Solvers - incrompressible

Advanced course

Content

- Solver types
- Model properties
  - Transport properties
  - Turbulence properties
  - RAS properties
- Boundary conditions
- Solver properties
  - fvSchemes
  - fvSolution
  - controlDict

Solver types

	Laminar	Turbulent
Steady state	simpleFoam (simulationType: laminar)	simpleFoam (simulationType: RASModel)
Unsteady state (time dependent)	icoFoam	pisoFoam pimpleFoam

web: http://www.openfoam.org/archive/1.7.0/docs/user/standard-solvers.php

Solver types

- IcoFoam
  - Ico: incompressible
- simpleFoam
  - Semi-Implicit Method for Pressure Linked Equations
  - Web: http://en.wikipedia.org/wiki/SIMPLE_algorithm
- pisoFoam:
  - Pressure Implicit with Splitting of Operator
  - Web: http://en.wikipedia.org/wiki/PISO_algorithm
- pimpleFoam
  - merged piso-simple

Solver types

- Next slides: example files based on simpleFoam solver
- Files can be reused for new geometries of the propeller case, without modification

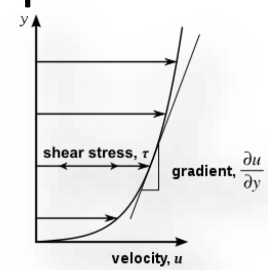
Model properties

- Transport properties
  - Location: \constant\transportProperties

```
transportModel   Newtonian;

nu               nu [ 0 2 -1 0 0 0 0 ] 1e-6;
```

  - simpleFoam:
    - For incompressible flow: $\rho = constant$
    - Therefore, only the kinematic viscosity is required
      - Dynamic viscosity μ: $\tau = \mu \cdot \frac{\partial u}{\partial y}$
      - Kinematic viscosity ν: $\nu = \frac{\mu}{\rho}$



    - Consequence: pressure, force, torque, ... results need to be multiplied by the density to obtain the real value

Model properties

- Turbulence properties
  - Location: \constant\turbulenceProperties

```
simulationType  RASModel;
```

  - simpleFoam:
    - works with the RASModel – Reynolds-Averaged Simulation
    - Type of RASModel is defined under “RASProperties”

Model properties

- RASProperties
  - Location: \constant\RASProperties

```
RASModel        kEpsilon;

turbulence      on;

printCoeffs     on;
```

  - Overview: http://www.openfoam.org/features/RAS.php
  - Some examples:
    - laminar: dummy model for laminar flow
    - k-epsilon: most commonly used turbulence model
    - K-omega
  - Chosen model has impact on required boundary parameters

Boundary conditions

- Location: \0
- K: turbulent kinetic energy

```
rotor
{
    type            kqRWallFunction;
    value           uniform 0.06;
}
```

- Epsilon: rate of dissipation of the turbulent energy

```
rotor
{
    type            epsilonWallFunction;
    value           uniform 0.0495;
}
```

- Nut: turbulent viscosity

```
rotor
{
    type            nutkWallFunction;
    value           uniform 0;
}
```

Solver properties

- fvSchemes: schemes for the finite volume approach

```
ddtSchemes
{
    default         steadyState;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss limitedLinearV 1;
    div(phi,k)      bounded Gauss limitedLinear 1;
    div(phi,epsilon) bounded Gauss limitedLinear 1;
    div((nuEff*dev(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

fluxRequired
{
    default         no;
    p               ;
}
```

Time derivative

gradient

Divergence (including turbulence parameters)

Laplacian operator

interpolation

Surface normal gradient

Flux calculation

Solver properties

- fvSolution: settings for the iterative solver

```
solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-08;
        relTol          0.05;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 20;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }

    k
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }

    epsilon
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        nSweeps         2;
        tolerance       1e-07;
        relTol          0.1;
    }
}
```

Pressure

Velocity

Turbulence intensity

Dissipation of turbulent energy

Solver properties

- fvSolution: settings for the iterative solver

```
SIMPLE
{
    nNonOrthogonalCorrectors 0;
    pRefCell          0;
    pRefValue         0;
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.5;
        k               0.5;
        epsilon         0.5;
    }
}
```

Relaxation factors

Reference pressure for the domain

Required when domain is closed

Boundary conditions

- controlDict

```
application     simpleFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         500;

deltaT          1;

writeControl    timeStep;

writeInterval   50;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

functions
{
    #include "forceCoeffs"
    #include "forces"
//    #include "wallBoundedStreamLines"
}
```

Solver type

(pseudo) Time control

Write control

Allow changes in control files during execution

Include extra output functions