Computational Homework Assignment 3

Student : Ran Li

December 20, 2016

1 Problem Setting Up

Problem 6.42:

Air at a temperature of $20°C$ and a velocity of 1 m/s flows over an aligned heated flat plate of length 1 m. A heat flux input of $200W/m^2$ is prescribed at the surface. Assuming laminar flow, calculate the temperature and velocity distribution in the flow.

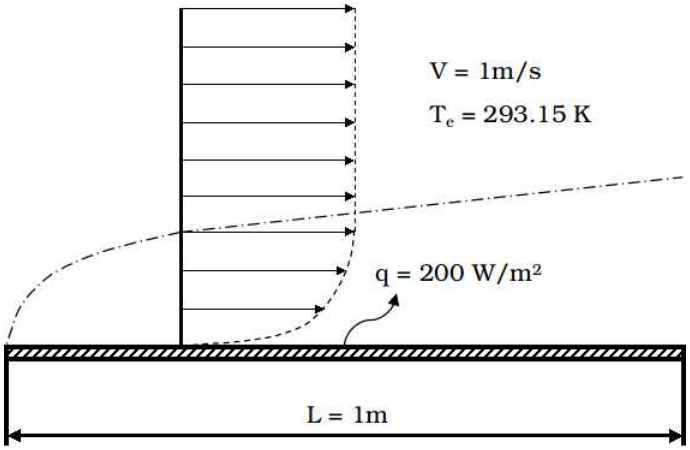


Figure 1: Problem 6.42 Setup

This problem is solved using OpenFOAM

2 Mathematical Description of the Problem

In the setting up of the problem, no relationship between temperature field/parameters and velocity field/parameters. So these two fields could be regarded as independent with respect to each other. So the idea is to solve for velocity field first and then derive the temperature field based on the velocity field acquired.

Part I: Velocity Field

This problem is a transient problem with a constant flow velocity of $V = 1m/s$. Since there's no body force term and pressure gradient is also neglected, governing equations for the problem could be composed as below:

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} = \nu\left(\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} = \nu\left(\frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2}\right) \tag{3}$$

Velocity profile of the flow field could be derived by solving the equations above, where equation 1 is the continuity equation and equations 2 and 3 are momentum equations along x and y axises. For our 2D boundary layer problem, these equations could be then simplified into the following format:

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = 0 \tag{4}$$

$$\frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} = \nu\left(\frac{\partial^2 u_x}{\partial y^2}\right) \tag{5}$$

where ν is kinematic viscosity of flow media (in this case it's air).

Part II: Temperature Field

Taking the flow as a 1D flow since we've already assumed that it's laminar flow. Deriving the energy equation near the plate:

$$\frac{\partial T}{\partial t} + u_x\frac{\partial T}{\partial x} = \alpha\frac{\partial^2 T}{\partial x^2} + \frac{\dot{Q}}{\rho C_p L} \tag{6}$$

where α is thermal diffusivity, ρ is density and C_p is heat capacity (constant pressure) of air.

Then the energy equation in the flow could be generally written as:

$$\frac{\partial T}{\partial t} + u_x\frac{\partial T}{\partial x} = \alpha\frac{\partial^2 T}{\partial x^2} \tag{7}$$

Thus once velocity field is acquired, temperature field could be solved with the equation above. Parameters involved in these equations are acquired from the standard table, with the assumption that coefficients at $20°C$ could be valid through out the calculation.

Parameter Name	Value
ν	$1.568 \times 10^{-5} m^2/s$
α	$22.07 \times 10^{-6} m^2/s$
ρ	$1.177 kg/m^3$
C_p	$1.0049 kJ/kgK$

Table 1: Parameters applied in our computation (at a temperature of $20°C$)

3 Discretization of Computational Domain

Setting up a 2D flowing domain and assume it's viscous all over the region. The region is of rectangular shape with longer edge having same length of $L = 1m$. Applying boundary conditions to the physical domain:

For velocity and temperature fields:

Patch	Inlet	Outlet	Upper Wall	Bottom Wall
Value	Fixed Value of $1m/s$	Zero Gradient	Zero Gradient	No Slip

(a) Velocity Field Boundary Condition

Patch	Inlet	Outlet	Upper Wall	Bottom Wall
Value	Zero Gradient	Zero Gradient	Zero Gradient	Fixed Gradient of $\frac{\dot{Q}}{\rho C_p}$

(b) Temperature Field Boundary Condition

Table 2: Boundary Conditions for Velocity Field and Temperature Field

Calculating the parameter mentioned in the table above: $\frac{\dot{Q}}{\rho C_p L} = 0.169 K/s$.

