Simulation of Turbulent Flows

- From the Navier-Stokes to the RANS equations
- Turbulence modeling
- k-ε model(s)
- Near-wall turbulence modeling
- Examples and guidelines

Navier-Stokes equations

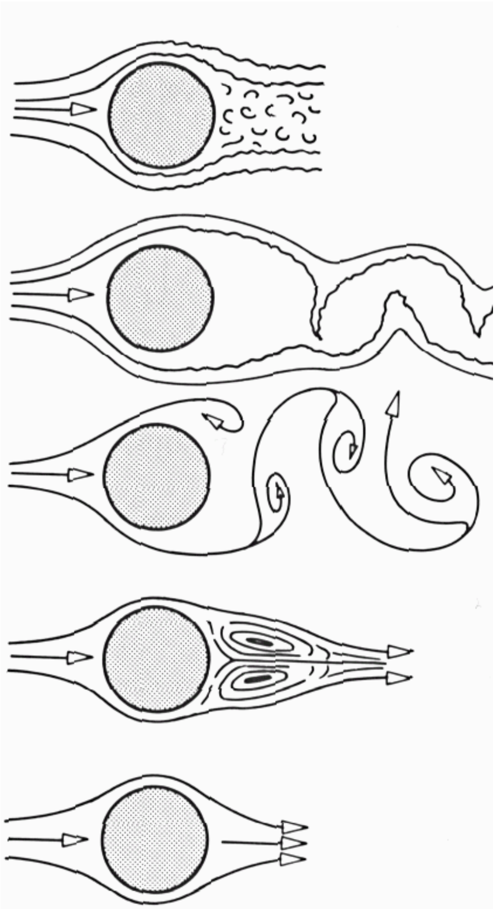
The Navier-Stokes equations (for an incompressible fluid) in an adimensional form contain one parameter: the Reynolds number:

$$\mathrm{Re} = \rho\, \mathrm{V}_{\mathrm{ref}} \mathrm{L}_{\mathrm{ref}} \,/\, \mu$$

it measures the relative importance of convection and diffusion mechanisms

What happens when we increase the Reynolds number?

Reynolds Number Effect

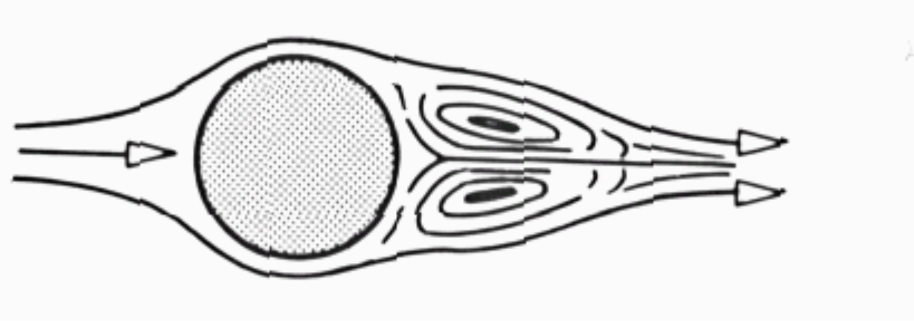


350K < Re	Turbulent Separation Chaotic
200 < Re < 350K	Laminar Separation/Turbulent Wake Periodic
40 < Re < 200	Laminar Separated Periodic
5 < Re < 40	Laminar Separated Steady
Re < 5	Laminar Attached Steady

Re

Experimental Observations

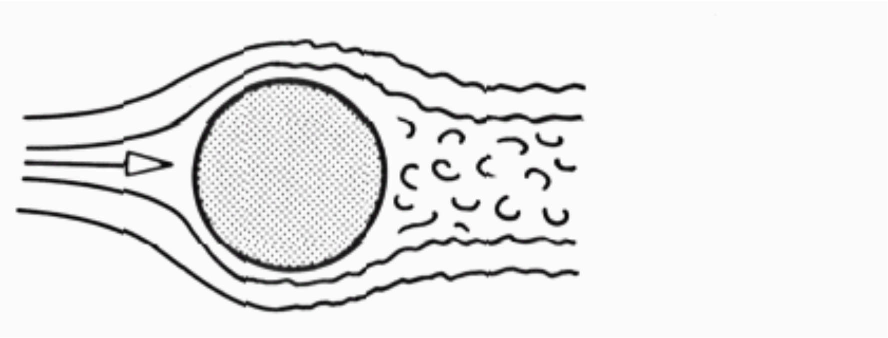
Laminar vs. Turbulent Flow



Laminar Flow

The flow is dominated by the object shape and dimension (large scale)

Easy to compute



Turbulent Flow

The flow is dominated by the object shape and dimension (large scale) and by the motion and evolution of small eddies (small scales)

Challenging to compute

Why turbulent flows are challenging?

Unsteady aperiodic motion

Fluid properties exhibit random spatial variations (3D)

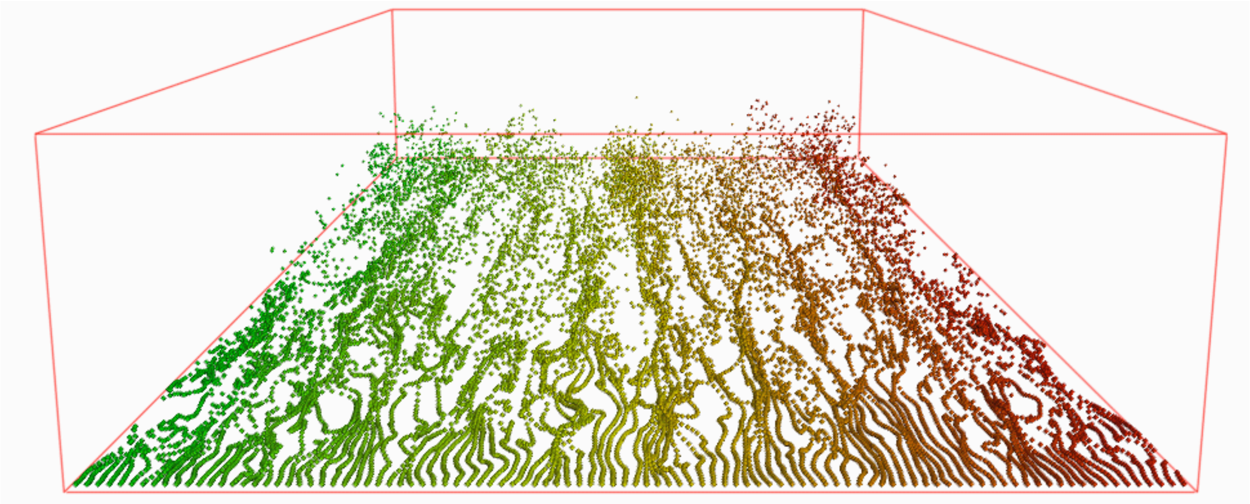
Strong dependence from initial conditions

