

Hands-On Training with OpenFOAM

Flow Around a 2-D Airfoil

Hrvoje Jasak

`h.jasak@wikki.co.uk`

Wikki Ltd, United Kingdom

Summary of Objectives: Basic Code and Case Structure

- Basic structure of the OpenFOAM case and CFD runs

Tutorial Steps

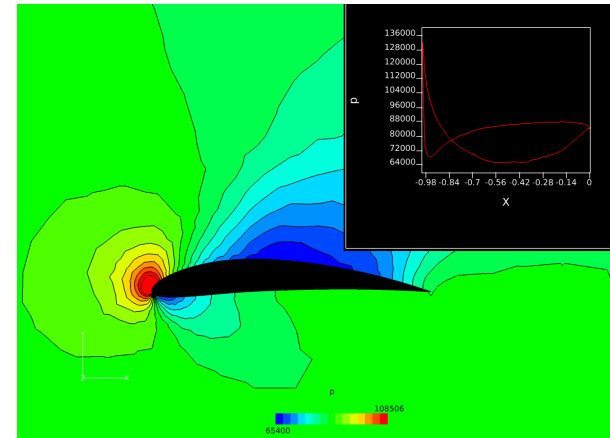
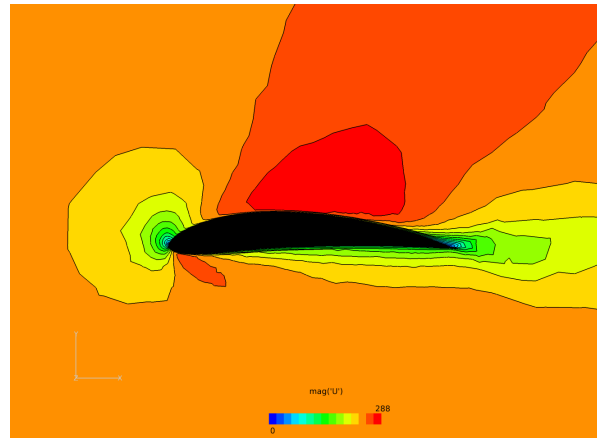
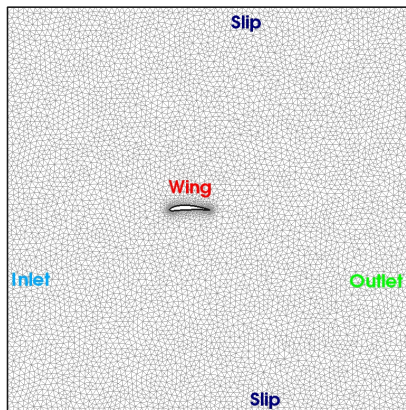
1. Basic review of case organisation
2. Convert the mesh generated in Pointwise
3. `constant/polyMesh/boundary`: set wall patch type
4. Set material properties: viscosity; turbulence model
5. 0 directory: set initial and boundary conditions for flow fields
6. `checkMesh`: analysis of mesh quality
7. Flow solver: `sonicFoam`
8. Basic post-processing with FieldView
9. Utilities and data manipulation: `Mach` number, `forceCoeffs` function object
10. Further post-processing
11. Basic review of solver and discretisation parameters

