TP Winter School TEC 21

Turbulent flow simulation using openFOAM

J. Chauchat

3 février 2023

1 Objectives

The TP is split into two parts, the first one dedicated to the RANS Simulation of the turbulent boundary layer. The second part is dedicated to the flow around a cylinder. The objectives of the TP are :

Part 1 : Turbulent boundary layer

- to perform boundary layer simulations with turbulence-averaged models $k-\omega$ and $k-\varepsilon$;
- understand the constraints on the grid resolution near the wall ;

For this part of the project, it is assumed that you have a a good knowledge of the analytical solution of the log law-of-the-wall. If you need a reminder please read appendix A

Part 2 : Flow around a cylinder using RANS & LES

- to perform numerical simulations of the flow around a 2D cylinder using the $k-\omega$ SST and $k-\epsilon$ models ;
- evaluate the sensitivity of the solution to the numerical parameters : RANS Vs URANS, numerical schemes, grid, time step ;
- evaluate the influence of turbulence modeling on the vortex-shedding frequency : the Strouhal number, using RANS and LES.

2 Turbulent boundary layer

2.1 How To start

The first objective is to reproduce Fuhrman et al. (2010) smooth wall measurements shown in figure 3. In this figure, the viscous and log layers solutions are also plotted for reference (see appendix A). The typical bulk Reynolds number of this flow is $Re_b = \bar{U}H/\nu = 1.9 \times 10^4$ for a open-channel flow of water ($\nu = 9.7 \times 10^{-7}\ m^2/s$) with water depth $H = 0.06\ m$. And the turbulent Reynolds number is $Re_* = u_* H/\nu = 10^3$ corresponding to a bed friction velocity of $u_* = 0.016\ m/s$. Both Reynolds numbers values are well-above transitional values meaning that the flow can be considered as highly turbulent.

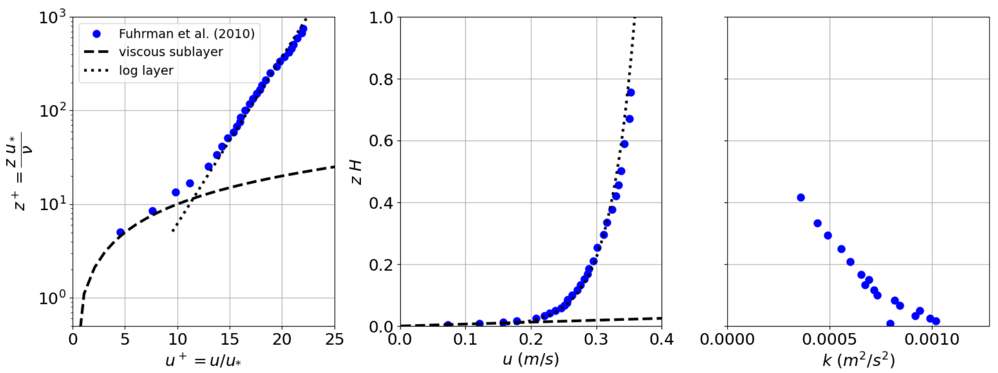

Figure 1 – Velocity profile in wall-units for the Fuhrman et al. (2010) smooth boundary configuration.

OpenFOAM is installed on the PC at LEGI, please log on the computer using your account. The subject and the LES data are available on the DAP server from LEGI at the following address :
`http://servdap.legi.grenoble-inp.fr/opendap/meige/16OASFOAM/`

If it does not work, use the following backup :

```
cp /fsnet/project/meige/app-dap-public/16OASFOAM/TP_WinterSchoolTEC21.pdf ./
```

Follow the steps below :

1. Open a terminal and type :

   ```
   module load openfoam/ python/ paraview/
   pip install fluidfoam
   mkdir -p $FOAM_RUN
   cd OpenFOAM/loginname-v2212plus/run/
   ```

2. Download the tar.gz archive 'BoundaryLayer.tar.gz'.

   ```
   wget http://servdap.legi.grenoble-inp.fr/opendap/meige/16OASFOAM/BoundaryLayer.tar.gz
   ```

   or

   ```
   cp /fsnet/project/meige/app-dap-public/16OASFOAM/BoundaryLayer.tar.gz ./
   ```