Contain a wide range of scales (eddies)

The implication is that the turbulent simulation MUST be always three-dimensional, time accurate with extremely fine grids

Direct Numerical Simulation

The objective is to solve the time-dependent NS equations resolving ALL the scale (eddies) for a sufficient time interval so that the fluid properties reach a statistical equilibrium



Grid requirement: N ~ $(Re_\tau)^{9/4}$ ~ $1x10^7$ for $Re_\tau = 800$

Time step requirement: Δt ~ $(Re_\tau)^{-1/2}$ ~ $1x10^{-5}$ for $Re_\tau = 800$

$y^+ = \rho\ y_p\ u_\tau\ /\mu$ $\quad$ $u_\tau = (\tau_w/\rho)^{1/2}$

Beyond DNS

DNS is possible but only for low Reynolds number flows (and simple geometries)

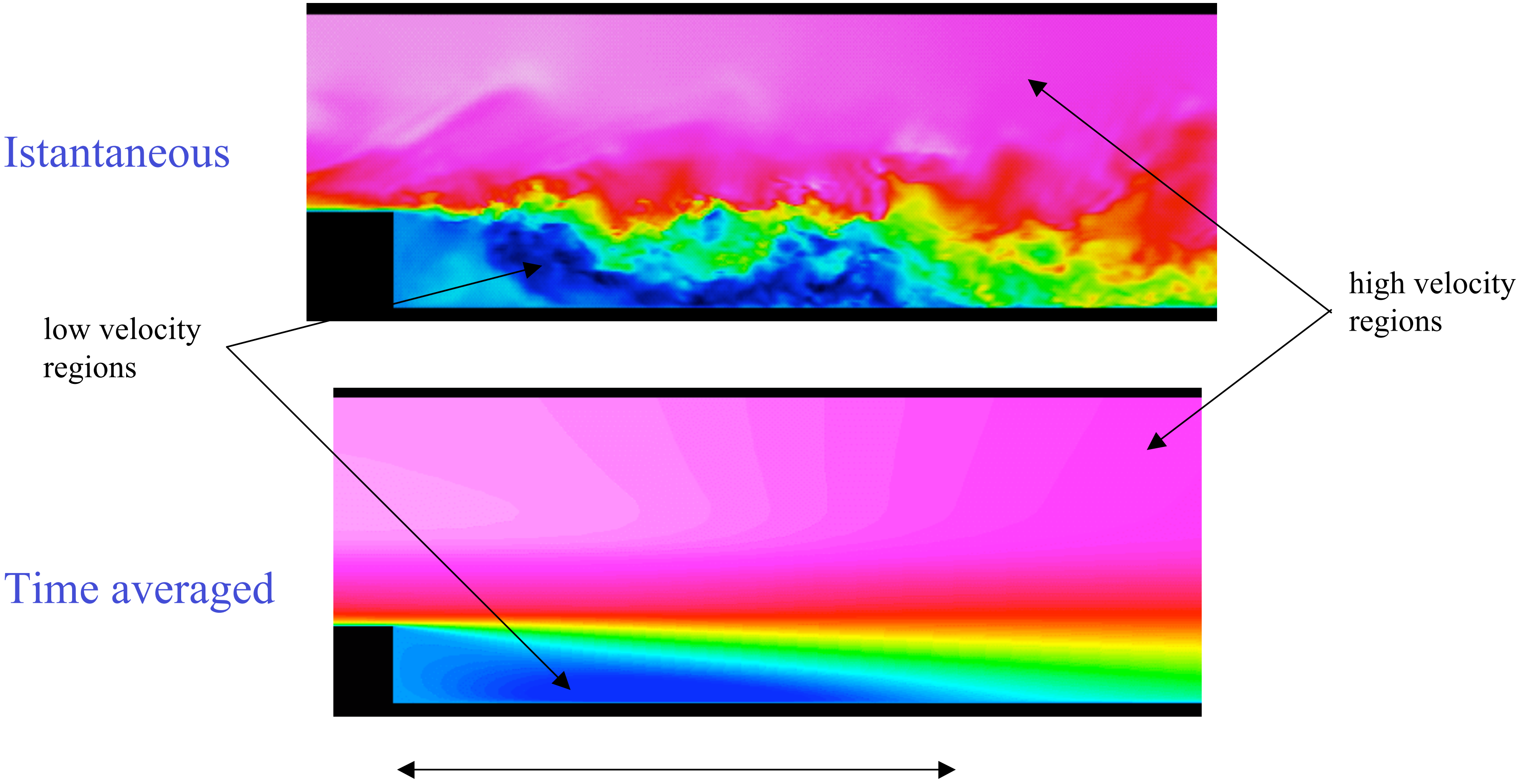
The (time and space) details provided by DNS are NOT required for design purposes

Time averaged quantities are appropriate for engineering purposes

Large scale resolution (not to the level of the smallest eddies) is enough for applications

Can we extract time-average and large-scale quantities at a reasonable computational cost?