For this problem, we generated a grid of 100 elements along length and 20 elements along thickness. In OpemFOAM, even 2D problem have to be meshed in 3D's manner. Yet by simply setting front and back faces as type of "empty" could easily converge the grid constructed in 3D into a 2D description. The grid is somehow a bit rough which is because we have to consider the effect of Courant Number since high resolution in space may result in instability of our computation (Courant Number exceeds 1). Smaller time interval reduces the spacial resolution of our grid and it's why we are using relatively rough grid in this case.

Grid generated using "blockMesh" command in OpenFOAM is shown down below:

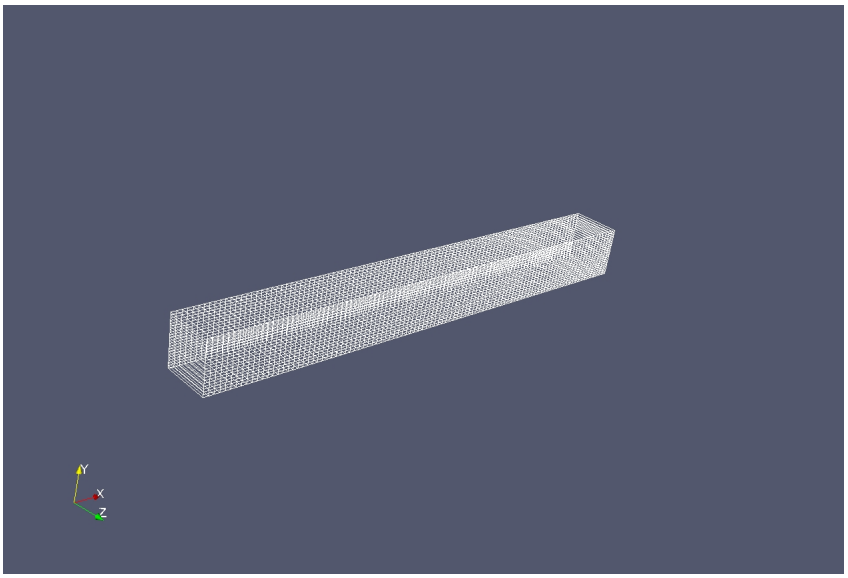
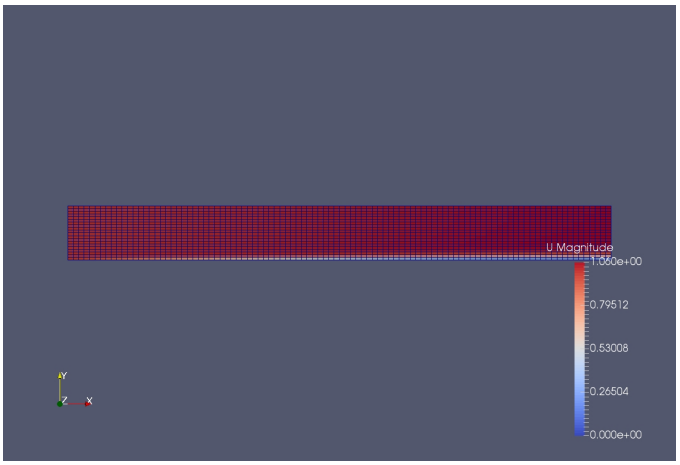


Figure 2: Grid Generated for Problem Solving, where air flow layer thickness is 0.1m

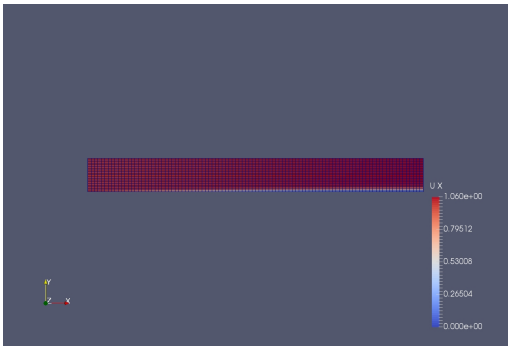
4 Numerical Solution and Results Analysis

Since in this problem, temperature field and velocity field are uncoupled, I applied solver "$icoFOAM$" to solve for velocity field and "$scalarTransportFoam$" to solve for temperature field.

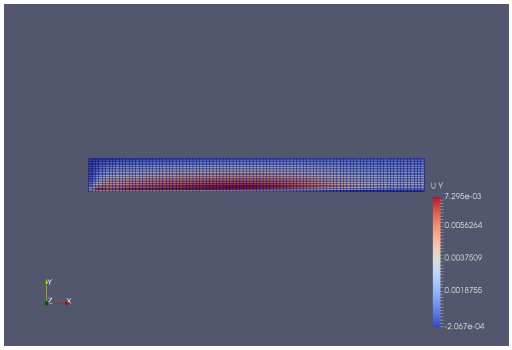
Applying boundary conditions mentioned above for velocity field and solve it using"$icoFOAM$"solver. Since this is a transient problem, we have to make sure that the system will reach steady state after certain amount of time. In my computation I calculated the flow field from $0s$ to $10s$ at an interval of $0.005s$. The result indicates clearly that there exist a boundary layer near the non-slipping surface, and it's adjacent domain do have minor velocity component along y axis.



