$$

10. The equation is solved in the **scalarTransportFoam.C** file.

11. Open the folder, and open the **scalarTransportFoam.C** file using the any editor of your choice.

```
cd
cd heattransfer/htFoam
gedit scalarTransportFoam.
```

12. To add the source term Q, which we defined, by adding the term Q in the equation as shown below.

```
fvScalarMatrix TEqn
            (
                fvm::ddt(T)
              + fvm::div(phi, T)
              - fvm::laplacian(DT, T)
              - Q
             ==
                fvOptions(T)
            );
```

13. The term **fvOptions(T)** is given to add the source term. Since we are explicitly adding the heat source directly, we can ignore it, since it won't be taken in to account while solving.

14. The term **fvm** indicates that the derivatives are solved implicitly.

15. The term **ddt** indicates the derivative with respect to time, **div** indicates divergence.

16. **DT** is the thermal diffusivity co-efficient.

17. The unit of **Q** should match it with other terms.The units of other terms in the equation are in K/sec. Hence the specified value of Q which is $1 * 10^7 W/m^3$ is divided with the density and specific heat capacity values of copper and we get value of Q as 2.8617 K/sec

    Density of Copper = $8960 kg/m^3$

    Specific Heat Capacity of Copper = $0.39 * 10^3$

18. Add the following lines in the file below the line *#include"CourantNo.H"*

```
dimensionedScalar Q
        (
        "Q",
        dimensionSet(0,0,-1,1,0,0,0),
        2.8617
        );
```

19. The complete file is shown below.

```
    #include "createFields.H"
    #include "createFvOptions.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

    Info<< "\nCalculating scalar transport\n" << endl;

    #include "CourantNo.H"

    dimensionedScalar Q
    (
    "Q",
    dimensionSet(0,0,-1,1,0,0,0),
    2.8617
    );

    while (simple.loop())
    {
        Info<< "Time = " << runTime.timeName() << nl << endl;

        while (simple.correctNonOrthogonal())
        {
            fvScalarMatrix TEqn
            (
                fvm::ddt(T)
              + fvm::div(phi, T)
              - fvm::laplacian(DT, T) - Q
             ==
                fvOptions(T)
            );

            TEqn.relax();
            fvOptions.constrain(TEqn);
            TEqn.solve();
            fvOptions.correct(T);
        }

        runTime.write();
    }

    Info<< "End\n" << endl;

    return 0;
}
```

Figure 1: Edited .C file

20. Save the file and close it.

21. Now go back one step backward in the folder, where you will find **Make** folder. Open it and open the file named as **files**

```
cd
cd heattransfer/htFoam
ls
cd Make
gedit files
```

22. Replace the lines as shown below.

```
scalarTransportFoam.C
EXE = $(FOAM_USER_APPBIN)/htFoam
```

23. Type the following in the terminal, to compile our new solver named **htFoam**.

```
cd
cd heattransfer/htFoam
wclean
wmake
```

24. The modified solver named **htFoam** will be compiled.

25. This solver can be used to run the case by typing **htFoam** in the terminal.

Coying the Mesh file

1. Delete the existing blockMeshDict in the system folder.

```
cd heattransfer/system
rm blockMeshDict
```

2. Now we can copy the blockMeshDict file made for our problem to the system directory.

3. The mesh is made using **blockMeshDict**,which is an OpenFoam Meshing Utility.

4. Copy the **blockMeshDict**file to your case directory by following command

```
cd ..
cd ..
cd opt/heattransfermesh
cp blockMeshDict /home/test<userId>/pipe/system
```

5. The patch names defined in the mesh are **front**, **back** and **walls**.

Setting Boundary Conditions

1. Now we need to make changes in the case folder.

2. Open the case file by

```
cd
cd heattransfer
```

3. It consists of three files, which are **0**,**constant** and **system**.

4. We need to make changes in files of the **0** folder to setup initial boundary conditions for our case.

5. Open the case folder and open the **0** folder, which contains two files which are **U** and **T**.

```
cd
cd heattransfer
ls
```

6. Since we have three boundary patches in our mesh :- front, back and walls, we need to enter these three face types in these two files as well.

7. Open the **T** file using editor of your choice.

```
cd
cd heattransfer/0
gedit T
```

8. The first line shows dimensions. These dimensions set consists of 7 basic units such as [Mass Length Time Temperature Quantity Current Luminosity].