Flow over a Backstep



Istantaneous

Time averaged

low velocity regions

high velocity regions

Length of the recirculation region is of engineering interest

Reynolds-Averaged Navier-Stokes Equations

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\frac{\mu}{\rho}\frac{\partial u_i}{\partial x_j}\right) \qquad \frac{\partial u_i}{\partial x_i} = 0$$

Define Reynolds-averaged quantities

$$u_i(x_k, t) = U_i(x_k) + u'(x_k, t)$$

$$U_i(x_k) = \lim_{T\to\infty} \frac{1}{T}\int_0^T u(x_k, t)dt$$

Substitute and average:

$$\cancel{\frac{\partial U_i}{\partial t}} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\frac{\mu}{\rho}\frac{\partial U_i}{\partial x_j}\right) + \boxed{\frac{\partial\left(-\overline{u_i' u_j'}\right)}{\partial x_j}} \qquad \frac{\partial U_i}{\partial x_i} = 0$$

$$R_ij = -\overline{u_i' u_j'}$$

Closure problem

Turbulence modeling

Define the Reynolds stresses in terms on known (averaged) quantities

1) Boussinesq hypothesis
   - simple relationship between Reynolds stresses and velocity gradients through the eddy viscosity (similar to molecular viscosity)
   - isotropic (eddy viscosity is a scalar!)

2) Reynolds stress transport models
   - equations derived directly manipulating the NS equations
   - still contain unknown (undetermined) quantities
   - no assumption of isotropy
   - very complicated and expensive to solve

3) Non-linear Eddy viscosity models (Algebraic Reynolds stress)

4) Model directly the divergence of the Reynolds Stresses

Eddy viscosity models

Boussinesq relationship: $R_i j = -\overline{u_i' u_j'} = 2\frac{\boxed{\mu_t}}{\rho} S_{ij}$ with: $S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\frac{(\mu + \mu_t)}{\rho}\frac{\partial U_i}{\partial x_j}\right]$$

Guidelines for defining the eddy viscosity:

1) Dimensional arguments
   - units are [m²/s]
   - define 2 out of three scales: velocity, length, time
2) Physical arguments
   - asymptotic analysis
   - consistency with experimental findings
3) Numerical arguments
   - simple and easy to compute

Classification of eddy viscosity models

The various models (about 200) are classified in terms of number of transport equations solved in addition to the RANS equations:

1) zero-equation/algebraic models:
   Mixing Length, Cebeci-Smith, Baldwin-Lomax, etc
2) one-equation models:
   Wolfstein, Baldwin-Barth, Spalart-Allmaras, k-model, etc
3) two-equation models:
   k-ε, k-ω, k-τ, k-L, etc.
4) three-equation models:
   k-ε-A
5) four-equation models:
   v^2-f model

Zero-equation model   Prandtl Mixing Length

$$\mu_t == \rho L^2_{mix} S = \rho L^2_{mix}\sqrt{2 S_{ij} S_{ij}}$$

From dimensional arguments and analogy with molecular transport

Definition of L is different for each problem (boundary layes, mixing layers, etc.)

Eddy viscosity is zero if the velocity gradients are zero

No “history” effect; purely local

L can be made “universal” using ad hoc functions of distance from the walls, pressure gradients, etc.

One-equation model k-model

$$TKE = k = \frac{1}{2}\overline{u_i' u_i'} = \frac{1}{2}\left(\overline{u'u'} + \overline{v'v'} + \overline{w'w'}\right)$$

An equation from k can be derived directly from the NS equations (using the definition)

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = R_{ij}\frac{\partial U_i}{\partial x_j} - \frac{\mu}{\rho}\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}} + \frac{\partial}{\partial x_j}\left(\frac{\mu}{\rho}\frac{\partial k}{\partial x_j} - \frac{1}{2}\overline{u_i' u_i' u_j'} - \overline{p' u_j'}\right)$$

convection production dissipation Viscous diffusion turbulent diffusion

$k^{1/2}$ is assumed to be the velocity scale

it still requires a length scale L as before to define the eddy viscosity

4 out of 7 terms in the k equation require further assumptions

Production is computed using the Boussinesq approximation

Dissipation is modeled (using dimensionality arguments) as $k^{3/2}/L$

Turbulent transport and pressure diffusion are modeled together:

$$\frac{1}{2}\overline{u_i' u_i' u_j'} + \overline{p' u_j'} \approx -\frac{\mu_t}{\rho\sigma_k}\frac{\partial k}{\partial x_j}$$

One-equation model   k-model

The final form of the model is:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = P_k - \epsilon + \frac{\partial}{\partial x_j}\left[\frac{1}{\rho}\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \qquad \epsilon \approx f(k, L_{mix})$$

The ONLY advantage with respect to zero-equation models is the inclusion of the history effects.

Modern one-equation models abandoned the k-equation and are based on a ad-hoc Transport equation for the eddy viscosity directly.

Spalart-Allmaras model:

$$\frac{\partial \tilde{\nu}}{\partial t} + U_j \frac{\partial \tilde{\nu}}{\partial x_j} = P_{\tilde{\nu}} - \epsilon_{\tilde{\nu}} + \frac{\partial}{\partial x_j}\left[\frac{1}{\rho}\left(\mu + \frac{\tilde{\nu}}{\sigma_{\tilde{\nu}}}\right)\frac{\partial \tilde{\nu}}{\partial x_j}\right]$$

Two-equation model k-ϕ family

The main drawback of the k one-equation model is the incomplete representation of the two scales required to build the eddy viscosity; two-equation models attempt to represent both scales independently.

- All models use the transport equation for the turbulent kinetic energy k
- Several transport variables are used

ε: turbulence dissipation rate
L: turbulent length scale
ω: inverse of turbulent time scale
ω^2
g
ψ
τ

Example:

$$\ell = \frac{k^{3/2}}{\varepsilon} \longrightarrow \nu_t = c_\mu k^{1/2} \ell = c_\mu \frac{k^2}{\varepsilon}.$$

k-ε model

The k equation is the same as before:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = \frac{\mu_t}{\rho} S^2 - \epsilon + \frac{\partial}{\partial x_j}\left[\frac{1}{\rho}\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]$$

The ε equation can be obtained from the NS equations but it contains several undetermined quantities; it is therefore derived "mimicking" the k equation

$$\frac{\partial \epsilon}{\partial t} + U_j \frac{\partial \epsilon}{\partial x_j} = \frac{\epsilon}{k}\left(C_{1\epsilon}\frac{\mu_t}{\rho} S^2 - C_{2\epsilon}\epsilon\right) + \frac{\partial}{\partial x_j}\left[\frac{1}{\rho}\left(\mu + \frac{\mu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right]$$

The eddy viscosity is obtained as: $\mu_t = \rho C_\mu \frac{k^2}{\epsilon}$

There are 5 *free* constants $\sigma_k, \sigma_\epsilon, C_{1\epsilon}, C_{2\epsilon}, C_\mu$

Determining the constants?

