Implementing Streamwise Periodic Boundary Condition in OpenFOAM

Harikrishnan S
Research Scholar
Department of Mechanical Engineering
Indian Institute of Technology Madras, Chennai, Tamilnadu, India

ABSTRACT

The objective of the present project is to describe the implementation of streamwise periodic boundary condition in open source CFD package OpenFOAM. Periodic boundary conditions are used in numerical investigations to save the computational time. Fully developed flow in plane channel has been considered in the present study. This project also explains setting up OpenFOAM case from existing tutorials available in the OpenFOAM repository. Geometry and mesh have been generated by using 'blockMesh' utility. Implementation of streamwise periodic boundary adds a source term in the Navier-Stokes equation. Source term has been added to the Navier-Stokes equation by using 'fvOption' utility. Steady, laminar flow has been considered and it is solved by using 'simpleFoam' solver. Obtained results are validated with the analytical results available in the literature.

1. Introduction

Periodic boundary conditions are used when the physical geometry of interest and the expected pattern of the flow/thermal solution have a periodically repeating nature. Periodic boundary conditions are widely used in numerical investigations of heat exchangers, turbo-machinery etc. to save computational time.

2. Problem Statement

Figure 1 shows the details of the plane channel geometry considered in the present study. Channel with an aspect ratio (L/H) 2 has been taken. Dimensions of the channel and flow conditions chosen in the present study have been shown in Table-1. OpenFOAM solves three-dimensional geometry in all cases, hence plane channel (two-dimensional geometry) has been extruded to third direction (z-axis) while generating the geometry. Table-2 shows the co-ordinates of the geometry chosen in the present study.

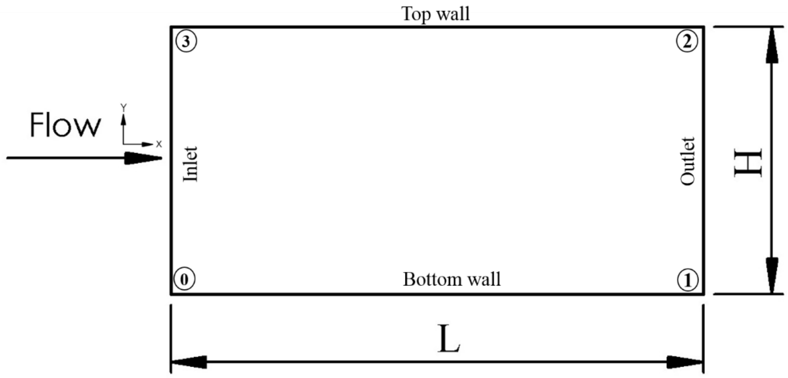

Figure 1: Geometry of plane channel

Table-1:- Details of geometry and flow conditions

Geometry details	Length of the channel (L) = 1m
	Height of the channel (H) = 0.5m
	Hydraulic diameter (D_H) = 0.5m
Fluid Property	Kinematic viscosity = 0.01 m^2/s
Reynolds number	Re =50

Table-2:- Co-ordinates of the geometry

Vertex number	Co-ordinate
0	(0 -0.25 0)
1	(1 -0.25 0)
2	(1 0.25 0)
3	(0 0.25 0)
4	(0 -0.25 0.1)
5	(1 -0.25 0.1)
6	(1 0.25 0.1)
7	(0 0.25 0.1)

3. Governing Equations and Boundary Conditions

In general, periodic boundary conditions can be classified into two,

1. Translational periodic:- Mostly used in heat exchangers
2. Rotational periodic:- Ex: Mostly used in turbo machinery

Simulating periodically fully developed flow in plane channel belongs to translational periodic category. Solving flow problems with periodic boundary conditions requires modification in momentum equation. Governing equations for mass, momentum and energy equations for periodically fully developed flow can be given as,

$$\nabla \cdot \vec{V} = 0 \tag{1}$$

$$\left(\vec{V} \cdot \nabla\right)\vec{V} = -\frac{\nabla \tilde{P}}{\rho} + \nu \nabla^2 \vec{V} + \frac{\beta}{\rho} \tag{2}$$

Where $\vec{V}$ is the flow velocity, μ, and ρ represent dynamic viscosity and density of the fluid considered, $\tilde{P}$ represent modified pressure (reduced pressure) which satisfies periodic boundary conditions and actual pressure is given by [1]

$$P(x, y) = -\beta x + \tilde{P}(x, y) \tag{3}$$

here β is the linear component of the pressure (Pressure gradient term).
There are two ways in which we can run translational periodic boundary condition cases.