9. keep the dimensions for the file as default. In the **T** file it will be

```
dimensions      [0 0 0 1 0 0 0];
```

10. In our case we need to define boundary conditions for three patches namely **front**,**back** and **walls**.

11. Delete the two other patches namely **lowerWall** and **frontAndBack**.

12. Rename the patch name **upperWall** as **walls**.

13. The temperature at front face and back face are fixed. Therefore we need use type as **fixedValue** for both the **front** and **back** patches

```
front
    {
        type            fixedValue;
        value           uniform <Given Value>;
    }

back
    {
        type            fixedValue;
        value           uniform <Given Value>;
    }
```

14. For the wall patch, constant heat flux boundary condition is to be specified.

15. As given in problem statement, we need to give 400 W constant heat loss around the pipe.

16. The heat flux due to conduction is given by

$$q^{''} = -k\nabla T \tag{2}$$

17. Since external heat loss is gives as 400 W, the heat flux is calculated by dividing the external heat source by surface area of the pipe.

$$A = \pi DL \tag{3}$$

18. Now substitute the heat flux value in the formula given and calculate the corresponding temperature gradient. Where the thermal conductivity of copper is given as $401\frac{W}{mK}$.

19. The calculated gradient is given as boundary condition for walls patch as shown below

```
walls
    {
        type                    fixedGradient;
        gradient                uniform -<calculated value>;
    }
```

20. The initial data for the overall domain can be initialized as 0 by the following line defined after dimensions

```
internalField    uniform 0;
```

21. Save this file and close it.

22. Now open the **U**.

```
cd
cd heattransfer/0
gedit U
```

23. Since this case is heat conduction problem we should define the velocity as 0.

24. The first line shows dimensions. These dimensions set consists of 7 basic units such as [Mass Length Time Temperature Quantity Current Luminosity].

25. keep the dimensions for the file as default. In the **U (m/s)** file it will be

```
dimensions      [0 1 -1 0 0 0 0];
```

26. Since velocity is vector we need to define three components for it.

27. The boundary patches are edited as shown below.

```
front
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

back
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
walls
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
```

28. The initial data for the overall domain can be initialized as 0 by the following line defined after dimensions

```
internalField    uniform (0 0 0);
```

29. **We have entered boundary conditions for all the files. Once again check the files for spelling mistakes and semicolons and space, since OpenFOAM is case sensitive and might throw back an error in case of these issues.**

Setting Physical Properties

1. The thermal diffusivity of copper is $0.00011 \frac{m^2}{s}$
2. Open the constant folder to edit the thermal diffusivity value.

```
cd
cd heattransfer/constant
gedit transportProperties
```

3. Change the value as 0.00011.

```
DT      DT [0 2 -1 0 0 0 0] 0.00011
```

4. Save the file and close it.

Setting Simulation Parameters

1. We will make changes to **controlDict** file of the **system** folder.
2. Go inside the system folder and open the controlDict file.

```
cd
cd heattransfer/system
gedit controlDict
```

3. Change the **endTime** as **1000**.
4. Change the **deltaT** as **1**.
5. Save this and close the file.
6. This completes the setup of our heattransfer case. Let us begin to solve it.

Meshing

1. The first step here is to mesh the geometry.
2. The mesh file has been provided.
3. Copy the blockMeshDict to the system directory

```
cd ..
cd ..
cd opt/heattransfer_mesh
cp blockMeshDict /home/<username>/heattransfer/system
```

4. Now we can mesh the geometry

5. To do this open the case file heattransfer in the terminal and type **blockMesh** (Note that M is capital) and press enter.

```
cd
cd heattransfer
blockMesh
```

6. Your terminal window will display your geometry parameters and also the total number of cells in the geometry.

7. Then type **checkMesh** command.

```
checkMesh
```

8. It will display whether the **mesh is OK** of it has any errors. In case of any error have a better look at the error in the terminal and make changes accordingly.