The constants can be determined studying simple flows:

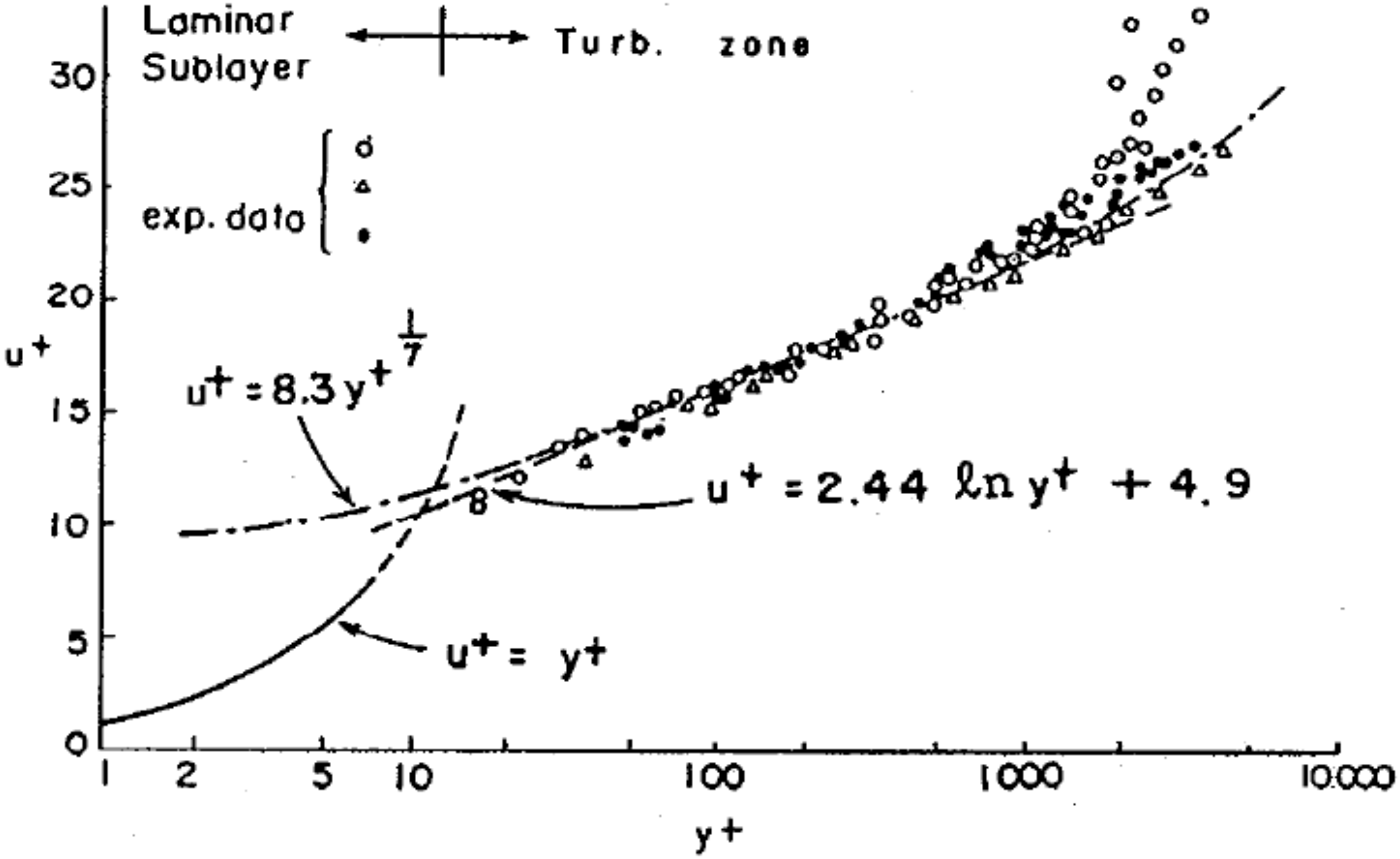
1. Decaying homogeneous isotropic turbulence $C_{2\epsilon};$
2. Homogeneous shear flow $C_{1\epsilon}$ $C_{2\epsilon};$
3. The Logarithmic Layer $C_{1\epsilon}$ $C_{2\epsilon}; C_\mu, \sigma_\epsilon$
4. …

Or by comparison with experimental data

Standard k-ε refers to a certain choice of the constants (Launder & Sharma 1972)

Structure of the Turbulent Boundary Layer

Universal Law (velocity profile)



At High Reynolds number the viscous dominated layer is so thin that it is very difficult to resolve it

Wall Function Approach (High-Re k-ε)

The laminar sublayer is NOT resolved

First grid point is assumed to be in the logarithmic layer ($y^+>11$) and the velocity is assumed to be described by:

$$u^+ = (1/\kappa)\ln(Ey^{+)}$$

A slip condition ($u \neq 0$) is imposed at the wall (imposed shear stress)

k boundary condition is usually imposed as a zero-gradient.

ε is obtained by equilibrium condition ($P_k=\varepsilon$)

If first grid point is too close (viscous layer) then the velocity is: $u^+ = y^+$

Near Wall Region Modeling

From a physical point of view:

It is important because solid walls are the main source of vorticity and turbulence (local extrema of turbulent kinetic energy, large variations of turbulence dissipation, etc.)

