An application of Openfoam-Electrostatic Precipitator

Mr. Madan M Jagtap
Saraswati College of Engineering,Kharghar

Dr. B.B. Shrivastava
Independent Researcher

Prof. S.N.Teli
Associate Professor and Head Mechanical Engg. Dept.
Saraswati College of Engineering,Kharghar

Abstract:

This study is basically for flow analysis in ESP (Electrostatic Precipitator) as an application of OpenFOAM CFD (Computational Fluid Dynamics) tool, It introduces a new methodology and also reduce analysis cost. OpenFOAM as distributed and available to large extent, One can use available CFD code for different applications. Present flow analysis of ESP is one such application in porousSimpleFoam solver in OpenFOAM.

IJSER

Introduction:

An **electrostatic precipitator** (**ESP**), or **electrostatic air cleaner** is a particulate collection device that removes particles from a flowing gas (such as air) using the force of an induced electrostatic. Electrostatic precipitators are highly efficient filtration devices that minimally impede the flow of gases through the device, and can easily remove fine particulate matter such as dust and smoke from the air stream. It is very much required to analyze flow through electrostatic precipitator, Use of Openfoam code using available solver make analysis easy.

Methodology:

Developing Code For Application:

Using Initial Boundary Conditions:

Given initial boundary conditions are values of Epsilon,k,nut,P,T and U, OpenFOAM Code explains values of all these parameters at different Locations in ESP at t = 0,

For Epsilon:

```
dimensions      [ 0 2 -3 0 0 0 0 ];

internalField   uniform 1;

boundaryField
{
   wall
   {
      type          epsilonWallFunction;
      value          uniform 1;
   }
   ELECTRO
   {
     type          epsilonWallFunction;
     value          uniform 1;
   }
   ANTI_SNEAKAGE
   {
      type          epsilonWallFunction;
      value          uniform 1;
```

```
    }

    OUT2
    {
        type            inletOutlet;
        inletValue      uniform 1;
        value           uniform 1;
    }
    OUT1
    {
        type            inletOutlet;
        inletValue      uniform 1;
        value           uniform 1;
    }
    INLET
    {
        type
turbulentMixingLengthDissipationRateInlet;
        mixingLength    0.01;
        value           uniform 1;
    }
}
```

For k:

```
dimensions      [ 0 2 -2 0 0 0 0 ];

internalField   uniform 5.0508375;

boundaryField
{
    wall
    {
        type            kqRWallFunction;
        value           uniform 5.0508375;
    }
    ELECTRO
    {
        type            kqRWallFunction;
        value           uniform 5.0508375;
    }
    ANTI_SNEAKAGE
    {
         type            kqRWallFunction;
         value           uniform 5.0508375;
    }
    OUT2
    {
        type            inletOutlet;
        inletValue      uniform 5.0508375;
        value           uniform 5.0508375;
    }
    OUT1
    {
        type            inletOutlet;
        inletValue      uniform 5.0508375;
        value           uniform 5.0508375;
    }
    INLET
    {
        type
turbulentIntensityKineticEnergyInlet;
        intensity       0.05;
        value           uniform 5.0508375;
    }
}
```

For nut:

```
dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 1.545e-05;

boundaryField
{
    wall
    {
        type            nutkWallFunction;
        value           uniform 1.5450e-05;
    }
    ELECTRO
    {
        type            nutkWallFunction;
        value           uniform 1.5450e-05;
    }
    ANTI_SNEAKAGE
    {
        type            nutkWallFunction;
        value           uniform 1.5450e-05;
    }
```

IJSER

```
    OUT2
    {
        type            calculated;
        value           uniform 1.5450e-05;
    }
    OUT1
    {
        type            calculated;
        value           uniform 1.5450e-05;
    }
    INLET
    {
        type            calculated;
        value           uniform 1.5450e-05;
    }
}
```

For P:

```
dimensions      [0 2 -2 0 0 0 0];

internalField   uniform -2250;

boundaryField
{
    wall
    {
        type            zeroGradient;
    }
    ELECTRO
    {
        type            zeroGradient;
    }
    ANTI_SNEAKAGE
    {
    type                zeroGradient;
    }
    OUT2
    {
        type            fixedValue;
        value           $internalField;
    }
    OUT1
    {
        type            fixedValue;
        value           $internalField;
```

IJSER

```
    }
    INLET
    {
        type            zeroGradient;

    }
}
```

For T:

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

internalField   uniform 293;

boundaryField
{

     wall
    {
        type            zeroGradient;
    }
    ELECTRO
    {
        type            zeroGradient;
    }
    ANTI_SNEAKAGE
    {
        type            zeroGradient;
    }
     OUT2
    {
        type            inletOutlet;
        value           $internalField;
        inletValue      $internalField;
    }

    OUT1
    {
        type            inletOutlet;
        value           $internalField;
        inletValue      $internalField;

    }
    INLET
    {
        type            fixedValue;
        value           $internalField;
```

