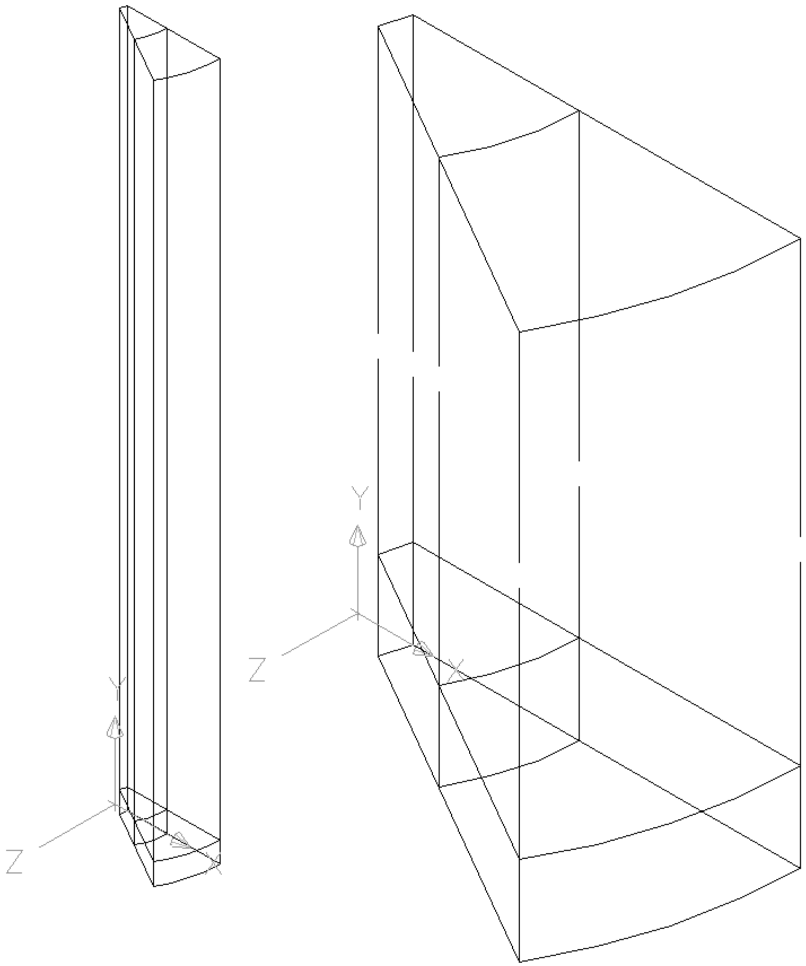
Natural Convection Boundary Layer

- Natural convective heat transfer from a hot cylincer to the surrounding air.
- The `buoyantSimpleFoam` solver is a steady state solver for buoyant turbulent, compressible flow used for ventilation and heat transfer.
- The axisymmetric nature of flow gives us a simplified field to solve.

Geometry

- The size of the field is $1.5m \times 0.6m \times 45^{\circ}$



Mesh Generation

- The blockMeshDict was modified to define 4 blocks with graded mesh and curved edges.The defined patches are as below:

```
wall pipeWall (
     (0 9 12 3)
     (3 12 15 6)
              )
wall tunnelWall (
     (5 8 17 14)
                )
wall downWall (
     (0 1 10 9)
     (1 2 11 10)
              )
```

```
cyclic cyclicRightAndLeft
 (
     (0 3 4 1)
     (1 4 5 2)
     (3 6 7 4)
     (4 7 8 5)
  (9 10 13 12)
  (10 11 14 13)
  (12 13 16 15)
  (13 14 17 16)
 )
 patch inlet
 (
     (2 5 14 11)
  )
 patch outlet
 (
     (6 15 16 7)
     (7 16 17 8)
  )
```

Boundary Condition

- Cylinder wall temperature is fixed to $T = 360$.
- Tunnel wall temperature is fixed to $T = 290$
- Top and bottom has the `zeroGradient` boundary condition.
- Velocity at all walls has a fixed zero value.
- All the properties on the cyclic boundaries are `cyclic` without any values.
- The turbulence properties(ϵ, k) are given a `zeroGradient` boundary condition.
- Because none of the boundary conditions for inlet and outlet hase converged, they also have been treated as wall boundary condition.

Initial Condition

- Temperature initial condition is fixed to a uniform temperature equal to the ambient temperature $T = 290$.
- Pressure is set to the reference pressure ($10^5 pa$).
- The velocity initial condition is also a uniform zero velocity field.
- The turbulence properties are calculated based on the maximum fluctuating velocity expected in the field considering a turbulence intensity of 10% as below:

$$U_{max} = 0.5\,\frac{m}{s} \implies u' = 0.05\,\frac{m}{s} \implies k = \frac{3}{2}(0.05)^2 = 0.00375\,\frac{m^2}{s^2}$$

and if we estimate the turbulent length scale as the maximum thickness of boundary layer $\approx 0.05m$. we can estimate the dissipation as below:

$$\epsilon = \frac{C_\mu^{0.75} k^{1.5}}{l} = 0.00075\,\frac{m^2}{s^3}$$

Discretization Scheme

The discretization scheme is set in the file `fvScheme` as follow:

- `ddtScheme` should be `steadyState`.
- `gradSchemes` is set to `linear` scheme.
- `divSchemes` are set to `Gauss upwind`.
- `laplacianSchemes` are selected as `linear`.
- The selected RAS turbulence model is k-Epsilon model with the coefficients as below:

```
Cmu              0.09;
C1               1.44;
C2               1.92;
C3               0.85;
alphah           1;
alphak           1;
alphaEps         0.76923;
```

- Velocity-pressure coupling method is `SIMPLE` method and the under-relaxation of the solution is required since the problem is steady.