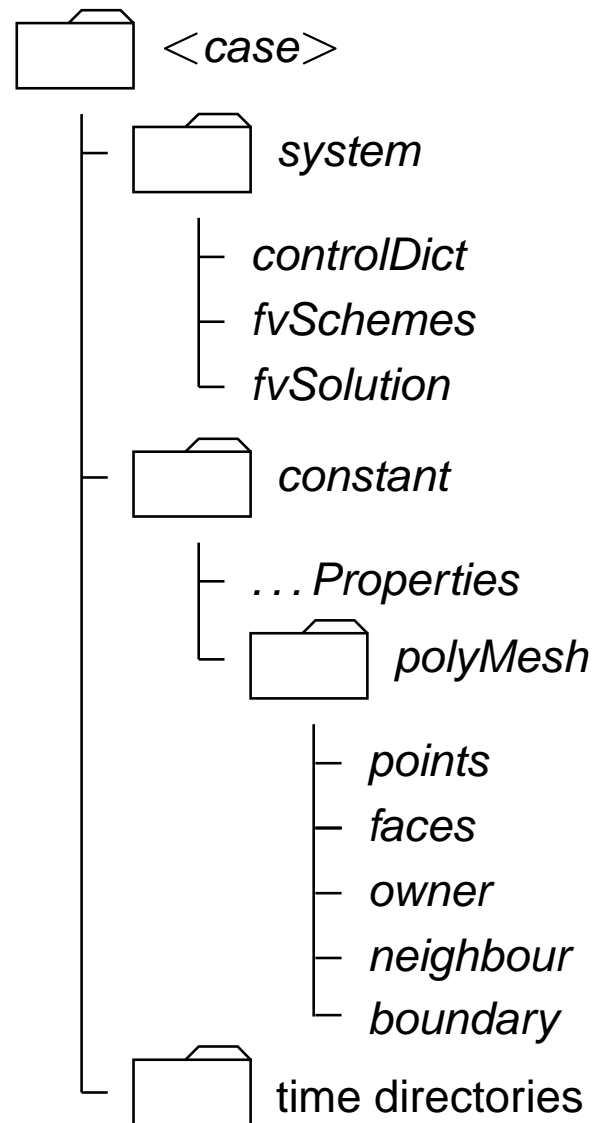
Case Setup

- Transient compressible turbulent flow simulation
- Material properties: air as ideal gas with constant properties
(name, nMoles, molWeight, Cv, Hf, mu, 1/Pr)
- Inlet/far field conditions:

$$\mathbf{u} = (242.43 \quad 0 \quad 0) \text{ m/s} \quad p_{inf} = 85419 \text{ Pa} \quad T = 260 \text{ K}$$

$$k = 220.4 \text{ m}^2/\text{s}^2 \quad \epsilon = 2688.26 \text{ m}^2/\text{s}^3$$





Data Organisation and Management

- Unlike “standard CFD practice”, in OpenFOAM `case` is a **directory**: each self-contained piece of heavy-weight data stored in its own file
- Light-weight data is presented in **dictionary form**: keyword-value pairs in free format. It can be changed and re-read during the run: solution steering
- Mesh data split into components for efficient management of moving mesh cases
- Time directories contain solution and derived fields (one per file)
- Support for compressed I/O: more efficient I/O and less disk space

Data Input and Output

- Dictionary format: file header (IOobject) and keyword-value entry pairs

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       transportProperties;
}

// Diffusivity
DT [0 2 -1 0 0 0 0] 0.01;
```

- Contents of dictionaries depends on their role
 - Material properties and physical model constants
 - Solution fields, initial and boundary conditions
 - Discretisation settings, solver controls I/O parameters etc.

Basic Controls: controlDict

- Basic controls of run start, end and write frequency

```
startFrom      startTime; // latestTime // firstTime
startTime      0;

stopAt         endTime;   // writeNow // nextWrite
endTime        2500;

deltaT         1;

writeControl    timeStep; // runTime // clockTime // cpuTime
writeInterval   50;

writeFormat     ascii;    // binary
writePrecision  6;
writeCompression uncompressed; // compressed

timeFormat      general;  // fixed // scientific
timePrecision   6;

runTimeModifiable yes;
```

Basic Controls: fvSchemes

- Equation discretisation controls: per-term basis

```
ddtSchemes
{
    default steadyState;
}
gradSchemes
{
    default          cellLimited leastSquares 1.0;
    //      grad(p)      Gauss linear;
}
divSchemes
{
    default          none;
    div(phi,U)       Gauss linearUpwindV Gauss linear;
    div(phi,k)       Gauss upwind;
    div(phi,omega)   Gauss upwind;
    div((nuEff*dev(grad(U).T()))) Gauss linear;
}
```

Basic Controls: fvSchemes

- Equation discretisation controls, Cont'd

