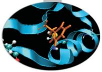
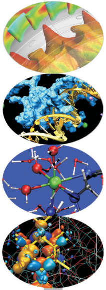


OpenFOAM selected solver



Roberto Pieri - *SCS Italy*

16-18 June 2014

Introduction to Navier-Stokes equations and RANS

Turbulence modelling

Numeric discretization

Navier-Stokes equations

$$\begin{cases} \overbrace{\nabla \cdot (\mathbf{U} \otimes \mathbf{U})}^{\text{Convective term}} - \underbrace{\nabla \cdot \nu \nabla \mathbf{U}}_{\text{Viscous term}} = -\frac{1}{\rho}\nabla P \\ \nabla \cdot \mathbf{U} = 0 \end{cases}$$

- Equations are directly derived from conservation laws.
- $\mathbf{U}$ is the velocity vector, P is pressure and ρ is density.
- System of partial differential equations.
- Equations are valid for viscous, incompressible, steady flows in laminar regime.

Reynolds Averaged Navier-Stokes

RANS equations

$$
\begin{cases}
\nabla \cdot (\overline{\mathbf{U}} \otimes \overline{\mathbf{U}}) + \underbrace{\nabla \cdot (\mathbf{U'} \otimes \mathbf{U'})}_{\text{Reynolds' stresses tensor}} - \nabla \cdot \nu \nabla \overline{\mathbf{U}} = -\nabla \overline{p} \\
\nabla \cdot \overline{\mathbf{U}} = 0
\end{cases}
$$

- $\mathbf{U} = \overline{\mathbf{U}} + \mathbf{U'}$.
- Equations are obtained decomposing velocity vector and averaging.
- The term $\nabla \cdot (\mathbf{U'} \otimes \mathbf{U'})$ represents a new unknown.
- A closure equation is required.
- $\overline{p} = \overline{P}/\rho$, it is only a mathematical function (equation of state is not present).

Turbulence modelling

There are two different class of models:

Eddy-viscosity models

- Based on Boussinesq hypotesis
- Very large number of models
- Different models for different flow conditions

Reynolds stress models

- More recent
- Equations for every term of Reynolds' stress tensor are required

We are going to discuss the first class of models.

Turbulence modelling

Eddy-viscosity models (I)

- Effective viscosity ν_e is defined as follow:

$$\nu_e\left(\mathbf{x}\right) = \nu + \nu_t\left(\mathbf{x}\right)$$

- Reynolds' stress tensor can be rewritten as follow:

$$\nabla \cdot \left(\mathbf{U'} \otimes \mathbf{U'}\right) = \nabla \cdot \left(\nu_t\left(\mathbf{x}\right)\nabla\overline{\mathbf{U}}\right)$$

- Momentume equation can be rewritten:

$$\nabla \cdot \left(\overline{\mathbf{U}} \otimes \overline{\mathbf{U}}\right) - \nabla \cdot \nu_e \nabla\overline{\mathbf{U}} = -\nabla\bar{p}$$

Turbulence modelling

Eddy-viscosity models (II)

The new system of equations is:

$$\begin{cases} \nabla \cdot \left(\overline{\mathbf{U}} \otimes \overline{\mathbf{U}} \right) - \nabla \cdot \nu_e \left(\mathbf{x} \right) \nabla \overline{\mathbf{U}} = -\nabla \bar{p} \\ \nabla \cdot \overline{\mathbf{U}} = 0 \end{cases}$$

with

$$\nu_e \left(\mathbf{x} \right) = \nu + \nu_t \left(\mathbf{x} \right)$$

- In this formulation the model is totally confinated in $\nu_t \left(\mathbf{x} \right)$.
- A model for the effective viscosity is needed.

Turbulence modelling

Eddy-viscosity models (III)

Eddy-viscosity models are divided in three classes depending on the number of differential equations needed for the closure of the problem.

- 0-equation models (mixing length).
- 1-equation models (Spalart-Allmaras, k equation, ...).
- 2-equations models ($k - \varepsilon$, $k - \omega$, ...).

Turbulence modelling

Eddy-viscosity models (IV)

An example of a 2-equation model is $k - \omega$.

- An equation for k is needed.
- An equation for ω is needed.
- The model is complete:

$$\nu_t = C_\mu \frac{k}{\omega}$$

where C_μ is a constant (possible tuning).

OpenFOAM solvers

- Large number of solvers.
- Choose the solver that best suits your case study (compressible/incompressible, heat transfer, multiphase...).
- A first setup is always given by the tutorials.
- Attention: tutorials' setup may not work for your case.

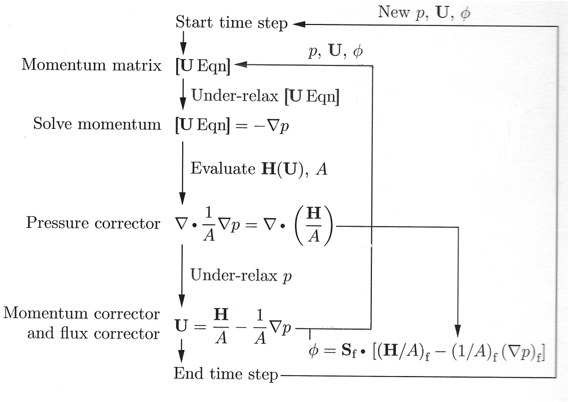
One of the most used solvers is *simpleFoam*.

OpenFOAM solvers

Semi-Implicit Method for Pressure-Linked Equations (simpleFoam)

- Suitable for incompressible, steady-state, viscous flows in laminar or turbulent regime.
- Used for internal and external flows.
- Very large documentation and test cases from the user community.

OpenFOAM solvers



SIMPLE algorithm

OpenFOAM solvers

SIMPLE implementation in OpenFOAM

```
solve
(
    fvm::div(phi, U)
  + turbulence->divDevReff(U)
  ==
  - fvc::grad(p)
);
```

- Top level code represents the equations being solved.
- *OpenFOAM* has functions for derivatives. *e.g.* div, grad, laplacian, curl.
- fvc:: returns a field, it is used to calculate the pressure gradient with current values (explicit).
- fvm:: returns an fvMatrix, it is used in order to discretise a term into matrix equation you wish to solve (implicit).
- solve function solves the equation.

OpenFOAM solvers

Other solvers

- *pisoFoam*: transient solver for incompressible flow;
- *pimpleFoam*: merged PISO-SIMPLE
  - can run transient; no Courant number limited, unlike PISO;
  - can run pseudo-transient: big time step to reach *steady-state* with minimal under-relaxation;
  - can be used in substitution of SIMPLE, gaining in stability of the solver.
- *buoyantBoussinesqSimpleFoam*: steady-state solver for buoyant, turbulent flow of incompressible fluids including Boussinesq approximation for stratified flow

$$\varrho_k = 1 - \beta \left(\bar{T} - T_0 \right)$$