In engineering applications:

Wall quantities (velocity gradients, pressure, etc.) are very important in several applications

Flow separation and reattachment are strongly dependent on a correct prediction of the development of turbulence near walls

Damping Functions Approach (Low-Re k-ε)

The equations are integrated to the wall WITHOUT assuming an universal law for the velocity profile and an equilibrium conditions for *k* and ε

The problem is that the model predicts the wrong behavior for *k* and ε near the solid walls (from DNS and experimental observation)

The equations are modified using algebraic functions to “damp” certain terms:

$$\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} = \frac{\mu_t}{\rho} S^2 - \epsilon + \frac{\partial}{\partial x_j} \left[\frac{1}{\rho} \left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right]$$

$$\frac{\partial \epsilon}{\partial t} + U_j \frac{\partial \epsilon}{\partial x_j} = \frac{\epsilon}{k} \left(f_1 C_{1\epsilon} \frac{\mu_t}{\rho} S^2 - f_2 C_{2\epsilon} \epsilon \right) + \frac{\partial}{\partial x_j} \left[\frac{1}{\rho} \left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right]$$

$$\mu_t = \rho C_\mu f_\mu \frac{k^2}{\epsilon}$$

The damping functions are designed to correct the behavior of the eddy viscosity

$$\sigma_k, \sigma_\epsilon, C_{1\epsilon}, C_{2\epsilon}, C_\mu, f_\mu, f_1, f_2$$

Damping Functions

We need to specify 5 constants plus 3 functions:

$$\sigma_k, \sigma_\epsilon, C_{1\epsilon}, C_{2\epsilon}, C_\mu, f_\mu, f_1, f_2$$

MANY choices are available (about 30 different formulations!). Fluent has 6 different versions

Classic model is the Launder and Sharma model:

$$f_1 = 1, \quad f_2 = 1 - 0.3e^{-Re_T^2}, \quad f_\mu = e^{\frac{-3.4}{(1+0.02Re_T)^2}}$$

$$Re_T = \rho k^2/\mu\epsilon$$

Others might be function of the distance from the wall, the pressure gradient, etc.

Two-Layer Approach

The computational domain is divided in two regions: viscosity affected (near wall) and fully turbulent core (outer region)

Two different models are used: the complete k-ε model for the outer region and a simplified model (typically a one-equation k-based model) for the near-wall

The separation between the two regions is defined in terms of a distance from the wall (y^+~30)

The main assumption is related to the definition of ε which is based on

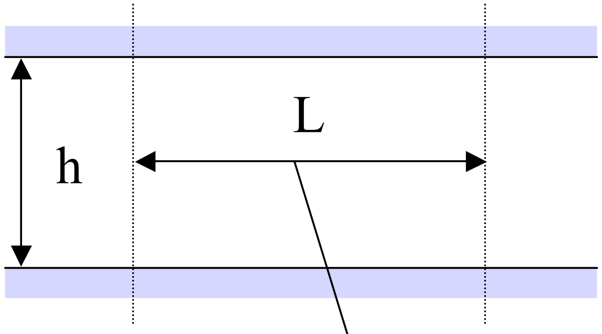
$$\epsilon \approx f(k, L_{mix})$$

Near-Wall Treatments for k-ε models

Approach	Physics	Grid requirement	Numerics	Accuracy
Wall functions	-	+	+	+/-
Two layer	+/-	+/-	+	+/-
Damping functions	+/-	-	-	+/-