1. By fixing pressure gradient (β)(mass flow rate will be varying in unsteady case)
2. By fixing mass flow rate (pressure gradient (β) will be varying in unsteady case)

Second case represents constant Re case. Details of boundary name, vertex number, and corresponding boundary conditions are presented in Table-3.

Table-3:- Boundary conditions

Boundary Name	**Vertex number**	**Boundary condition**
Inlet	7,3,0,4	Cyclic
Outlet	6,5,1,2	Cyclic
Top wall	3,7,6,2	No slip
Bottom wall	0,4,5,1	No slip
Front and Back	0,3,2,1 4,5,6,7	Empty

4. Simulation Procedure

First step in setting up of an OpenFOAM case is to identify most similar tutorial available in the OpenFOAM repository. Tutorial similar to present problem viz. periodically fully developed flow, available in the repository is 'channel395' tutorial (OpenFOAM-4.x/tutorials/incompressible/pimpleFoam/channel395/). This case deal with three-dimensional LES simulation of periodically fully developed channel flow. We need to modify the available tutorial to our requirement. Mesh generation and implementation of boundary conditions are adopted from cavity tutorial (OpenFOAM/OpenFOAM-4.x/tutorials/incompressible/icoFoam/cavity). Since we are dealing with steady, laminar case, solver settings are taken from pitzDaily tutorial (OpenFOAM-4.x/tutorials/incompressible/simpleFoam/pitzDaily/) from simpleFOAM.
Steps involved in setting up the case are

1. Copy the channel395 tutorial to user directory (eg. Desktop/mycase/channel395).
2. Replace the file name to mycase (Desktop/mycase/mycase).

4.1 Creating geometry and mesh

i. Since geometry for the present problem is rectangular domain, we can generate the geometry and mesh by modifying the 'blockMeshDict' file available in cavity tutorial (OpenFOAM/OpenFOAM-4.x/tutorials/incompressible/icoFoam/cavity).
ii. Copy 'blockMeshDict' file from the cavity tutorial and paste it to 'system' folder in user directory (Desktop/mycase/mycase/system).
iii. Suitable modifications have been done in the 'blockMeshDict' file in order to generate geometry for the present problem. Major modifications are

- Modifying the vertices co-ordinates as shown in Fig. 2.

(a)
```
convertToMeters 0.1;

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

(b)
```
convertToMeters 1;

vertices
(
    (0 -0.25 0)
    (1 -0.25 0)
    (1 0.25 0)
    (0 0.25 0)
    (0 -0.25 0.1)
    (1 -0.25 0.1)
    (1 0.25 0.1)
    (0 0.25 0.1)
);
```

Figure 2: Vertices details in (a) cavity tutorial and (b) present case

- Modifying the grid size (From 20x20x1 to 100x100x1) as shown in Fig. 3.

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

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

Figure 3: Grid size in (a) cavity tutorial and (b) present case

- Cavity tutorial has only three boundaries. Present geometry has six boundaries in this problem. Hence we need to modify the 'blockMeshDict' file accordingly. Split three boundaries (movingWall, fixedWalls and frontAndBack) present in the cavity tutorial to six boundaries (inlet, outlet, topWall, bottomWall and frontAndBack). Figure 4 shows the vertices details in cavity tutorial and the present case.

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

(a)

```
boundary
(
    topWall
    {
        type wall;
        faces
        (
            (3 7 6 2)
        );
    }
   bottomWall
    {
        type wall;
        faces
        (
            (0 4 5 1)
        );
    }
    inlet
    {
        type cyclic;
        neighbourPatch  outlet;
        faces
        (
            (7 3 0 4)
        );
    }
    outlet
    {
        type cyclic;
        neighbourPatch  inlet;
        faces
        (
            (6 5 1 2)
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            (0 3 2 1)
            (4 5 6 7)
        );
    }
```

(b)

Figure 4: Vertices and boundaries (a) cavity tutorial and (b) present case

iv. Now mesh can be generated by using the command 'blockMesh' in the terminal. Figure 5 depicts the isometric view of the generated mesh.

Figure 5: Isometric view of mesh

v. Important modifications to be done for mesh with periodic boundary condition is defining inlet and outlet as cyclic patch and identifying its neighbor patch. This can be done by modifying boundary (Desktop/mycase/mycase/constant/polyMesh/boundary) file by modifying boundary type to cyclic and adding matchTolerance, transform, neighbourPatch details as shown in Fig. 6.