3. Unpack the archive :

   ```
   tar xvf BoundaryLayer.tar.gz
   ```

4. Change to the directory :

   ```
   cd BoundaryLayer/Fuhrman/Smooth/KOmega
   ```

5. Run the first simulation :

   ```
   ./Allrun
   ```

6. Run the post-processing script 'plot_FuhrmanEtAl2010_smooth.py' :

   ```
   cd ../../../Py/
   python plot_FuhrmanEtAl2010_smooth.py
   ```

2.2 Turbulence modeling of boundary flows using the $k-\omega$ model

In the $k-\omega$ turbulence model, the turbulent eddy viscosity ν_t is calculated as :

$$\nu_t = \frac{k}{\omega} \qquad (1)$$

The fluid turbulent kinetic energy equation reads as :

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \underbrace{R_{ij}\frac{\partial u_i}{\partial x_j}}_{=G} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]$$

where $R_{ij} = \nu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ represents the Reynolds stress tensor. The equation for the fluid specific rate of turbulent energy dissipation ω reads :

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = C_{1\omega}\frac{\omega}{k}R_{ij}\frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] - C_{2\omega}\omega^2$$

The different coefficient values can be found in table 1.

For a smooth surface, the classical boundary conditions consists in imposing the analytical solution of the ω profile in the viscous sublayer :

$$\omega_{vis} = \frac{6\nu}{\beta_1 \Delta z_{wall}^2}$$

This solution is accurate as long as the first grid point is located in the viscous sublayer $z^+ < z^+_{lam} \approx 12$.

Table 1 – $k-\omega$ model coefficients.

C_μ	$C_{1\omega}$	$C_{2\omega}$	$C_{3\omega}$	$C_{4\omega}$	σ_k	σ_ω	S_c
0.09	5/9	3/40	0.35	0 or 1	2.0	2.0	1.0

If this condition is not fulfilled, openFOAM propose to blend the viscous sublayer solution with the log layer solution [1] :

$$\omega_{log} = \frac{\sqrt{k}}{C_\mu \kappa \Delta z_{wall}}$$

In this situation, the production term in the k and ω equations may be replaced by its analytical expression :

$$G = (\nu + \nu_t)|\vec{n} \cdot (\nabla u)_f| C_\mu^{1/4} \frac{\sqrt{k}}{\kappa \Delta z_{wall}}$$

These boundary conditions are implemented in openFOAM as the 'omegaWallFunction'.

2.3 Sensitivity analysis to the grid size for $k-\omega$ model

Copy the case folder 'KOmega' with a different name, KOmega1 for example :

```
cp -r KOmega ./KOmega1
cd KOmega1
```

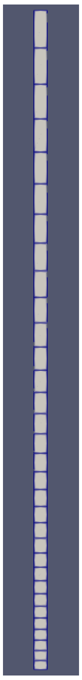

Figure 2 – Mesh of the KOmega simulation.

Change the grid parameters to evaluate the sensitivity of the numerical solution to the grid size. As explained in section 2.2, the key parameter is $\Delta z^+_{wall} = \Delta z_{wall} u_* / \nu$. The grid size is imposed in the file 'system/blockMeshDict' at line :

```
hex (3 0 4 7 2 1 5 6) (1 1 32) simpleGrading (1 1 4)
```

1. https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-wall-turbulence-omegaWallFunction.html

where the number 32 represents the total number of cells in the vertical direction and the value 4 in 'simpleGrading (1 1 4)' represents the grading factor defined as the ratio between the smallest and the largest cell size.

Open the python script 'plot_FuhrmanEtAl2010_smooth.py' and modify the following lines :

```
caseList = ['KOmega']
labelList = ['k-omega / smooth BC']
```

as follows :

```
caseList = ['KOmega','KOmega1']
labelList = [r'$dz_1^+$=8',r'$dz_1^+$=1']
```

and rerun the script :

```
python plot_FuhrmanEtAl2010_smooth.py
```

What can you conclude about the first cell size requirement for smooth boundary layer simulation using the standard $k-\omega$ model ?