Summary and Comparison

Example: Turbulent Channel Flow



Periodic boundaries

Problem set-up

Material Properties:
$\rho = 1 kg/m^3$
$\mu = 0.0017 kg/ms$

Reynolds number:
h = 2m, L=1m
$Re_\tau = \rho U_\tau h/\mu$ = **590**

Boundary Conditions:
Periodicity $\Delta p/L=1=U_\tau$
No-slip top/bottom walls

Initial Conditions:
u = 1; v = p = 0

Turbulence model:
k-ε with wall functions

Solver Set-Up

Segregated Solver

Discretization:
2nd order upwind
SIMPLE

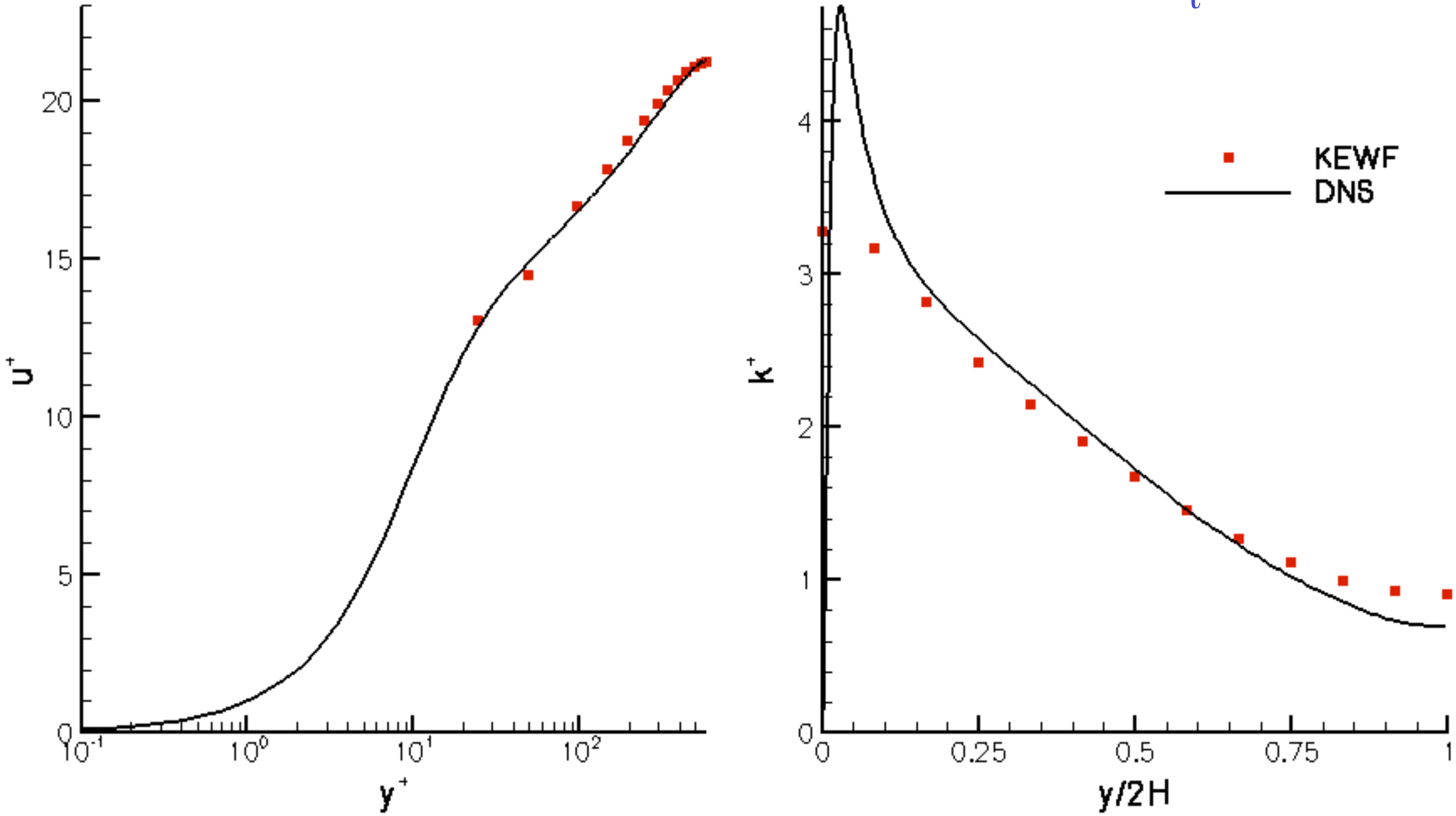
Multigrid
V-Cycle

Grid

SAME GRID USED FOR THE LAMINAR FLOW @ Re=20

Turbulent Channel Flow

$Re_\tau = 590$

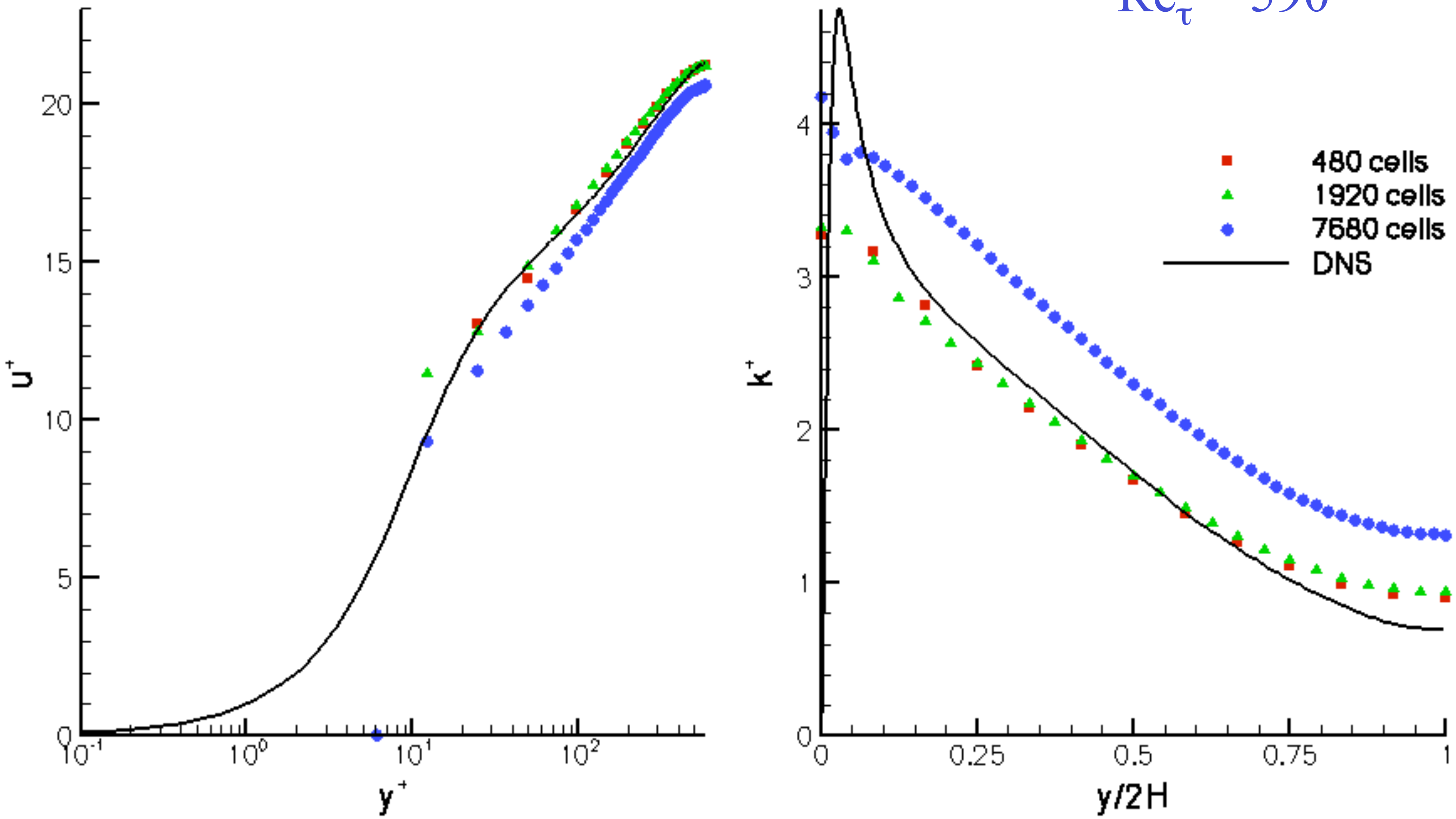


Velocity and k profiles

Grid Convergence?