```
    }
}
```

For U:

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

internalField   uniform (0 0 0);

boundaryField
{
    wall
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    ELECTRO
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

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

    OUT2
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    OUT1
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
    INLET
    {
        type            fixedValue;
        value           uniform (0 12.05 0);
    }
}
```

Defining Boundary in Polymesh:
It may Vary based on Applications:

```
6
(
    wall
    {
        type            wall;
        nFaces          94288;
        startFace       1432637;
    }
    ELECTRO
    {
        type            wall;
        nFaces          166668;
        startFace       1526925;
    }
    ANTI_SNEAKAGE
    {
        type            wall;
        nFaces          700;
        startFace       1693593;
    }
    OUT2
    {
        type            patch;
        nFaces          99;
        startFace       1694293;
    }
    OUT1
    {
        type            patch;
        nFaces          99;
        startFace       1694392;
    }
    INLET
    {
        type            patch;
        nFaces          140;
        startFace       1694491;
    }
)
```

Defining Solver For Application:

```
application     porousSimpleFoam;
```

IJSER

```
startFrom      startTime;

startTime      0;

stopAt         endTime;

endTime        0.02;

deltaT         0.00005;

writeControl   timeStep;

writeInterval  1;

purgeWrite     0;

writeFormat    binary;

writePrecision 6;

writeCompression off;

timeFormat     general;

timePrecision  6;

graphFormat    raw;

runTimeModifiable true;
```

Defining fvSchemes :

```
ddtSchemes
{
   default        steadyState;
}

gradSchemes
{
   default        Gauss linear;
   grad(U)        Gauss linear;
   grad(p)        Gauss linear;
}

divSchemes
{
   div(phi,U)      Gauss upwind;
   div((nuEff*dev(T(grad(U)))))        Gauss
linear;
   div(phi,epsilon) Gauss upwind;
   div(phi,k)      Gauss upwind;
}

laplacianSchemes
{
   laplacian(nuEff,U)        Gauss        linear
uncorrected; //limited; //corrected;
   laplacian(rAU,p)        Gauss        linear
uncorrected; //limited; // corrected;
   laplacian(DepsilonEff,epsilon)        Gauss
linear uncorrected; // limited; // corrected;
   laplacian(DkEff,k)        Gauss        linear
uncorrected; //limited; // corrected;
   laplacian(1,p)   Gauss linear uncorrected;
//limited; // corrected;
}

interpolationSchemes
{
   default        linear;
}

snGradSchemes
{
   default        limited; //corrected;
}

fluxRequired
{
   default        no;
   p              ;

}
```

Defining fvSolutions

```
solvers
{
   p
   {
      solver         GAMG;
      tolerance       1e-08;
```

IJSER

```
        relTol          0.05;
        smoother        GaussSeidel;
        cacheAgglomeration off;
        nCellsInCoarsestLevel 20;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-06;
        relTol          0.1;
    }

    "(k|epsilon)"
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }
}

SIMPLE
{
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.7;
        k               0.9;
        epsilon         0.9;
    }
}
```

Paraview results: Graphic results have been obtained from PARAVIEW, It shows flow pattern obtained interns of streamlines, outer Mesh form of ESP and flow obtained from graphics part. This flow visualization explains internal flow distribution