Tips : You can use the python function meshdesign.getgz from fluidfoam [2] to compute the grading value, denoted as *gz*, for a given domain size *h*, first cell size *dz1* (here taken as equal to $dz1^+ = 1$ i.e. $dz1 = \nu/u_*$) and number of cells *N*. In a terminal, open a ipython shell :

```
ipython
import fluidfoam
dz1 = 9.7e-7/0.016
z,dz,gz = fluidfoam.meshdesign.getgz(0.06, dz1, 32)
```

2.4 Sensitivity analysis to the grid size for $k-\varepsilon$ model

An openFOAM case is provided in the folder 'BoundaryLayer/Fuhrman/Smooth/KEpsilon'. The $k-\varepsilon$ model is described in appendix B.

For a smooth surface, the classical boundary conditions consists in imposing the analytical solution of the ϵ profile in the viscous sublayer :

$$\epsilon_{vis} = \frac{2k\ \nu}{\Delta z_{wall}^2}$$

This solution is accurate as long as the first grid point is located in the viscous sublayer $z^+ < z^+_{lam} \approx 12$.

If this condition is not fulfilled, openFOAM propose to blend the viscous sublayer solution with the log layer solution [3] :

$$\epsilon_{log} = \frac{C_\mu k^{3/2}}{\nu_{twall} \Delta z_{wall}}$$

In this situation, the production term in the k and ϵ equations may be replaced by its analytical expression :

$$G = (\nu + \nu_t)|\vec{n} \cdot (\nabla u)_f| C_\mu^{1/4} \frac{\sqrt{k}}{\kappa \Delta z_{wall}}$$

These boundary conditions are implemented in openFOAM as the 'epsilonWallFunction'.

As for the $k-\omega$ model, perform a sensitivity analysis to the grid size. What do you conclude ?

3 Flow around a cylinder using RANS & LES

3.1 How To start

The first objective is to reproduce the flow configuration that you investigated experimentally in the wind tunnel. The typical bulk Reynolds number of this flow is $Re_D = U_0 D/\nu = 2.7 \times 10^4$, the fluid is air ($\nu = 1.45 \times 10^{-5}\ m^2/s$), the cylinder has a diameter of $D = 0.04\ m$ and the bulk velocity of the incoming flow is

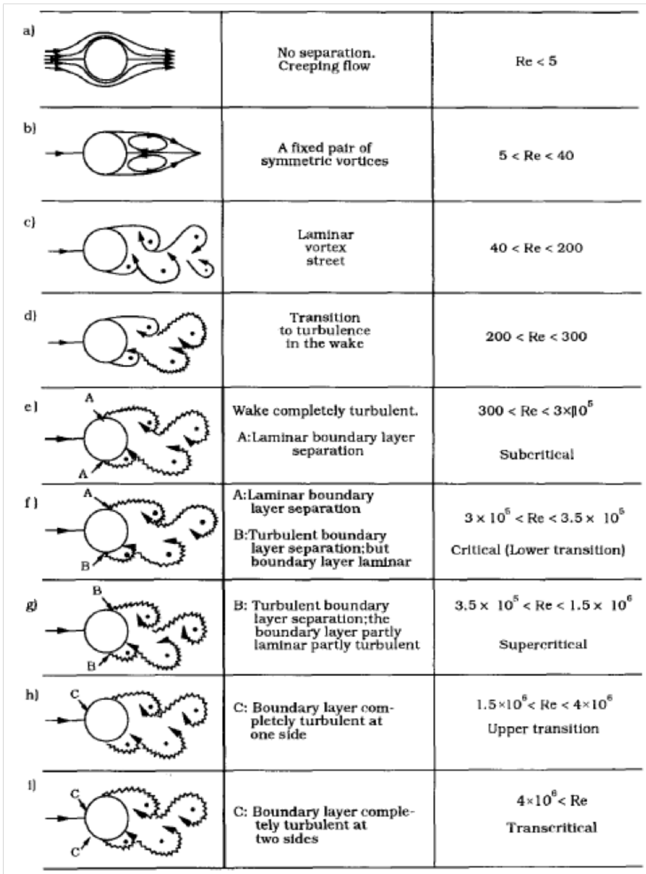

Figure 1.1 Regimes of flow around a smooth, circular cylinder in steady current.

