NAZARBAYEV UNIVERSITY
SCHOOL OF ENGINEERING AND DIGITAL SCIENCES

How to implement VLES kOmegaSST turbulence model in OpenFOAM

Bagdaulet Kamalov
Mechanical and Aerospace Engineering, MSc
School of Engineering and Digital Sciences
Supervisor: Prof. Yong Zhao

Overview

- Introduction to Hybrid RANS-LES method
- Resolution Control Function, F_r
- SST k-Omega turbulence model
- Resolution Control Function, F_r with model constants
- Copying source files to local directory
- Modifying copied files
- Running the VLES kOmegaSST turbulence model
- Results

Hybrid RANS-LES method

- Hybrid turbulence modeling which combines different turbulent models becoming more important.
- Computationally not expensive as Large Eddy Simulation (LES) but more accurate than Reynolds-Averaged Navier-Stokes (RANS).
- By rescaling the conventional RANS equations through the introduction of a resolution control function F_r into the turbulent viscosity, the model can be considered as Very Large Eddy Simulation (VLES).

Resolution Control Function F_r

$$F_r = min\left(1.0, \left[\left(1.0 - exp\left(-\beta L_c / L_k\right)\right) / \left(1.0 - exp\left(-\beta L_i / L_k\right)\right)\right]^n\right)$$

$$L_c = C_x \left(\Delta_x \Delta_y \Delta_z\right)^{\frac{1}{3}} \quad L_i = k^{\frac{3}{2}} / \varepsilon \quad L_k = \nu^{\frac{3}{4}} / \varepsilon^{\frac{1}{4}}$$

Where L_c ,L_i and L_k are the turbulent cutoff length scale, integral length scale and Kolmogorov length scale, respectively, and Δ_x, Δ_y and Δ_z are mesh scales in different directions, ν is the laminar viscosity.

Recommended values for n and β are $n = \frac{4}{3}$ and $n = 2$ and $\beta = 2.0\times10^{-3}$ based on the studies of the Speziale [1].

$$C_x = \sqrt{0.3} C_s / C_\mu$$

The model constant C_x can be calibrated using $C_\mu = 0.09$ which model constant of the turbulence model (in this case SST k-omega model) and $C_s = 0.1$ which is typical Smagorinsky LES model constant [2].

$$C_x = 0.61$$

$$\mu_t^{sub} = F_r \mu_t^{RANS}$$

Addition of Resolution Control Function for any RANS turbulence model.

$$\nu_t = a_1 \frac{k}{\max(a_1 \omega, b_1 F_{23} \mathbf{S})}$$

Turbulence viscosity of the SST k-omega from OpenFOAM: User Guide.

$$\mu_t = F_r \nu_t$$

Turbulence viscosity scaled by Resolution Control Function.

SST k-omega turbulence model

Model equations

The turbulence specific dissipation rate equation is given by:

$$\frac{D}{Dt}(\rho\omega) = \nabla\cdot(\rho D_\omega \nabla\omega) + \frac{\rho\gamma G}{\nu} - \frac{2}{3}\rho\gamma\omega(\nabla\cdot\mathbf{u}) - \rho\beta\omega^2 - \rho(F_1 - 1)CD_{k\omega} + S_\omega,$$

and the turbulence kinetic energy by:

$$\frac{D}{Dt}(\rho k) = \nabla\cdot(\rho D_k \nabla k) + \rho G - \frac{2}{3}\rho k(\nabla\cdot\mathbf{u}) - \rho\beta^*\omega k + S_k.$$

The turbulence viscosity is obtained using:

$$\nu_t = a_1 \frac{k}{\max(a_1\omega, b_1 F_{23}\mathbf{S})}$$

$\varepsilon = 0.09k\omega$ Relation between epsilon and omega.