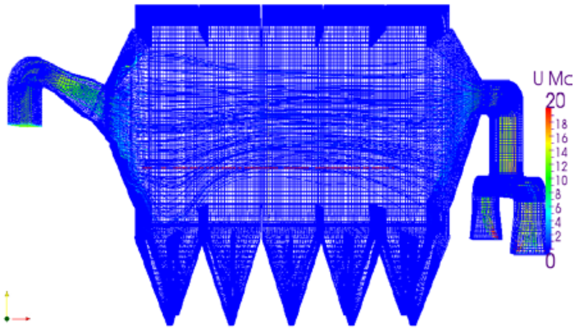


Fig.1 Velocity Variation

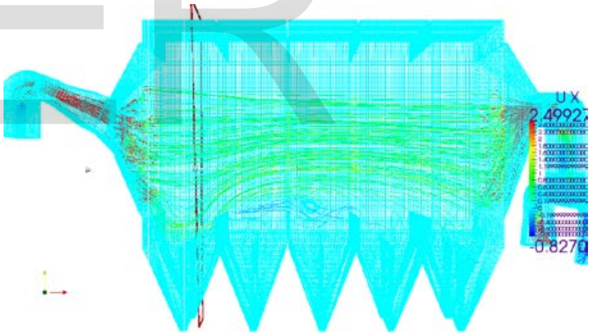


Fig.2 Velocity Variation in X direction

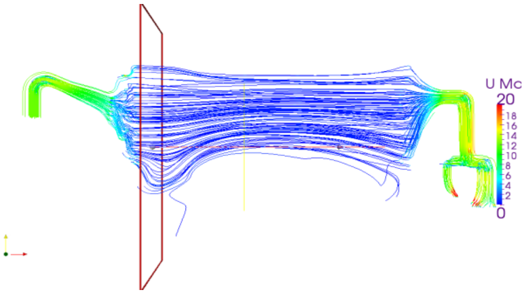


Fig.3 Stream Lines

IJSER

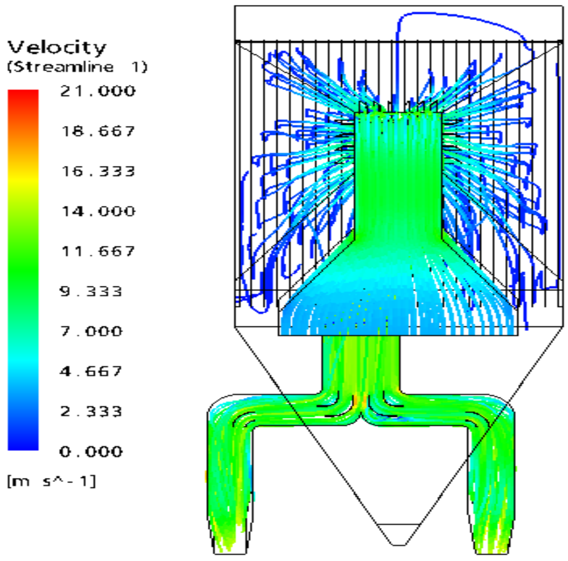
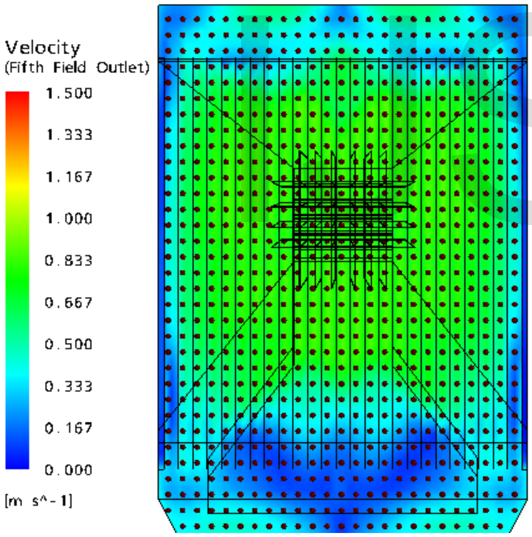


Fig.4 Side View of Velocity Field

Results:



Note: Only the velocity values inside the dotted block have been considered for RMS calculations

RMS values for each field is below 25%

AVERAGE VELOCITY	0.6877
R.M.S	24.41
1.15 X AVG VEL	0.7909
1.4 X AVG VEL	0.9628
POINTS BELOW 1.15 X AVG VEL	323
POINTS BELOW 1.4 X AVG VEL	550
TOTAL POINTS	550
% POINTS BELOW 1.15 X AVG VEL	58.72
% POINTS BELOW 1.4 X AVG VEL	100

References:

[1] M.G. Rasul, W. Widianto, B. Mohanty, Assessment of the thermal performance and energy conservation opportunities in cement industry in Indonesia, Applied Thermal Engineering 25 (2005) 2950–2965.

[2] M.G. Rasul, B.S. Tanty, B. Mohanty, Modeling and analysis of blast furnace performance for efficient utilization of energy, Applied Thermal Engineering 27 (2007) 78–88.

[3] Anatol Jaworek, Andrzej Krupa, Tadeusz Czech, Modern electrostatic devices and methods for exhaust gas cleaning: a brief review, Journal of Electrostatics 65 (2007) 133–155

[4] L. Zhao, E. Dela Cruz, K. Adamiak,, A.A. Berezin, J.S. Chang, A numerical model of a wire-plate electrostatic precipitator under electro hydrodynamic flow conditions, in: Conference proceedings. The 10th International Conference on Electrostatic Precipitator, Australia, 2006.

[5] G. Skodras, S.P. Kalidas, D. Sofialidis, O. Faltsi, P. Grammelis, G.P. Sakellaropoulos, Particulate removal via electrostatic precipitators – CFD simulation, Fuel Processing Technology 87 (2006) 623–631.

[6] Y. Zhuang, Y.J. Kim, T.G. Lee, P. Biswas, Experimental and theoretical studies of ultra-fine particle behavior in electrostatic precipitators, J. Electrostatics 48 (2000) 245–260.

[7] S.H. Kim, K.W. Lee, Experimental study of electrostatic precipitator performance and comparison with existing theoretical prediction models, J. Electrostatics 48 (1999) 3–25.

[8] W. Deutsch, Bewegung und Ladung der Elektrizitastrager im Zylinderkondensator, Ann. Phys. 68 (1922) 335–344.

[9] G. Cooperman, A new theory of precipitator efficiency, Atmos. Environ. 5 (1971) 541–551.

[10] G.L. Leonard, M. Mitchner, S.A. Self, Experimental study of the effect of turbulent diffusion on precipitator efficiency, J. Aerosol. Sci. 13 (1982) 271–284.

IJSER