FIGURE 3 – Region of flow around a smooth, circular cylinder in a steady current ?.

$U_0 = 10$ m/s. At this Reynolds number, the flow in the wake is expected to be fully turbulent and the boundary layer separation is laminar (see figure 3 in appendix).

From the experiments in the wind tunnel, it is possible to perform the FFT of the velocity signal (see figure 5) which exhibits a clear peak at 51 Hz. This frequency is the vortex shedding frequency which is classically made dimensionless to give the so-called Strouhal number :

$$St = \frac{f_0 D}{U_0}$$

The classical value for the Strouhal number in this configuration is $St = 0.2$ as shown in the figure. The goal of this TP is to reproduce this flow with an incompressible flow solver, here pimpleFOAM, to evaluate the capabilities of different turbulence modeling approaches to reproduce the vortex shedding downstream a cylinder.

The RANS configurations are available from the git repository :

```
https://gricad-gitlab.univ-grenoble-alpes.fr/chauchaj/tp_num_tec21.git
```

You may either download an archive directly from gitlab or type the following command in a terminal :

```
git clone https://gricad-gitlab.univ-grenoble-alpes.fr/chauchaj/tp_num_tec21.git
```

2. https ://fluidfoam.readthedocs.io/en/latest/generated/fluidfoam.meshdesign.html
3. `https://www.openfoam.com/documentation/guides/latest/doc/guide-bcs-wall-turbulence-epsilonWallFunction.html`

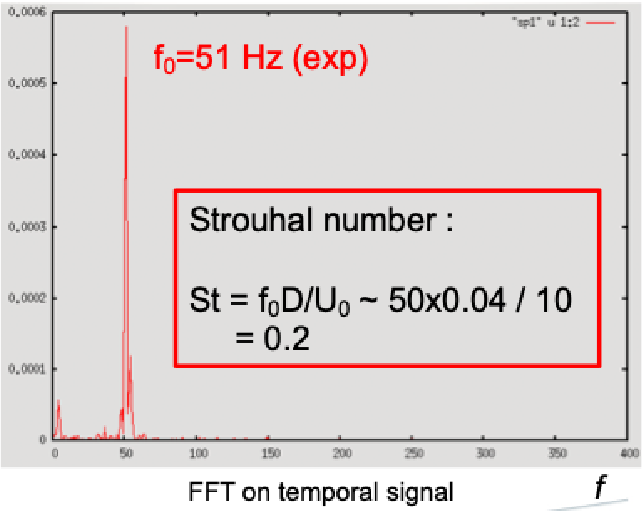

FIGURE 4 – FFT of the velocity signal taken from the experiments in the LEGI wind tunnel.

3.2 Numerical simulation of the flow around a cylinder : $k-\omega$ SST model

Perform simulations with the $k-\epsilon$ and the $k-\omega$ SST model using the RANS approach :

```
cd tp_num_tec21/RANS/KWSST/
./Allrun
simpleFoam >log&
paraview contourU.pvsm
```

You should see the following figure :