Resolution Control Function, F_r with model constants

$$F_r = min\left(1.0, \left[\left(1.0 - exp\left(-\beta L_c/L_k\right)\right)/\left(1.0 - exp\left(-\beta L_i/L_k\right)\right)\right]^n\right)$$

$$F_r = \min(1.0, [(1.0 - \exp(-\beta L_c/L_k)/(1.0 - \exp(-\beta L_i/L_k)]^{4/3})$$

$$F_r = \min(1.0, \left[\left(1.0 - \exp(0.09)^{\frac{1}{4}} \beta C_x (\Delta_x \Delta_y \Delta_z)^{\frac{1}{3}} (k\omega)^{\frac{1}{4}} / \nu^{\frac{3}{4}}\right) / \left(1.0 - \exp(-\frac{\beta}{(0.09)^{\frac{3}{4}}} \left(\frac{k}{\nu\omega}\right)^{\frac{3}{4}}\right)\right]^{4/3})$$

Copying source files to local directory

- First create a folder with the name mykOmegaSST then copy OpenFOAM source files to the created folder and it can be done by following commands.

```
cp -r $FOAM_SRC/TurbulenceModels/turbulenceModels/RAS/kOmegaSST mykOmegaSST
cd mykOmegaSST
mv kOmegaSST.H mykOmegaSST.H
mv kOmegaSST.C mykOmegaSST.C
cd ..
cp -r $FOAM_SRC/TurbulenceModels/turbulenceModels/Base/kOmegaSST/kOmegaSSTBase.C mykOmegaSST
cp -r $FOAM_SRC/TurbulenceModels/turbulenceModels/Base/kOmegaSST/kOmegaSSTBase.H mykOmegaSST
cd mykOmegaSST
mv kOmegaSSTBase.H newkOmegaSSTBase.H
mv kOmegaSSTBase.C newkOmegaSSTBase.C
cp -r
$FOAM_SRC/TurbulenceModels/incompressible/turbulentTransportModels/turbulentTransportModels.C
maketurbulentTransportModels.C
cp -r $FOAM_SRC/TurbulenceModels/incompressible/Make/ .
```

Modifying copied files

- In this step, copied files will be modified. First, `maketurbulentTransportModels.C` file will be changed as following.

```
#include "mykOmegaSST.H"
```

```
makeRASModel(mykOmegaSST);
```

- Your edited file should look like picture shown in the right.

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Copyright (C) 2013-2019 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.

\*---------------------------------------------------------------------------*/

#include "turbulentTransportModels.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

makeBaseTurbulenceModel
(
    geometricOneField,
    geometricOneField,
    incompressibleTurbulenceModel,
    IncompressibleTurbulenceModel,
    transportModel
);

// -------------------------------------------------------------------------- //
// RAS models
// -------------------------------------------------------------------------- //

#include "mykOmegaSST.H"
makeRASModel(mykOmegaSST);

// ************************************************************************* //
```

Modifying copied files

- Delete all the statements and add following lines to the Make/Files.

```
maketurbulentTransportModels.C

LIB = $(FOAM_USER_LIBBIN)/libmyincompressibleTurbulenceModels
```

- As well as, for Make/Options do the same (delete all the statements) and add following lines.

```
EXE_INC = \
    -I$(LIB_SRC)/TurbulenceModels/turbulenceModels/lnInclude \
    -I$(LIB_SRC)/TurbulenceModels/incompressible/lnInclude \
    -I$(LIB_SRC)/transportModels \
    -I$(LIB_SRC)/finiteVolume/lnInclude \
    -I$(LIB_SRC)/meshTools/lnInclude \

LIB_LIBS = \
    -lincompressibleTransportModels \
    -lturbulenceModels \
    -lfiniteVolume \
    -lmeshTools
```

Modifying copied files

- Now, mykOmegaSST.C, mykOmegaSST.H, newkOmegaSSTBase.C and newkOmegaSSTBase.H files will be modified. First all the occurrences of kOmegaSST will be changed to newkOmegaSST and mykOmegaSST as following.