```
inlet
{
    type            cyclic;
    inGroups        1(cyclic);
    nFaces          100;
    startFace       20000;
    matchTolerance  0.0001;
    transform       unknown;
    neighbourPatch  outlet;
}
outlet
{
    type            cyclic;
    inGroups        1(cyclic);
    nFaces          100;
    startFace       20100;
    matchTolerance  0.0001;
    transform       unknown;
    neighbourPatch  inlet;
}
```

Figure 6: Modifications to be made in boundary file to incorporate cyclic boundary condition

4.2 Setting up boundary conditions

Files present in '0' folder (Desktop/mycase/mycase/0) has been replaced with files taken from '0' folder of cavity tutorial. Since we are dealing with laminar steady flow we need only 'p' and 'U' files, other files can be deleted. Boundaries present in the cavity tutorial has been modified to add six boundaries of present case in both 'p' and 'U' files. Details regarding the boundary conditions for cavity tutorial and present case are listed in Table-4.

Table-4:- Details of boundary conditions of cavity tutorial and present case

Cavity			Present study		
Boundary	p	U	Boundary	p	U
movingWall	zeroGradient	fixedValue	bottomWall	zeroGradient	noSlip
fixedWalls	zeroGradient	noSlip	topWall	zeroGradient	noSlip
frontAndBack	empty	empty	inlet	cyclic	cyclic
			outlet	cyclic	cyclic
			frontAndBack	empty	empty

4.3 Setting up source term

Implementation of periodic boundary condition adds source term in the momentum equation. Source term can be added to governing equation in OpenFOAM by using the utility ‘patchMeanVelocityForce’ available in ‘fvOptions’. This can be found in Desktop/mycase/mycase/constant/fvOptions. The utility ‘meanVelocityForce’ has been used in channel395 tutorial, which keeps the average velocity of the domain to an average value in each time step. In present case, we need to keep the average velocity of inlet patch only, hence ‘patchMeanVelocity’ has been used in ‘fvOption’. Figure 7 shows the ‘fvOption’ file in (a) channel395 and (b) present case. Average velocity at the inlet is given in the Ubar section in the file.

```
momentumSource
{
    type            meanVelocityForce;
    active          yes;

    meanVelocityForceCoeffs
    {
        selectionMode   all;

        fields      (U);
        Ubar            (0.1335 0 0);
    }
}
```

(a)

```
momentumSource
{
    type            patchMeanVelocityForce;
    active          yes;