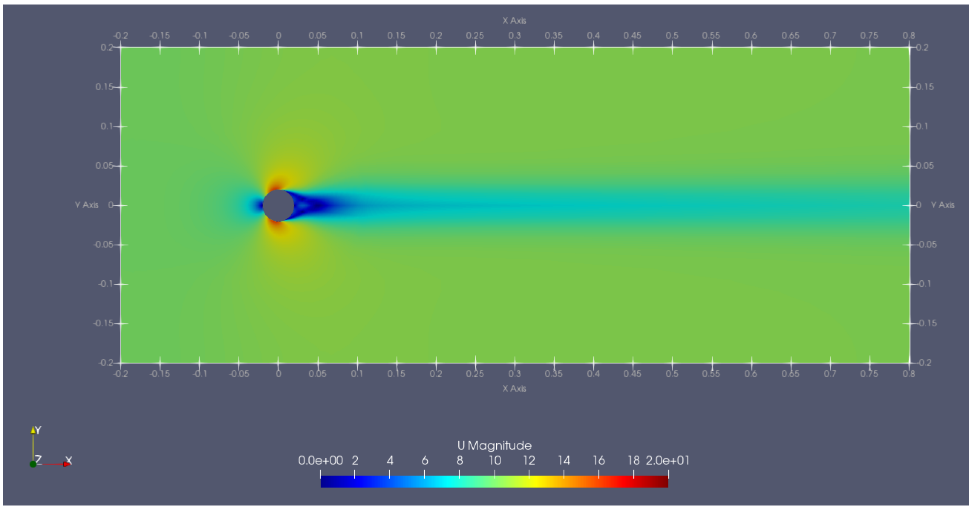

FIGURE 5 – Isocontour of velocity magnitude obtained from the steady state RANS $k-\omega$ SST model.

Open the script and adjust time in compare_velocity_profiles.py to correspond to your latest time

```
cd ../../PY
python compare_velocity_profiles.py
```

The convergence of the solver can be visualized using the commands :

```
foamLog log
gnuplot plotLogs.plt
```

The plot shows the evolution of the residual of the pressure equation as a function of the iteration count. When it reaches the convergence criteria, here set to 10^{-5}, the simulation stops.

You can characterize the recirculation length using the velocity field, it corresponds to the position at which the streamwise velocity becomes positive downstream the cylinder.

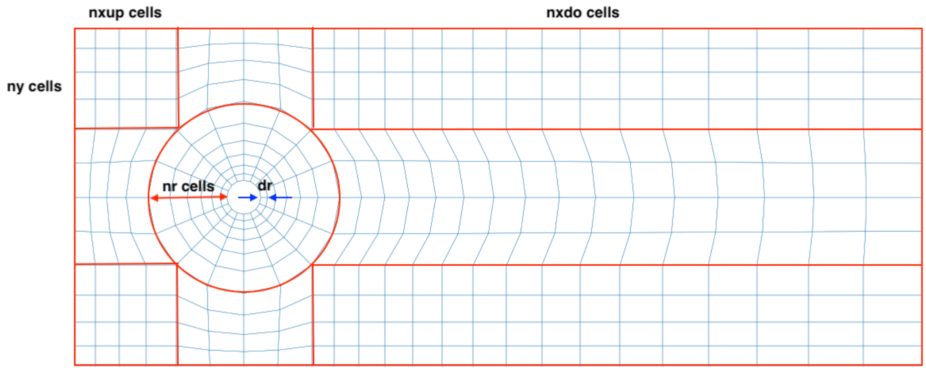

FIGURE 6 – Mesh of the cylinder configuration.

This first simulation is using a first order scheme (upwind). Once you have done this simulation, you can change the divergence schemes from 'upwind' to 'linearUpwind' (second order) in the file 'system/fvSchemes' :

```
divSchemes
{
    default         none;

    div(phi,U)      Gauss linearUpwind grad(U);
//    div(phi,U)      Gauss upwind phi;

    turbulence      Gauss linearUpwind;
//    turbulence      Gauss upwind phi;
...
```

and rerun the simulation

```
simpleFoam >log &
```

by default, the code will rerun from the latest time step until it reaches the convergence criteria in terms of residual (see file system/fvSolution). You can look at the residual to understand what the solver does.

URANS simulation using the $k-\omega$ SST with first order schemes

The first order schemes are Euler in time and upwind for advection (div scheme in openFOAM)

```
cd ../../URANS/ORDER1/KWSST
./Allrun
pimpleFoam > log &
```

You can visualize the velocity profiles using the script 'compare_velocity_profiles.py' by modifying the configName parameter to 'RANS_URANS'.

```
cd ../../../PY
python compare_velocity_profiles.py
```

URANS simulation using the $k-\omega$ SST with second order schemes

The second order schemes are Crank-Nicolson in time and linearUpwind for advection (2nd order)

```
cd ../../ORDER2/KWSST
./Allrun
pimpleFoam > log &
```

You can visualize the velocity profiles using the script 'compare_velocity_profiles.py' by modifying the configName parameter to 'URANS_O1O2'.

