OpenFOAM Simulations of coalescing and bouncing of droplets

Prerequisites:

1. Ubuntu can be downloaded from Microsoft Store or type `wsl --install` in powershell.

2. openfoam2312 can be installed by typing

   ```
   sudo sh -c "wget -O - https://dl.openfoam.com/add-debian-repo.sh | bash"
   sudo apt update
   sudo apt install openfoam2312
   ```

   in ubuntu.

3. ParaView for post processing can be downloaded from https://www.paraview.org/download/

Github Repository

Case directory for this OpenFOAM case looks like

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         09-07-2025     12:48                0
d-----         09-07-2025     12:48                constant
d-----         09-07-2025     12:48                system


    Directory: D:\amr\0


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         24-06-2025     03:10           1409 alpha.water
-a----         24-06-2025     03:10           1523 p_rgh
-a----         24-06-2025     03:10           1444 U


    Directory: D:\amr\constant


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         24-06-2025     03:29           1831 dynamicMeshDict
-a----         24-06-2025     03:29            905 g
-a----         24-06-2025     03:29           1088 transportProperties
-a----         24-06-2025     03:29            871 turbulenceProperties


    Directory: D:\amr\system


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         24-06-2025     03:18           1744 blockMeshDict
-a----         24-06-2025     03:18           1352 controlDict
-a----         24-06-2025     03:18            928 decomposeParDict
-a----         24-06-2025     03:18           1300 fvSchemes
-a----         24-06-2025     03:18           2156 fvSolution
-a----         24-06-2025     03:18            652 refineMeshDict
-a----         24-06-2025     03:18            680 sampling
-a----         24-06-2025     03:18           1212 setFieldsDict
-a----         24-06-2025     03:18            923 topoSetDict.inner
-a----         24-06-2025     03:18            916 topoSetDict.outer
```

We have covered the entire project into seven subproblems

1. Coalescing in a coarse mesh without any mesh refinement
2. Coalescing of droplets with AMR(adaptive mesh refinement)
3. Coalescing of droplets using local refinement
4. Combining our knowledge of AMR and local refinement
5. Learning about 2d axis symmetry in OpenFOAM
6. Combining our knowledge of refinement and 2d axisymmetric case

Coalescing in a coarse mesh without any mesh refinement

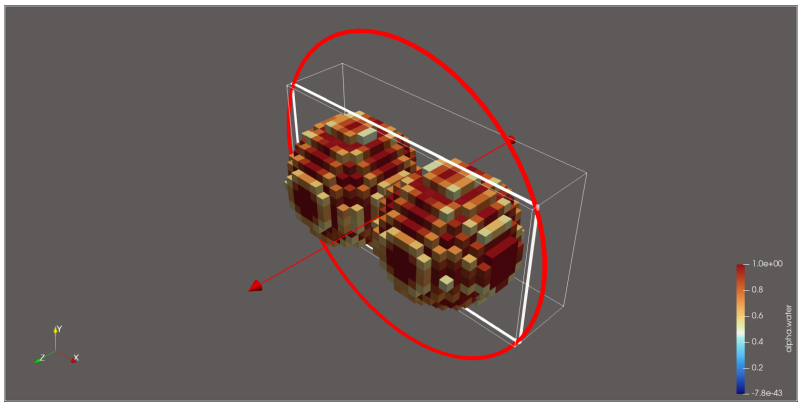
The case directory looks like

```
    Directory: D:\downloads\initial


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         09-07-2025     14:30                0
d-----         09-07-2025     14:30                constant
d-----         09-07-2025     14:30                system


    Directory: D:\downloads\initial\0


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         09-07-2025     14:30           1374 alpha.water
-a----         09-07-2025     14:30           1277 p_rgh
-a----         09-07-2025     14:30           1281 U


    Directory: D:\downloads\initial\constant


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         09-07-2025     14:30            909 g
-a----         09-07-2025     14:30           1087 transportProperties
-a----         09-07-2025     14:30            871 turbulencePropertie