(a) Magnitude of Velocity



(b) Velocity along x axis



(c) Velocity along y axis

Figure 3: Velocity Field Acquired from Solver"$icoFOAM$" at $t = 10s$

Then we incouple this result to solve for temperature field using "$scalarTransportFoam$". This solver will read in pre-calculated flow field (in the format of a vector field) and calculate scalar field based on it. In our case the temperature field and velocity field are not coupled and thus we could apply such method. If parameters like viscosity is related with temperature or diffusion coefficients related with coordinates we have to apply coupling methods.

Temperature Field acquired from our calculation is shown down below:

Figure 4: Temperature Field Acquired from Solver "*scalarTransportFoam*"

Attachment: OpenFOAM Scripts

Part I: Flow field solution

Grid Generation:

Listing 1: blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
|
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox
|
|  \\    /   O peration          | Version:  3.0.x
|
|   \\  /    A nd                | Web:      www.OpenFOAM.org
|
|    \\/     M anipulation       |
|
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
convertToMeters 1;

vertices
(
    (0 0 0)
    (1 0 0)
    (1 0.1 0)
    (0 0.1 0)
    (0 0 0.1)
    (1 0 0.1)
    (1 0.1 0.1)
    (0 0.1 0.1)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (100 20 1) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
    upperWall
    {
        type patch;
        faces
        (
```

```
            (3 7 6 2)
        );
    }
    lowerWall
    {
        type wall;
        faces
        (
            (1 5 4 0)
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            (0 3 2 1)
            (4 5 6 7)
        );
    }
);

mergePatchPairs
(
);

// ************************************************************************* //
```

Boundary Conditions:

Listing 2: boundary

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  3.0.x                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
```

```
    format      ascii;
    class       polyBoundaryMesh;
    location    "constant/polyMesh";
    object      boundary;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

5
(
    inlet
    {
        type            patch;
        nFaces          20;
        startFace       3880;
    }
    outlet
    {
        type            patch;
        nFaces          20;
        startFace       3900;
    }
    upperWall
    {
        type            patch;
        nFaces          100;
        startFace       3920;
    }
    lowerWall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          100;
        startFace       4020;
    }
    frontAndBack
    {
        type            empty;
        inGroups        1(empty);
        nFaces          4000;
        startFace       4120;
    }
)

// ************************************************************************* //
```

Control Dictionary:

Listing 3: controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
|
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
|
|  \\    /   O peration     | Version:  3.0.x
|
|   \\  /    A nd           | Web:      www.OpenFOAM.org
|
|    \\/     M anipulation  |
|
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     icoFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         10.0;

deltaT          0.005;

writeControl    timeStep;

writeInterval   20;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;
```

```
writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;


// ************************************************************************* //
```

Initial Velocity Field:

Listing 4: U

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
|
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox
|
|  \\    /   O peration          | Version:  3.0.x
|
|   \\  /    A nd                | Web:      www.OpenFOAM.org
|
|    \\/     M anipulation       |
|
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1 0 0);
    }
```

```
    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

Part II: Tempreture field solution

Boundary Conditions:

Listing 5: boundary

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
|
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox
|
|  \\    /   O peration          | Version:  3.0.x
|
|   \\  /    A nd                | Web:      www.OpenFOAM.org
|
|    \\/     M anipulation       |
|
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
```

```
    format      ascii;
    class       polyBoundaryMesh;
    location    "constant/polyMesh";
    object      boundary;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

5
(
    inlet
    {
        type            patch;
        nFaces          20;
        startFace       3880;
    }
    outlet
    {
        type            patch;
        nFaces          20;
        startFace       3900;
    }
    upperWall
    {
        type            patch;
        nFaces          100;
        startFace       3920;
    }
    lowerWall
    {
        type            wall;
        inGroups        1(wall);
        nFaces          100;
        startFace       4020;
    }
    frontAndBack
    {
        type            empty;
        inGroups        1(empty);
        nFaces          4000;
        startFace       4120;
    }
)

// ************************************************************************* //
```

Tempreture Field:

Listing 6: T

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
|                                |                                            |
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox      |
|                                |                                            |
|  \\    /   O peration          | Version:  3.0.x                            |
|                                |                                            |
|   \\  /    A nd                | Web:      www.OpenFOAM.org                 |
|                                |                                            |
|    \\/     M anipulation       |                                            |
|                                |                                            |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 273.15;

boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform 293.15;
    }

    outlet
    {
        type            zeroGradient;
    }

    upperWall
    {
        type            zeroGradient;
    }

    lowerWall
```

```
    {
        type            zeroGradient;
        gradientValue   uniform 0.169;
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

Velocity field is acquired from the last step of part I. The script is too long to be implanted.