Solving the Equations

- In `controlDict`, the time step `deltaT` should be set to 1 to act as a counter.
- The `endTime` is set to a big number to allow the solution to converge.
- The convergence can be monitored by the `pyFoamPlotWatcher.py`.
  We run this in a separate terminal window:

```
$ touch log
$ pyFoamPlotWatcher.py log
```

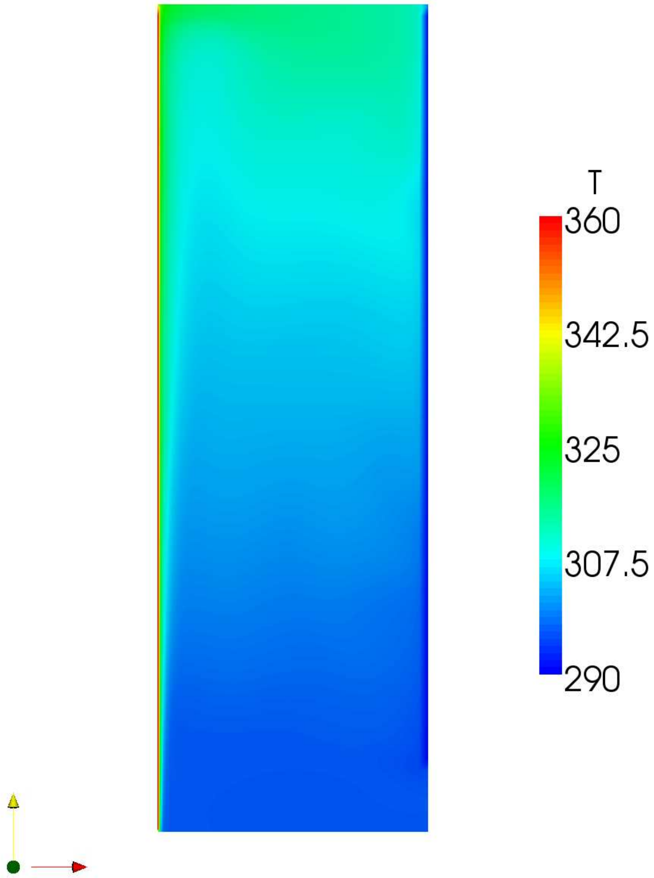
- The solver that we use in this simulation is `buoyantSimpleFoam`:

```
$ buoyantSimpleFoam >&log
```

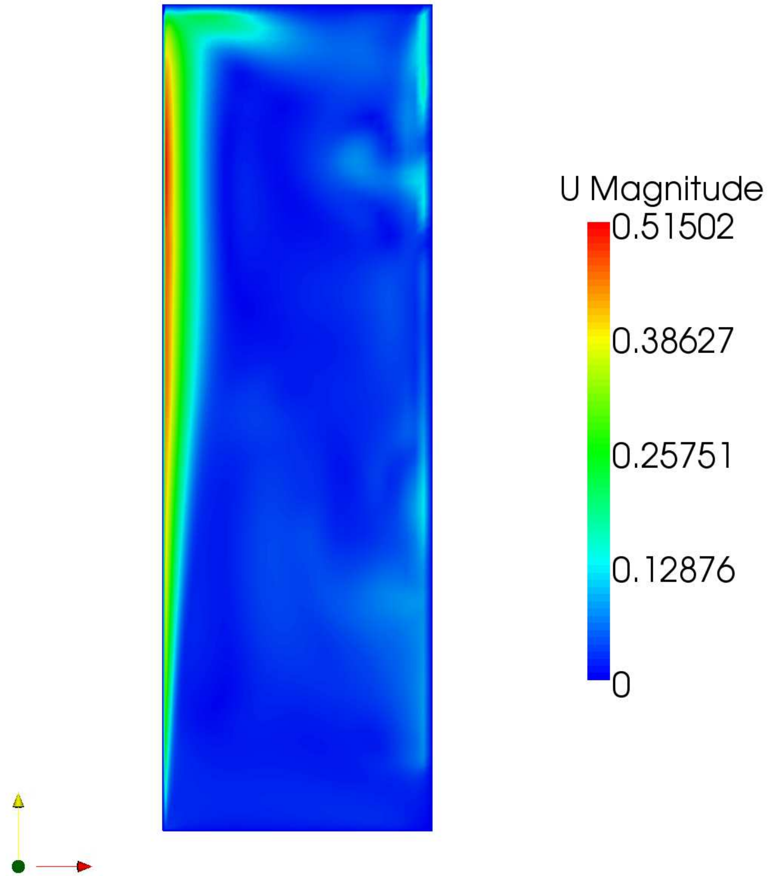
- First run was in a coarse mesh with 12800 cells and then the result mapped into a finer mesh with 48000 cells.

```
$ mapFields ../NC_coarse -consistent
```

Results



Results



Results