    Directory: D:\downloads\initial\system


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         09-07-2025     14:30           1507 blockMeshDict
-a----         09-07-2025     14:30           2674 controlDict
-a----         09-07-2025     14:30            926 decomposeParDict
-a----         09-07-2025     14:30           1300 fvSchemes
-a----         09-07-2025     14:30           2147 fvSolution
-a----         09-07-2025     14:30            674 sampling
-a----         09-07-2025     14:30           1053 setFieldsDict
```

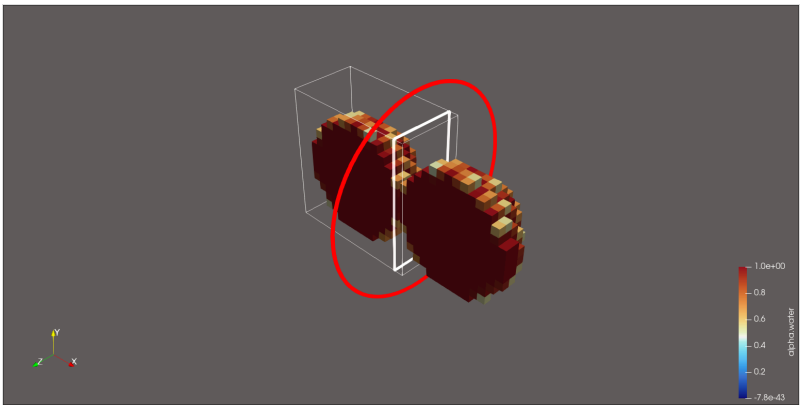
The zip file for the same is in Github named initial

Here we are making a symmetric case, about both the plane in non-gravitic directions to reduce the computing power.

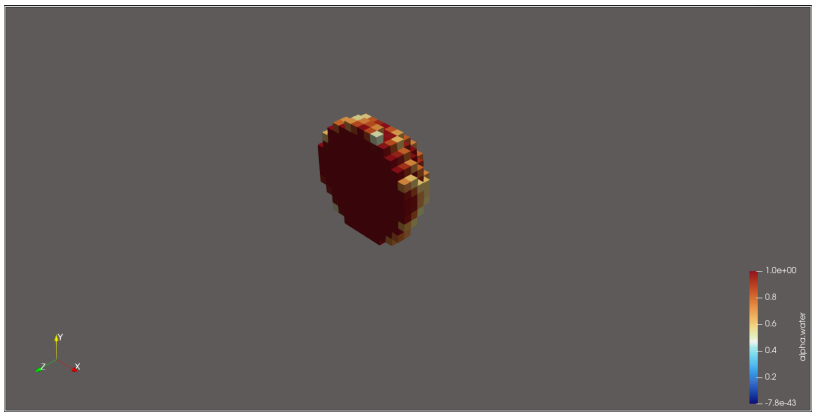
For reference:
Consider this view



We have symmetry around these two plane



So we can reduce the case to



Making the two walls symmetric.

Important snippets from each document that are to be edited for different sample cases

1. blockMeshDict

Used to define the initial mesh before any refinement.

```
vertices
(
 (0 0 -1.862)     //0
 (2.114 0 -1.862)     //1
 (2.114 2.114 -1.862)     //2
 (0 2.114 -1.862)     //3
 (0 0 0)     //4
 (2.114 0 0)     //5
 (2.114 2.114 0)     //6
 (0 2.114 0)     //7
);

blocks
(
   hex (0 1 2 3 4 5 6 7)  (151 151 133) simpleGrading (1 1 1)
);
```

To edit the number of cells or to edit the size of cells, edit the 151 151 133 under the blocks section and to adjust the size of the cells, change the dimensions of the entire block or increase or decrease the number of cells in appropriate proportions.

2. ControlDict

Used to control various simulation properties like timesteps.