Then you can plot the time series of the drag and lift coefficient as well as the time series of the velocity probes and the associated Fourrier transform by using the script :

```
cd ../../../PY
python compare_Strouhal.py
```

change the configName to see the order 1 and then the order 2 solution. Adjust the parameter 'timestart' to focus the analysis on the statistically steady state time window.

Bonus : Perform a sensitivity analysis to the grid size and the time step for the $k-\omega$ SST model.

The mesh is generated using the python script

```
python generateMesh.py
```

you can increase the mesh resolution by increasing the following parameters : nr, $nxup$, $nxdo$ and ny.
The time step can be reduced in the file 'system/contolDict', the parameter is 'deltaT'.

3.3 Numerical simulation of the flow around a cylinder : LES

The LES is too expensive to be ran during the TP. The subgrid scale model used for this simulation is the dynamic Lagrangian subgrid model and the mesh is made of 2,869,248 cells and the simulation has run over $t = 0.63742$ s corresponding to $t/T_b = 160$ bulk time with $T_b = D/U_0 = 0.004$ s.

You can visualize the data by making a link using the command :

```
./createLinks
paraview isoQ.pvsm
```

Then modify the script 'compare_Strouhal.py' to plot the Fourrier transform of the lift and velocity probes and estimate the Strouhal number.

References

Fuhrman, D. R. , Dixen M. and Jacobsen,N. G. (2010) Physically consistent wall boundary conditions for the $k-\omega$ turbulence model, Journal of Hydraulic Research, 48 :6, 793-800, DOI : 10.1080/00221686.2010.531100.

Wilcox, D. C. (2008). Formulation of the k-ω turbulence model revisited. AIAA Journal, 46(11) :2823-2838.

A Analytical description of the turbulent boundary layer

The turbulent boundary layer over a smooth wall can be modeled using Prandtl mixing length approach using the following assumptions :

— the shear stress profile τ is almost constant in the near-wall region and equal to $\tau_w = \rho^f u_*^2$

— the most effective eddy length scale is a linear function of the distance to the wall : $l_m = \kappa z$

Using the eddy viscosity concept one can write the stress-velocity shear rate relationship as :

$$\rho^f l_m^2 \frac{\mathrm{d}U_l}{\mathrm{d}z} = \rho^f u_*^2$$

$$\Leftrightarrow \kappa^2 z^2 \left|\frac{\mathrm{d}U_l}{\mathrm{d}z}\right| \frac{\mathrm{d}U_l}{\mathrm{d}z} = u_*^2$$

$$\Leftrightarrow \frac{\mathrm{d}U_l}{\mathrm{d}z} = \frac{u_*}{\kappa z}$$

Upon integration from δ_v to z, this equation gives the so-called log-law of the wall :

$$\int_{\delta_v}^{z} \frac{\mathrm{d}U_l}{\mathrm{d}z} \mathrm{d}z = \int_{\delta_v}^{z} \frac{u_*}{\kappa z} \mathrm{d}z$$

$$\Leftrightarrow U_l(z) - U_l(\delta_v) = \frac{u_*}{\kappa} \ln\left(\frac{z}{\delta_v}\right)$$

where δ_v represents the viscous sublayer thickness equal to $\delta_v \approx 12\nu/u_*$.

In the viscous sublayer, turbulence is negligible and the stress-velocity shear rate relationship reads :

$$\rho^f \nu \frac{\mathrm{d}U_v}{\mathrm{d}z} = \rho^f u_*^2$$

$$\Leftrightarrow \frac{\mathrm{d}U_v}{\mathrm{d}z} = \frac{u_*^2}{\nu}$$

Upon integration from 0 to z and assuming a no-slip boundary condition at the wall, this equation gives the viscous sublayer profile :

$$\int_{0}^{z} \frac{\mathrm{d}U_v}{\mathrm{d}z} \mathrm{d}z = \int_{0}^{z} \frac{u_*^2}{\nu} \mathrm{d}z$$

$$\Leftrightarrow U_v(z) = \frac{u_*^2}{\nu} z$$

This solution gives the integration constant for the logarithmic layer solution :

$$U_l(z) = U_v(\delta_v) + \frac{u_*}{\kappa} \ln\left(\frac{z}{\delta_v}\right)$$

$$\Leftrightarrow U_l(z) = \frac{u_*^2}{\nu} 12 \frac{\nu}{u_*} + \frac{u_*}{\kappa} \ln\left(\frac{z\ u_*}{12\ \nu}\right)$$

$$\Leftrightarrow U_l(z) = \frac{u_*}{\kappa} \ln\left(\frac{z\ u_*}{12\ \nu}\right) + 12\ u_*$$