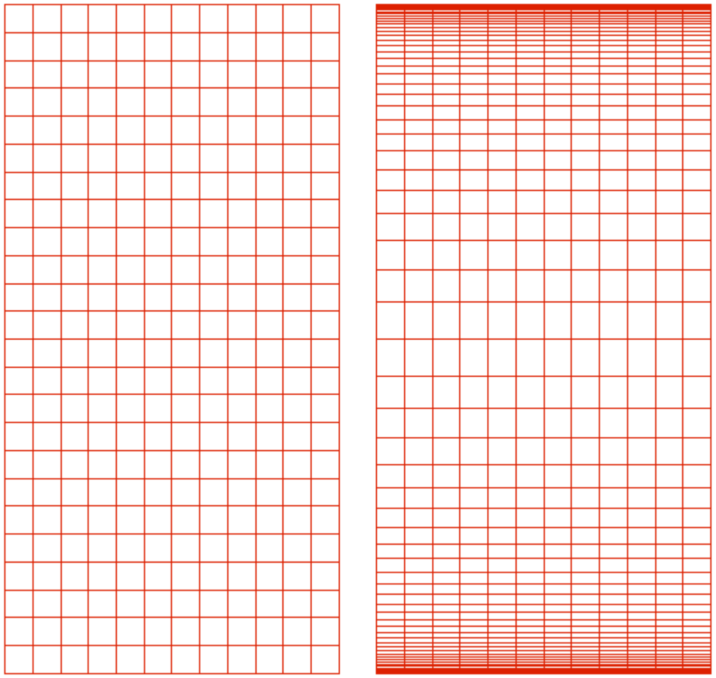
Turbulence Channel Flow

$Re_\tau = 590$



Velocity and k profiles

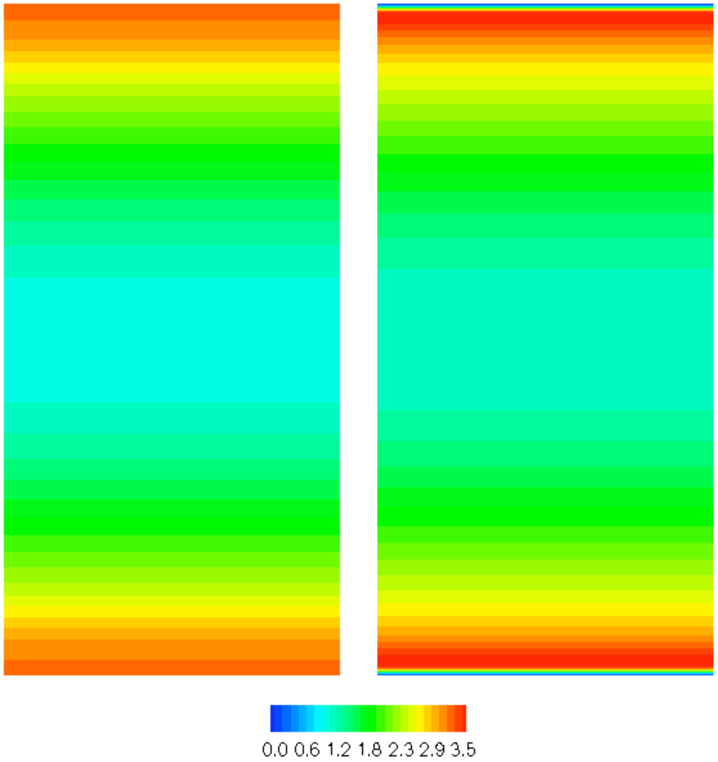
From High-Re to Low-Re k-e



Wall clustering

$y^+=30$ $y^+\sim1$

Grid



Wall boundary condition

dk/dy=0 k=0

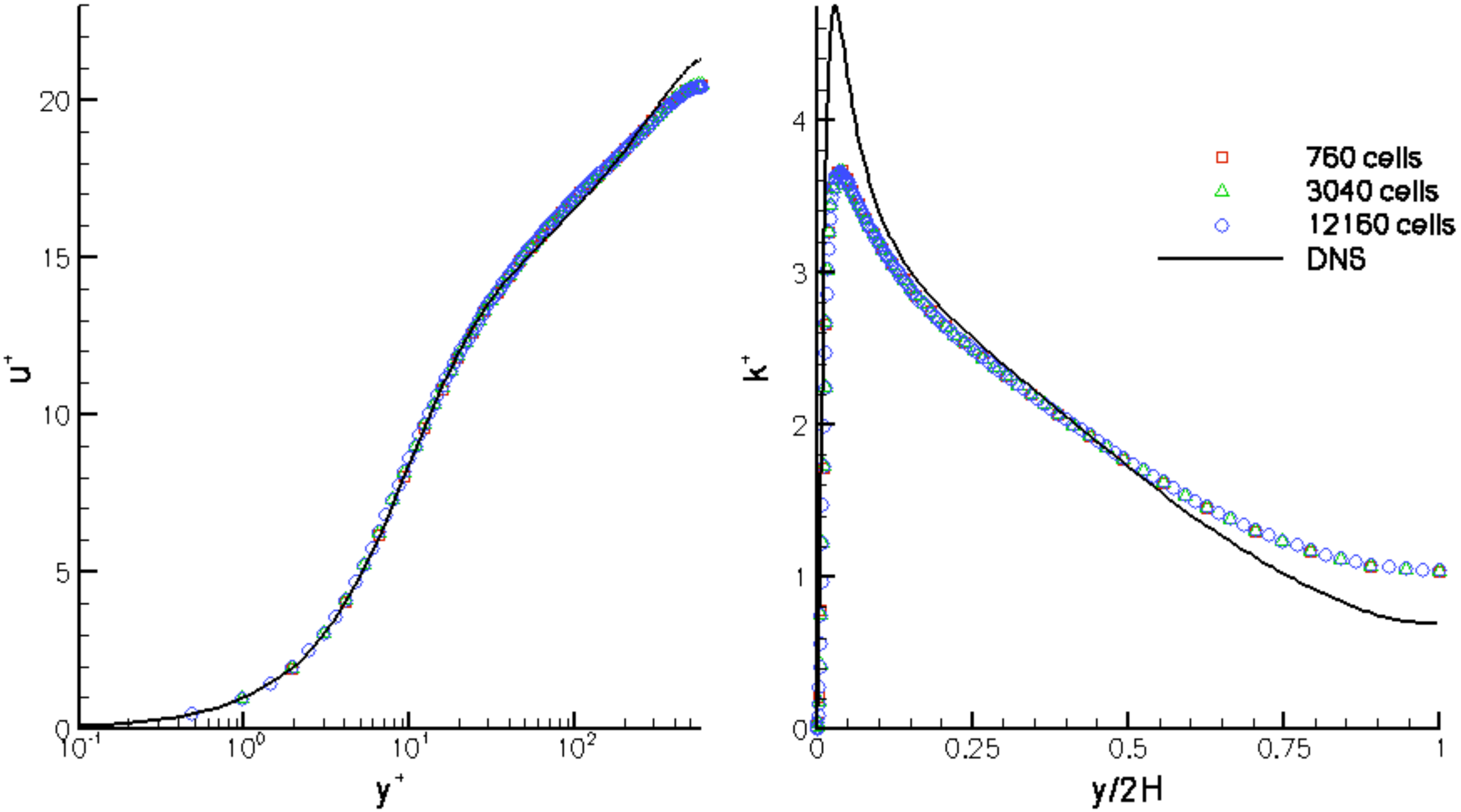
Turbulent kinetic energy

$Re_\tau = 590$

Low-Re k-ε model

Turbulence Channel Flow

$Re_\tau = 590$



Velocity and k profiles

Pros & Cons of k-ε model

+ Simple
+ Affordable
+ Reasonably accurate for wide variety of flows (without separation)
+ History effects