```
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         0.0045;
deltaT          0.000001;
writeControl    adjustable;
writeInterval   0.0001;
```

To adjust the time steps change the deltaT preferably in orders below and not equal to the smallest mesh size in secs and meters.
And change the end time according to your need and the write interval too.

3. decomposeParDict

Used to divide the entire task to various cores of the processor change the subdomains according to your requirements or system but make sure the n under the coeffs block multiply themselves to number of sub-domains.

```
numberOfSubdomains 8;
method          simple;
coeffs
{
    n           (2 2 2);
}
```

4. setFieldsDict

Used to define regions of water in the domain.

```
defaultFieldValues
(
    volScalarFieldValue alpha.water 0
);

regions
(
    sphereToCell
            {
              centre (0.0005 0.000483 0);
              radius 0.0005;
              fieldValues
              (
                        volScalarFieldValue alpha.water 1
              );
            }

);
```

It can be edited using the center and radius according to the needs.

5. alpha.water

```
 bottomwall
    {
        type            constantAlphaContactAngle;
        theta0          165;
        limit           gradient;
        value           uniform 0;
    }
```

Theta0 is used to change the contact angle between the droplet and bottom wall.

Steps to follow in terminal to execute the file

1. Open ubuntu
2. Go to case directory
3. Type the following

```
openfoam2312
blockMesh
checkMesh
setFields
decomposePar
mpirun -np 8 interFoam -parallel
```

Coalescing of droplets with AMR (Adaptive Mesh Refinement)

AMR here is done using the property alpha water or phase fraction. We are defining a region with phase fraction 0.05 to 0.99 with 0 being entirely air and 1 being entirely water and refining the mesh in the region.

The file for the same is labeled as amr in Github.

The case directory looks exactly like the previous case but with a dynamicMeshDict file in the constant folder.

The snippet from the dynamicMeshDict file that will help us make changes in different sample cases.

```
dynamicRefineFvMeshCoeffs
{
        refineInterval      1;          // Frequency of refinement (every timestep in this case)

        field               alpha.water;   // Field used for refinement (e.g., alpha for multiphase)
        lowerRefineLevel    0.05;       // Lower threshold for refinement (e.g., if alpha > 0.05)
        upperRefineLevel    0.99;       // Upper threshold for refinement (e.g., if alpha < 0.99)

        unrefineLevel       1;       // Threshold below which the mesh is coarsened
        nBufferLayers       1;          // Number of buffer layers around refined cells
        maxRefinement       2;          // Maximum allowable refinement level
        maxCells            4000000;       // Maximum number of cells in the domain
}
```

This refinement is kind of aggressive, try to lower the maxRefinement or narrow down the lower and upper refine level and increase the refineInterval to adjust for lower spec computers.

Instructions to execute the file are similar to the previous.

Coalescing of droplets using local refinement

We achieved this by using topologicalSetDict and refineMeshDict.

topologicalSetDict selects a certain region in a domain.

refineMeshDict in our case takes the region defined and refines it in directions specified in the file.

Here in our case we are defining it in two steps for a transition layer to form in between.

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10-07-2025     17:25                0
d-----        10-07-2025     17:25                constant
d-----        10-07-2025     17:25                system


    Directory: D:\localref\0


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10-07-2025     17:25           1374 alpha.water
-a----        10-07-2025     17:25           1277 p_rgh
-a----        10-07-2025     17:25           1281 U


    Directory: D:\localref\constant


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10-07-2025     17:25            909 g
-a----        10-07-2025     17:25           1087 transportProperties
-a----        10-07-2025     17:25            871 turbulenceProperties


    Directory: D:\localref\system


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10-07-2025     17:25           1480 blockMeshDict
-a----        10-07-2025     17:25           2674 controlDict
-a----        10-07-2025     17:25            926 decomposeParDict
-a----        10-07-2025     17:25           1371 fvSchemes
-a----        10-07-2025     17:25           2062 fvSolution
-a----        10-07-2025     17:25            222 refineMeshDict
-a----        10-07-2025     17:25            674 sampling
-a----        10-07-2025     17:25           1053 setFieldsDict
-a----        10-07-2025     17:25            912 topoSetDict.inner
-a----        10-07-2025     17:25            905 topoSetDict.outer
```