This solution is often presented in dimensionless form by introducing the wall unit : $z^+ = z\ u_*/\nu$ and by scaling the velocity by u_* : $u^+ = U/u_*$. The solution can then be written as :

$$u^+ = \begin{cases} z^+ \quad \text{for} \quad z^+ < 12 \\ \dfrac{1}{\kappa} \ln(z^+) + C^+, \quad \text{for} \quad z^+ \geq 12 \end{cases} \tag{2}$$

where $C^+ = 12 - \ln(12)/\kappa$. Experimental measurements suggests that $\kappa \approx 0.41$ and $C^+ \approx 5.0$ for smooth walls even if the exact values are still under debate in the scientific community. The sharp transition between the viscous and the log layers is also a strong assumption. In reality, this transition is smooth and occurs in the so-called buffer layer in the range $5 < z^+ < 30$.

B $k - \varepsilon$ model

In the $k - \varepsilon$ model the turbulent eddy viscosity ν_t^f is calculated as :

$$\nu_t^f = C_\mu \frac{k^2}{\varepsilon}, \tag{3}$$

where C_μ is an empirical coefficient (see Table 2). The Turbulent Kinetic Energy (TKE) k is computed from the solution of equation (4) :

$$\frac{\partial k}{\partial t} + u_j^f \frac{\partial k}{\partial x_j} = \frac{R_{ij}^f}{\rho^f} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\left(\nu^f + \frac{\nu_t^f}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] - \varepsilon \tag{4}$$

The balance equation for the rate of turbulent kinetic energy dissipation ε is written as :

$$\frac{\partial \varepsilon}{\partial t} + u_j^f \frac{\partial \varepsilon}{\partial x_j} = C_{1\varepsilon} \frac{\varepsilon}{k} \frac{R_{ij}^f}{\rho^f} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\left(\nu^f + \frac{\nu_t^f}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] - C_{2\varepsilon}\frac{\varepsilon^2}{k}$$

Table 2 summarizes the model coefficients.

Table 2 – $k - \varepsilon$ model coefficients.

C_μ	$C_{1\varepsilon}$	$C_{2\varepsilon}$	$C_{3\varepsilon}$	$C_{4\varepsilon}$	σ_k	σ_ε	S_c
0.09	1.44	1.92	1.2	0 or 1	1.0	1.3	1

C Comments on RANS models

C.1 Standard $k - \epsilon$ model

The baseline two-transport-equation model solving for kinetic energy k and turbulent dissipation ϵ. Turbulent dissipation is the rate at which velocity fluctuations dissipate. This is the default $k - \epsilon$ model. Coefficients are empirically derived ; valid for fully turbulent flows only. In the standard k-e model, the eddy viscosity is determined from a single turbulence length scale, so the calculated turbulent diffusion is that which occurs only at the specified scale, whereas in reality all scales of motion will contribute to the turbulent diffusion. The k-e model uses the gradient diffusion hypothesis to relate the Reynolds stresses to the mean velocity gradients and the turbulent viscosity. Performs poorly for complex flows involving severe pressure gradient, separation, strong streamline curvature. The most disturbing weakness is lack of sensitivity to adverse pressure gradients ; another shortcoming is numerical stiffness when equations are integrated through the viscous sublayer which are treated with damping functions that have stability issues [F. R. Menter, "Zonal Two Equation $k-\omega$ Turbulence Models for Aerodynamic Flows," AIAA Paper #93-2906, 24th Fluid Dynamics Conference, July 1993 ; F. R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal, vol. 32, no. 8, pp. 1598-1605, 1994]. Notes : The author's self-investigation for flow through a pipe is consistent with the statements that this model is valid for flows without separation and for fully turbulent flow. Compared to a finned problem which had separation and which predicted erroneous results with the k-e model, this pipe flow did not have separation and results of k-e and k-w models showed good agreement for high Reynolds numbers. In this pipe flow, as Reynolds number was decreased, the difference between the inlet pressures predicted by the k-e and k-w models increased. Note that, based on the author's limited experience, results for temperature are less sensitive to model choice and for velocity seem indifferent. Pressure results seem highly sensitive to both the model choice and the mesh. Be careful to check all results before deciding that results are valid. For additional details, see section entitled "Comparison of k-e and k-w Models."