Solving

1. We need to solve our case now. The solver for this case is htFoam, which we created.

2. Open the case folder by following command

```
cd
cd heattransfer
```

3. In your terminal window, type htFoam and press enter.

```
htFoam
```

4. The iterations running will be seen in your terminal window.

5. Wait till the iterations are over..

Post-Processing

1. Open the case folder.

```
cd
cd heattransfer
```

2. To view the results in again open paraview from your terminal using the command paraFoam.

paraFoam

3. This will Open up the paraview window as shown in the figure.

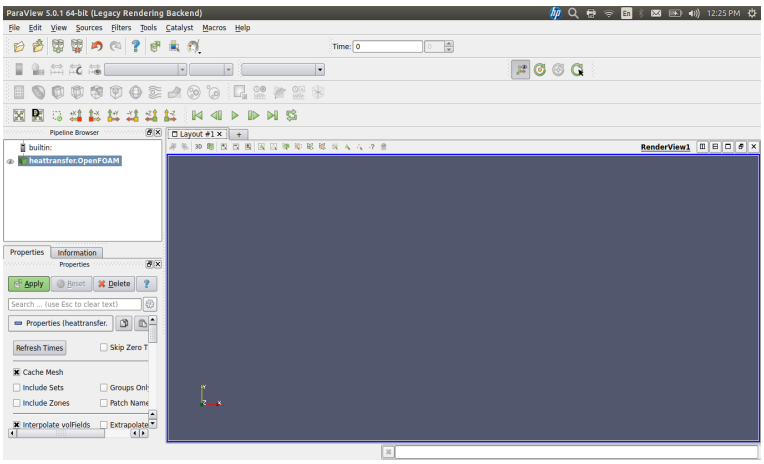


Figure 2: Paraview window

4. In the pipeline browser check the T and uncheck the U.

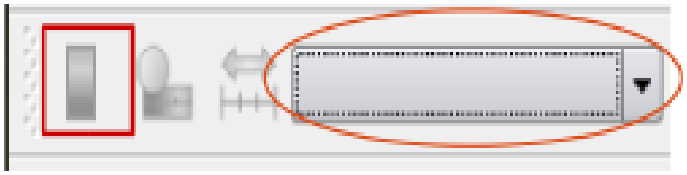
5. Click Apply to view the geometry

6. Now from the Drop down menu in the top menu bar select **T**.

7. Now to see the temperature contour click on the play button on the top menu bar as shown in figure below.

Figure 3: Paraview window

8. Now click on the button adjacent to the temperature drop down menu which says (Rescale to data range) when you move the cursor over it.



9. The temperature contour is shown below.

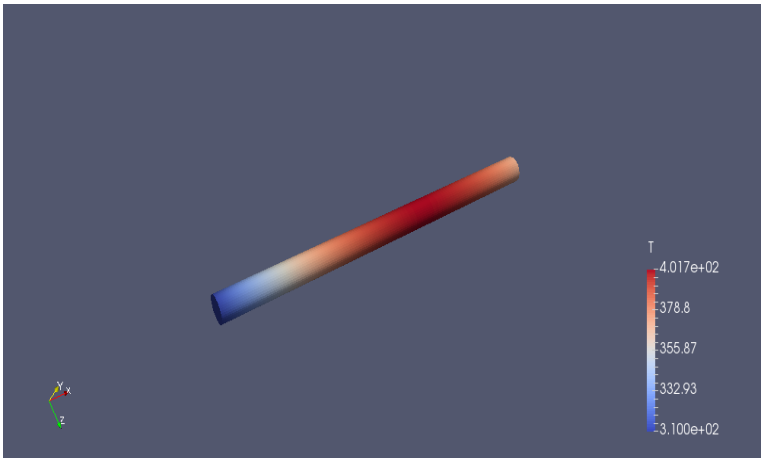


Figure 4: Temperature Contour

Plotting Profile in paraFoam

1. Now we need to plot the axial variation of temperature (along the length of the pipe).
2. To do so, on the left most top of the paraview window go to Filters → Recent → Plot Over Line
3. Click on the X-Axis on the **Pipeline Browser** and then click Apply.
4. You will see the temperature profile along the length of the pipe as shown below.

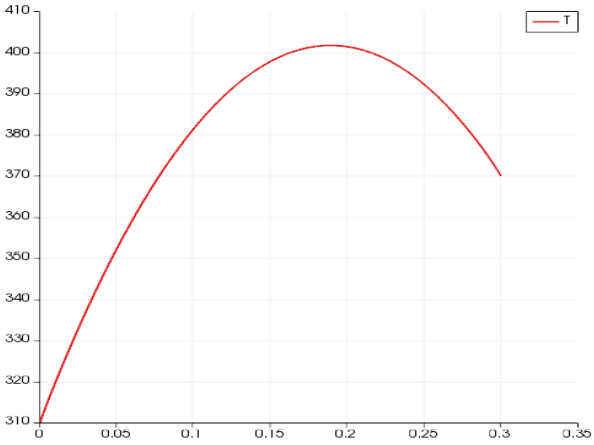


Figure 5: Axial Variation of Temperature

5. This brings us to the end of this tutorial.