Important snippets from each document that are to be edited for different sample cases

1. topologicalSetDict

For both inner and outer you can change the box dimensions depending on your use and domain case. But make sure that all the values in the former point are smaller than the later.

```
actions
(
   {
      name refineBox;
      type cellSet;
      action new;
      source boxToCell;
      sourceInfo
      {
         box (0 0 -0.0015) (0.0015 0.0015 0);
      }
   }
);
```

2. refineMeshDict

It refines the region defined by the topologicalSetDict. We can decide what directions we want refinement in.

```
set refineBox;

coordinateSystem global;

directions ("x" "y" "z");

useHexTopology yes;
writeMesh yes;
```

Change the directions as per your use in this file

The Github for the same is labeled as localref.

Instruction to execute the simulation:

1. Open ubuntu
2. Go to case directory
3. Type the following:

```
openfoam2312
blockMesh
topoSet -dict system/topoSetDict.outer
refineMesh
topoSet -dict system/topoSetDict.inner
refineMesh
```

4. After these steps copy the u alpha.water and p_rgh from 0 to latest time
5. Reopen the terminal and type the following

```
setFields
touch new.foam
decomposePar
mpirun -np 6 interFoam -parallel //change as per your requirement
reconstructParMesh
reconstructPar
```

Combining our knowledge of AMR and local refinement

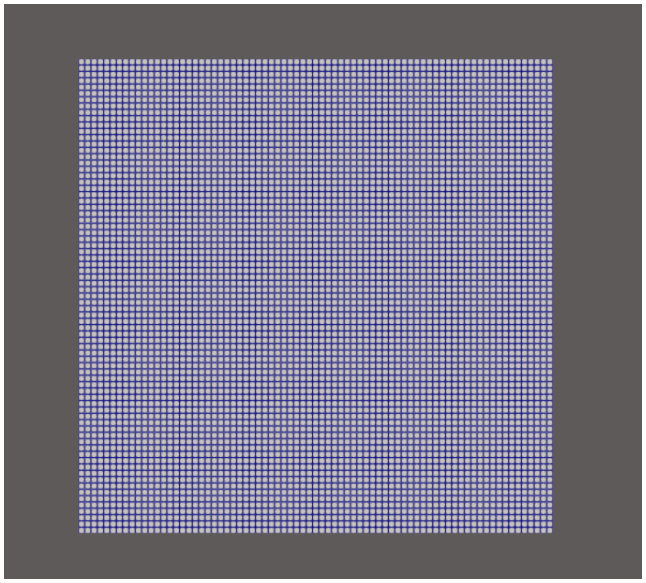
For this we will just paste our dynamicMeshDict to our locally refined case. The Github for this is named as finals.
Our case directory will look like this

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        09-07-2025     12:48                0
d-----        09-07-2025     12:48                constant
d-----        09-07-2025     12:48                system


    Directory: D:\amr\0


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        24-06-2025     03:10           1409 alpha.water
-a----        24-06-2025     03:10           1523 p_rgh
-a----        24-06-2025     03:10           1444 U


    Directory: D:\amr\constant


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        24-06-2025     03:29           1831 dynamicMeshDict
-a----        24-06-2025     03:29            905 g
-a----        24-06-2025     03:29           1088 transportProperties
-a----        24-06-2025     03:29            871 turbulenceProperties