    patchMeanVelocityForceCoeffs
    {
        selectionMode   all;
        fields          (U);
        patch           inlet;
        Ubar            (1.0 0 0);
        relaxation      0.2;
    }
}
```

(b)

Figure 7: ‘fvOption’ file in (a) channel395 and (b) present case

4.4 Solver details

Steady and laminar flow has been considered in the present study. Laminar flow model can be implemented in the OpenFOAM by using ‘simulationType’ option available in the ‘turbulenceProperties’ file in constant folder. The ‘simulationType’ option can be changed from default ‘LES’ (of channel395) to ‘laminar’. In order to run steady simulations, controDic, fvSchemes and fvSolution files are taken from pitzDaily tutorial from the simpleFOAM tutorial folder. ‘simpleFOAM’ command has been used in terminal to run computations.

5. Validation of the Results

Obtained results analyzed with the help of paraFoam software. Figure 8 depicts the x-velocity contour for the considered geometry. It can be inferred from the figure that no variation is observed in the flow direction. Analytical results available in the literature for the x-velocity profile in periodically fully developed channel flow is given as [2],

$$u = 1.5\,u_{avg}\left(1 - \left(\frac{y}{H/2}\right)^2\right) \tag{4}$$

Figure 9 shows the comparison of computed values of x-velocity with the analytical results. This result indicate that the simulated results are in good agreement with the analytical results. Results are plotted with the help of 'gnuplot' (mycase$ gnuplot gnuplot.txt) and the commands used in plotting gnuplot is available in the file 'gnuplot.txt' available in the mycase directory.

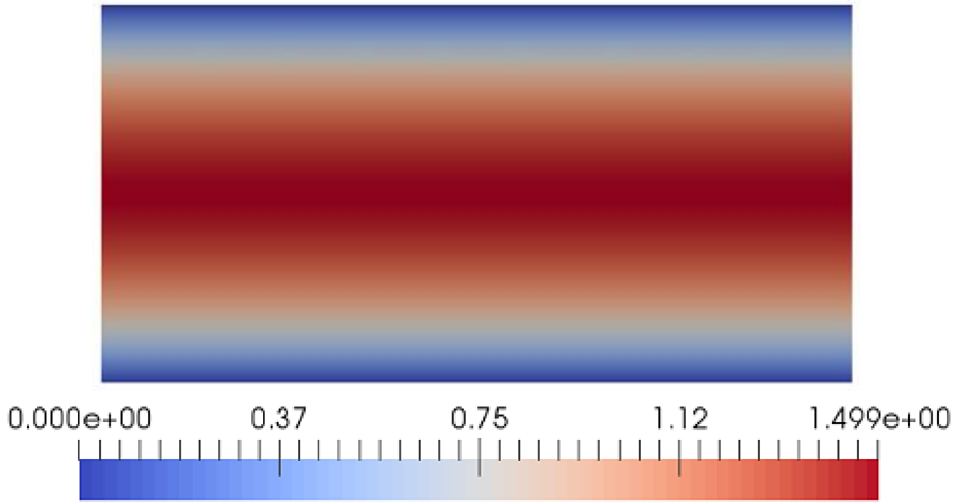

Figure 8: X-velocity contour for the geometry

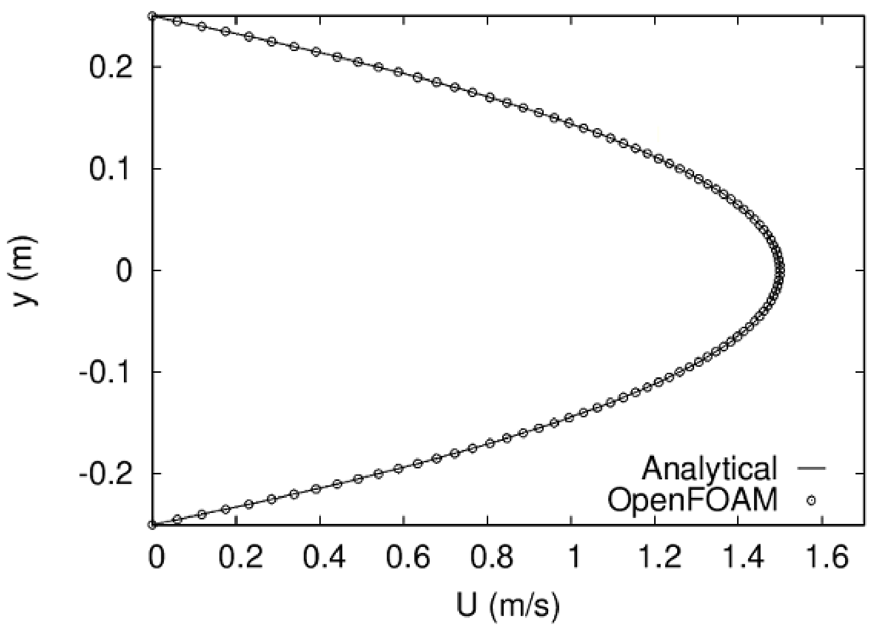

Figure 9: Comparison of computed x-velocity profile with analytical result

REFERENCES

[1] Patankar, S.V., Liu, C.H. and Sparrow, E.M., 1977. Fully developed flow and heat transfer in ducts having streamwise-periodic variations of cross-sectional area. *Journal of Heat Transfer*, *99*(2), pp.180-186.

[2] Fox, R.W., McDonald, A.T. and Pritchard, P.J., 1998. Introduction to Fluid Mechanics. 1998.