```
sed -i s/kOmegaSST/mykOmegaSST/g mykOmegaSST.C
sed -i s/kOmegaSST/mykOmegaSST/g mykOmegaSST.H
sed -i s/kOmegaSSTBase/newkOmegaSSTBase/g newkOmegaSSTBase.C
sed -i s/kOmegaSSTBase/newkOmegaSSTBase/g newkOmegaSSTBase.H
```

- Also, occurences of kOmegaSST in kOmegaSSTBase.* have to be changed to mykOmegaSST and occurences of kOmegaSSTBase in kOmegaSST.* have to be changed to newkOmegaSST manually, or like the code shown below.

```
sed -Ei '/(kOmegaSSTBase|mykOmegaSST|dummy)/!s/kOmegaSST/mykOmegaSST/g’ mykOmegaSST.*
```

Modifying copied files

- Then, following lines must be found in kOmegaSSTBase.C file.

```
    this->nut_ = a1_*k_/max(a1_*omega_, b1_*F2*sqrt(S2));
    this->nut_.correctBoundaryConditions();
    fv::options::New(this->mesh_).correct(this->nut_);

    BasicTurbulenceModel::correctNut();
```

- Then change above lines to following codes.

```
 scalarField Lt = be_*Cx_*Foam::pow(nb_*this->k_*this->omega_, 1.0/4.0)*
                    Foam::pow(this->mesh_.V().field(), 1.0/3.0)/
                    (Foam::pow(this->nu(), 3.0/4.0))->internalField();
    scalarField Mt = 1.0 - Foam::exp(-Lt);
    scalarField lt = be_*Foam::pow(this->k_/nb_*this->omega_*this->nu(), 3.0/4.0);
    scalarField mt = 1.0 - Foam::exp(-lt);
    // Recalculate viscosity
    this->nut_.internalField() == Foam::min(1.0, Foam::pow(Mt/mt, 4.0/3.0))*
       (this->a1_*this->k_/max(this->a1_*this->omega_, this->b1_*this->F2()*sqrt(S2)))
       ->internalField();
    this->nut_.correctBoundaryConditions();
    fv::options::New(this->mesh()).correct(this->nut_);

    BasicTurbulenceModel::correctNut();
```

Modifying copied files

- Also, add this lines to newkOmegaSSTBase.C section Constructors

```
be_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "be",
            this->coeffDict_,
            0.002
        )
    ),
    Cx_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "Cx",
            this->coeffDict_,
            0.61
        )
    ),
    nb_
    (
        dimensioned<scalar>::lookupOrAddToDict
        (
            "nb",
            this->coeffDict_,
            0.09
```

- Add this lines to newkOmegaSSTBase.C section MemberFunctions

```
        be_.readIfPresent(this->coeffDict());
        Cx_.readIfPresent(this->coeffDict());
        nb_.readIfPresent(this->coeffDict());
```

Modifying copied files

- Add this lines to newmykOmegaSST.H section default model coefficients.

```
    be              0.002;
     Cx              0.61;
     nb              0.09;
```

- Add this lines to newmykOmegaSST.H section model coefficients.

```
            dimensionedScalar be_;
            dimensionedScalar Cx_;
            dimensionedScalar nb_;
```

- Once all the files copied and modified now it is time to compile it and it can be done by typing following command in the terminal.