    Directory: D:\amr\system


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        24-06-2025     03:18           1744 blockMeshDict
-a----        24-06-2025     03:18           1352 controlDict
-a----        24-06-2025     03:18            928 decomposeParDict
-a----        24-06-2025     03:18           1300 fvSchemes
-a----        24-06-2025     03:18           2156 fvSolution
-a----        24-06-2025     03:18            652 refineMeshDict
-a----        24-06-2025     03:18            680 sampling
-a----        24-06-2025     03:18           1212 setFieldsDict
-a----        24-06-2025     03:18            923 topoSetDict.inner
-a----        24-06-2025     03:18            916 topoSetDict.outer
```

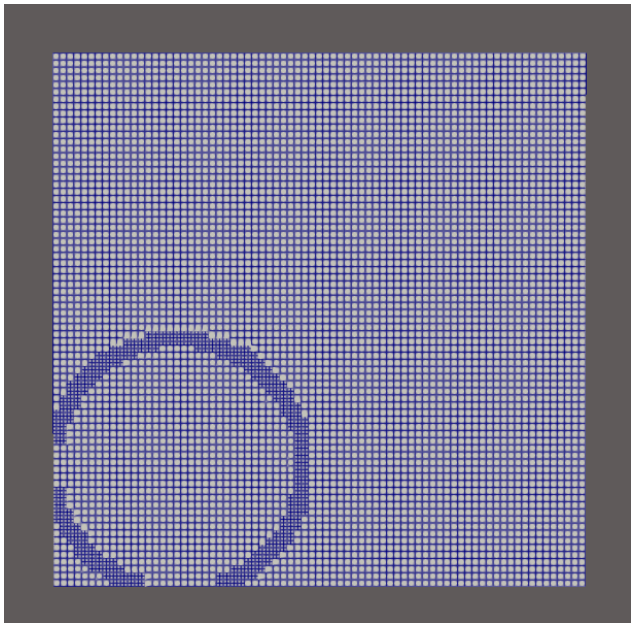
Instructions to run are similar to the previous one

The progression of mesh from step 1 to 4 will look like this

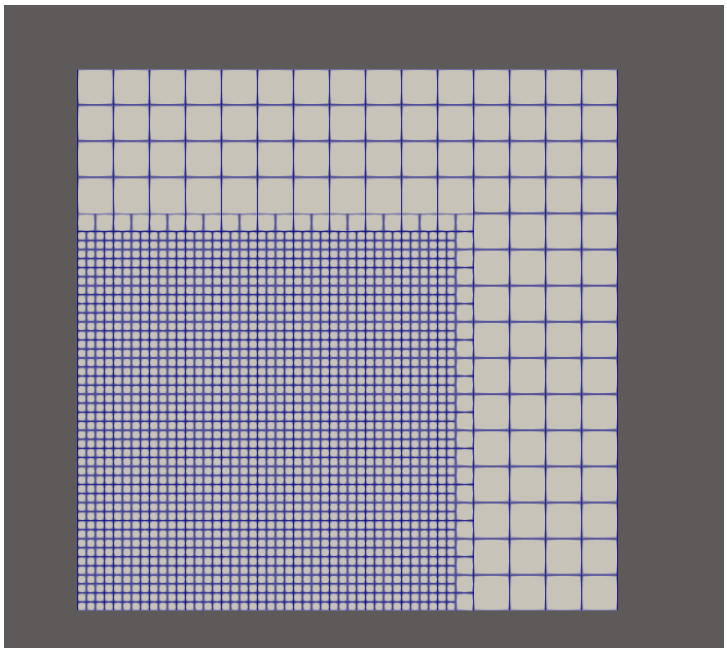
1. For simple coarse case



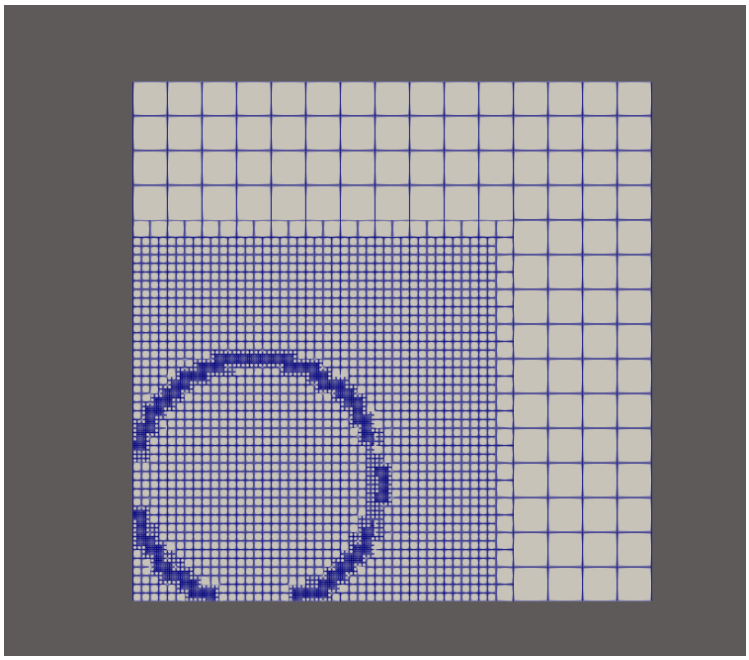
2. With only adaptive mesh case



3. With only local refinement