```
laplacianSchemes
{
    default          Gauss linear limited 0.5;
}
interpolationSchemes
{
    default          linear;
    interpolate(U)    linear;
}
snGradSchemes
{
    default          limited 0.5;
}
fluxRequired
{
    default          no;
    p;
}
```


Basic Controls: fvSolution

- Linear equation solver settings: per equation in top-level solver

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner   DIC;

        tolerance       1e-8;
        relTol           0.01;
    }
    U
    {
        solver          PBiCG;
        preconditioner   DILU;

        tolerance       1e-07;
        relTol           0;
    }
}
```

Basic Controls: fvSolution

- Global algorithmic settings and under-relaxation factors

```
PISO
{
    momentumPredictor yes;

    nCorrectors      2;
    nNonOrthogonalCorrectors 0;

    nAlphaCorr      1;
    nAlphaSubCycles 2;
    cAlpha           1;
}
relaxationFactors
{
    p              0.3;
    U              0.7;
    k              0.7;
    omega          0.7;
}
```

Basic Controls: Structure of Mesh Files

- Mesh files at start of simulation located in `constant/polyMesh` directory
- `points`, `faces`: basic lists of primitive entries
- `owner`, `neighbour`: lists of face-to-cell addressing
- Note: **OpenFOAM uses strongly ordered face lists** for efficiency

```
11M 2011-07-22 11:19 points
37M 2011-07-22 11:19 faces
4.0M 2011-07-22 11:19 owner
14M 2011-07-22 11:19 neighbour
1.8K 2011-07-22 11:19 boundary
```

- Additional mesh files: sets and zones, mesh modifiers, parallel mapping etc.

```
586 2011-03-14 09:43 pointZones
24K 2011-03-14 09:43 faceZones
868K 2011-03-14 09:43 cellZones
78K 2011-03-14 09:43 meshModifiers
4.0K 2011-03-14 09:43 sets/
```

Basic Controls: Structure of Mesh Files

- Boundary definition: patch types and strong ordering

```
(
    Wing                                <-- patch name
    {
        type                            wall;    <-- constrained patch type
        nFaces                          154;    <-- number of faces
        startFace                       23579;  <-- start face in face list
    }
    FrontAndBack
    {
        type                            empty;   <-- constrained patch type
        nFaces                          30712;
        startFace                       23733;
    }
    Inlet
    {
        type                            patch;    <-- (free) patch type
        nFaces                          74;
        startFace                       54445;
        ...
    }
)
```

Basic Controls: Structure of Mesh Files

- Boundary definition: patch types

```
...
    Slip
    {
        type                patch;
        nFaces              148;
        startFace           54519;
    }
    leftPlane
    {
        type                symmetryPlane;
        nFaces              74;
        startFace           54667;
    }
}
```

Field: Initial and Boundary Conditions

- Definition of initial and boundary conditions, per-field basis

```
dimensions      [0 1 -1 0 0 0 0]; // [kg m s K mol A Cd]
internalField    uniform (40 0 0);
boundaryField
{
    Body-4
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
    Inlet-12
    {
        type          fixedValue;
        value          uniform (40 0 0);
    }
    Slip-10
    {
        type          slip;
    }
    ...
}
```

Field: Initial and Boundary Conditions

- Definition of initial and boundary conditions, Cont'd
- Fields located in time directories: $0/p$, $0/U$
- Boundary conditions defined on a **per-field basis**
- Note: consistency of boundary conditions related to the physics solver

Basic Controls: Utility Controls

- Utility controls based under `system`. Example:

```
numberOfSubdomains 4;
```

```
method            metis;
```

```
globalFaceZones ( insideZone outsideZone );
```

```
simpleCoeffs
```

```
{  
    n            (4 1 1);  
    delta        0.0001;  
}
```

```
metisCoeffs
```

```
{  
    processorWeights 4( 1 1 1 1 );  
}
```

```
roots ( );
```