Pros : Robust. Widely used despite the known limitations of the model. Easy to implement. Computationally cheap. Valid for fully turbulent flows only. Suitable for initial iterations, initial screening of alternative designs, and parametric studies.

Cons : Performs poorly for complex flows involving severe pressure gradient, separation, strong streamline curvature. Most disturbing weakness is lack of sensitivity to adverse pressure gradients ; another shortcoming is numerical stiffness when equations are integrated through the viscous sublayer which are treated with damping functions that have stability issues [F. R. Menter, "Zonal Two Equation $k-\omega$ Turbulence Models for Aerodynamic Flows," AIAA Paper #93-2906, 24th Fluid Dynamics Conference, July 1993 ; F. R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal, vol. 32, no. 8, pp. 1598-1605, 1994].

C.2 Standard $k-\omega$ model

A two-transport-equation model solving for kinetic energy k and turbulent frequency ω. This is the default $k-\omega$ model. This model allows for a more accurate near wall treatment with an automatic switch from a wall function to a low-Reynolds number formulation based on grid spacing. Demonstrates superior performance for wall-bounded and low Reynolds number flows. Shows potential for predicting transition. Options account for transitional, free shear, and compressible flows. The k-e model uses the gradient diffusion hypothesis to relate the Reynolds stresses to the mean velocity gradients and the turbulent viscosity. Solves one equation for turbulent kinetic energy k and a second equation for the specific turbulent dissipation rate (or turbulent frequency) w. This model performs significantly better under adverse pressure gradient conditions. The model does not employ damping functions and has straightforward Dirichlet boundary conditions, which leads to significant advantages in numerical stability. This model underpredicts the amount of separation for severe adverse pressure gradient flows.

Pros : Superior performance for wall-bounded boundary layer, free shear, and low Reynolds number flows. Suitable for complex boundary layer flows under adverse pressure gradient and separation (external aerodynamics and turbomachinery). Can be used for transitional flows (though tends to predict early transition).

Cons : Separation is typically predicted to be excessive and early. Requires mesh resolution near the wall.

C.3 $k-\omega$ SST model

Shear Stress Transport (SST) is a variant of the standard $k-\omega$ model. Combines the original Wilcox $k-\omega$ model for use near walls and the standard $k-\epsilon$ model away from walls using a blending function, and the eddy viscosity formulation is modified to account for the transport effects of the principle turbulent shear stress [F. R. Menter, "Zonal Two Equation k-w Turbulence Models for Aerodynamic Flows," AIAA Paper #93-2906, 24th Fluid Dynamics Conference, July 1993 ; F. R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," AIAA Journal, vol. 32, no. 8, pp. 1598-1605, 1994]. Also limits turbulent viscosity to guarantee that $\tau_T \approx k$. The transition and shearing options are borrowed from standard $k-\omega$. No option to include compressibility.

Pros : Offers similar benefits as standard $k-\omega$. The SST model accounts for the transport of turbulent shear stress and gives highly accurate predictions of the onset and the amount of flow separation under adverse pressure gradients. SST is recommended for high accuracy boundary layer simulations.

Cons : Dependency on wall distance makes this less suitable for free shear flows compared to standard $k-\omega$. Requires mesh resolution near the wall. A Reynolds Stress model may be more appropriate for flows with sudden changes in strain rate or rotating flows while the SST model may be more appropriate for separated flows.