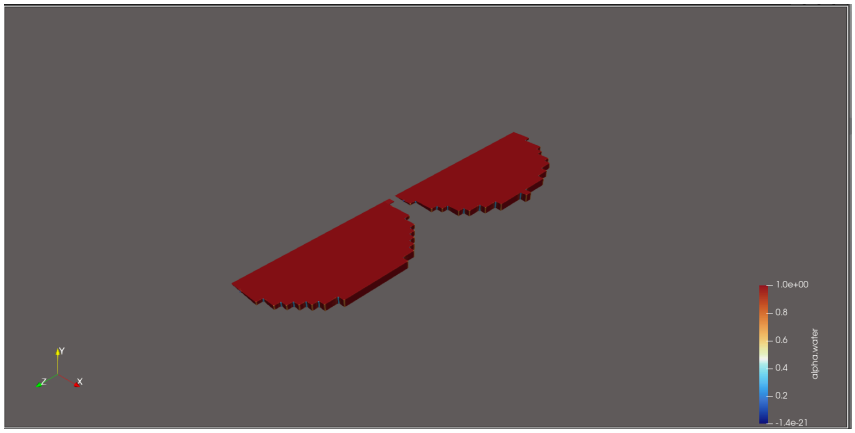
4. With local refinement and adaptive mesh refinement



NOTE this is just a depiction of how they will look like not exactly like the original files.

Learning about 2D-Axis symmetry in OpenFOAM

To optimize our case even more by neglecting gravity and considering symmetry around the axis considering the symmetry around the plane we optimize our case even more.
For reference consider this view:



This model is axis-symmetric about the z-axis and symmetric about the x-y plane about the origin which if constructed comes out similar to our initial programmed case but with a droplet nearby giving a collision.

So it is very efficient in computing the smaller mesh size which we need for bouncing.

The Github for same is named as 2drops

Important snippets from each document that are to be edited for different sample cases

1. **<u>blockMeshDict</u>**

Used to define the domain of interest. In our case the two walls make an angle of 5 degrees. To make any changes in the mesh make sure to make them proportional