```
wmake libso
```

Running the VLES kOmegaSST turbulence model

- First copy the pitzDaily tutorial and create mesh then add following lines to the `controlDict`.

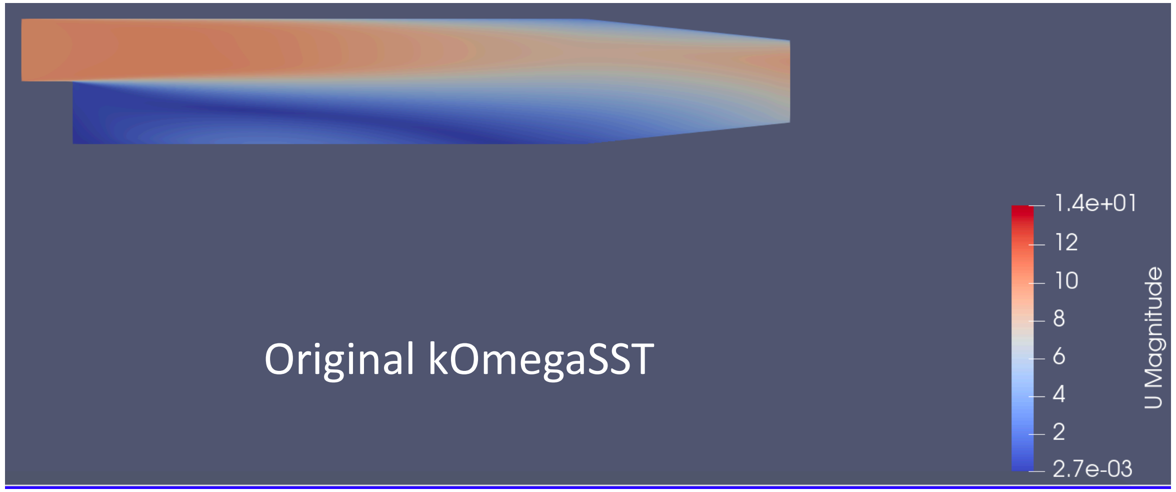
```
libs ("libmyincompressibleTurbulenceModels.so");
```

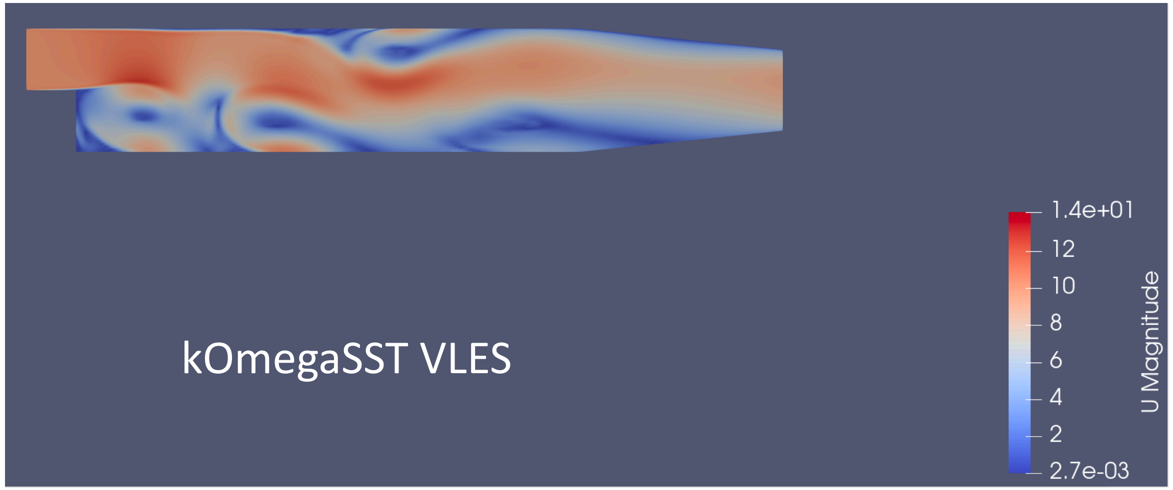
- Then, change the turbulence model to VLESkOmegaSST by opening `constant/turbulencePropeerties`.

```
RASModel mykOmegaSST;
```

- Then, change `epsilon` file to `omega`, and `fvsolution` file.
- Finally, the pitzDaily tutorial can be simulated using implemented turbulence model.

Results


1.4e+01
12
10
8
6
4
2
2.7e-03
U Magnitude
Original kOmegaSST


1.4e+01
12
10
8
6
4
2
2.7e-03
U Magnitude
kOmegaSST VLES

References

[1] Han, Xingsi & Krajnović, Siniša. (2011). A New Very Large Eddy Simulation Model for Simulation of Turbulent Flow. 10.1007/978-3-642-31818-4_11.

[2] F.R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. In *Proceedings of the fourth international symposium on turbulence, heat and mass transfer*, pages 625–632, Antalya, Turkey, 2003. Begell House.