```
vertices
(
 (0 0 0) //0
 (199.8096443 8.723877473 0) //1
 (199.8096443 8.723877473 400) //2
 (0 0 400) //3
 (199.8096443 -8.723877473 0) //4
 (199.8096443 -8.723877473 400) //5
);

blocks
(
    hex (0 4 1 0 3 5 2 3) (250 1 500) simpleGrading (1 1 1)
);
```

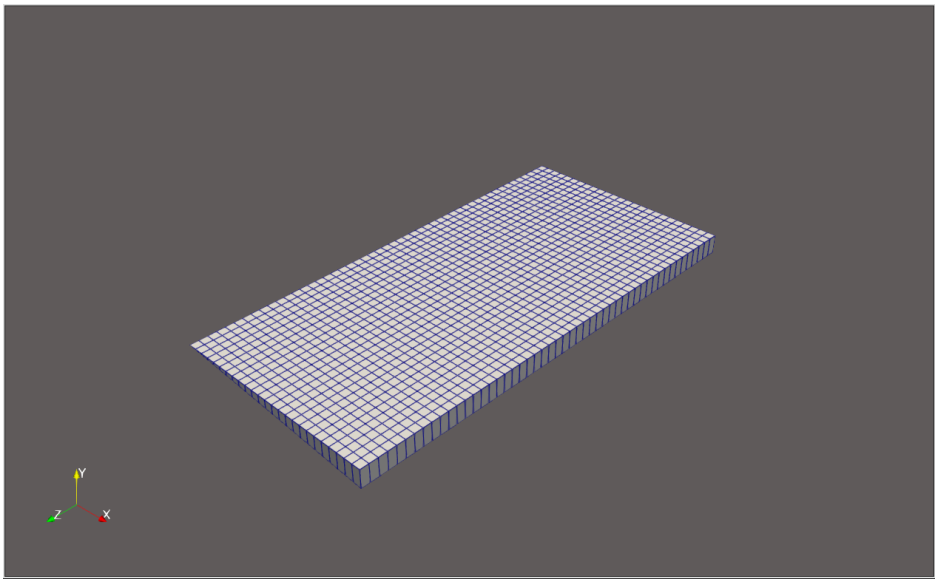
Instruction to execute the simulation:

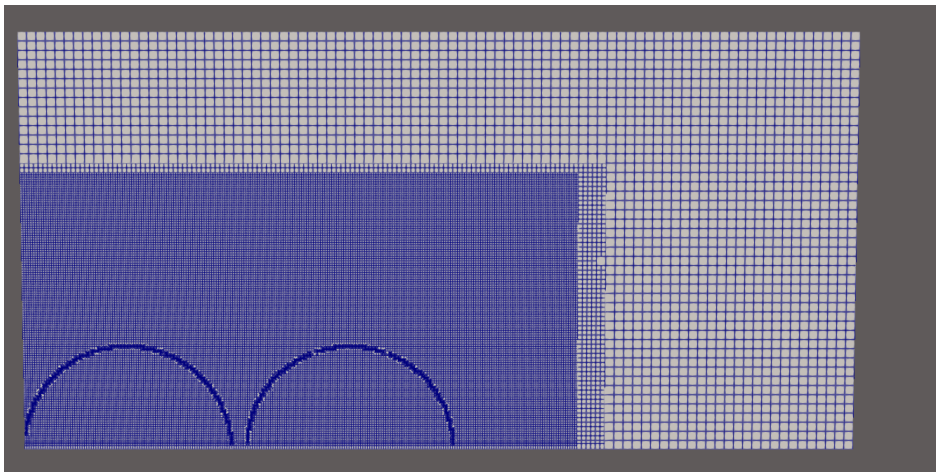
1. Open ubuntu
2. Go to case directory
3. Type the following:

```
openfoam2312
blockMesh
setFields
touch new.foam
decomposePar
mpirun -np 6 interFoam -parallel //change as per your requirement
reconstructParMesh
reconstructPar
```

Combining our knowledge of refinement and 2D-axisymmetric case

The difference between the normal 2d axisymmetric and refined 2d axisymmetric looks similar to following





The Github for the same is mentioned as bounceonit.

Instruction to execute the simulation:

1. Open ubuntu
2. Go to case directory
3. Type the following:

```
openfoam2312
blockMesh
topoSet -dict system/topoSetDict.outer
refineMesh
topoSet -dict system/topoSetDict.inner
refineMesh
```

4. After these steps copy the u alpha.water and p_rgh from 0 to latest time
5. Reopen the terminal and type the following

```
setFields
touch new.foam
decomposePar
mpirun -np 6 interFoam -parallel //change as per your requirement
reconstructParMesh
reconstructPar
```