- Overly diffusive
- Cannot predict different flows with the same set of constants (universality)
- Source terms are stiff numerically
- Not accurate in the region close to no-slip walls where k and ε exhibit large peaks (DNS and experimental observations)
- Near wall treatment

A lot of variants have been introduced to overcome these problems

One (or more) constants become coefficients varying with S, distance from the walls, pressure gradient, etc.: RNG k-ε, realizable k-ε…

Alternatives/Improvements to k-ε models

The k-ω model was developed from the realization that most of the problems experienced by k-ε-type models are due to the modeling of the ε equation which is neither accurate or easy to solve (ε has a local extrema close to the wall)

Mathematically this is equivalent to a change of variables $\omega \sim \varepsilon/k$

The v^2-f model is based on the argument that k/ε is the correct turbulent time scale in the flow (close to the wall and in the outer region) but k is not the appropriate turbulent velocity scale

An additional equation for the correct velocity scale v^2 (independent from k) has to be solved. Moreover, the damping effect produced from the presence of the wall is NOT local (as assumed in the damping function approach) but must be accounted for globally using an elliptic equation

Reynolds Stress Models

Attempt to model directly the “new” terms appearing in the RANS equations

Mathematically is expensive because we have 6 additional equations:

$$\frac{\partial R_{ij}}{\partial t} + U_j \frac{\partial R_{ij}}{\partial x_j} = P_{ij} + \Phi_{ij} - \epsilon_{ij} + \frac{\partial J_{ijk}}{\partial x_k}$$

More importantly ONLY the production term are exact, everything else MUST be modeled

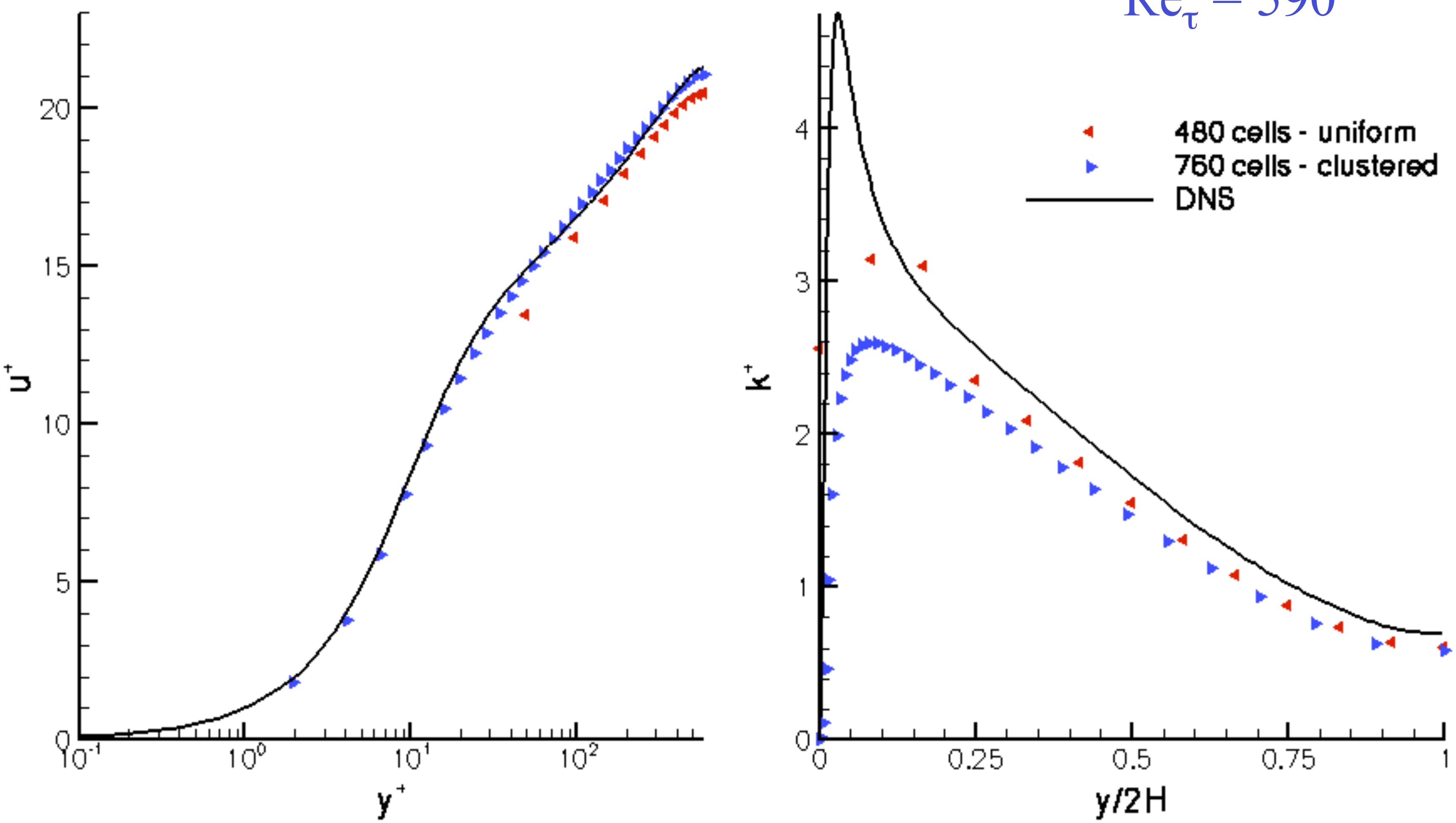
$$P_{ij} = \rho \left(R_{ik} \frac{\partial U_j}{\partial x_k} + R_{jk} \frac{\partial U_i}{\partial x_k} \right)$$

RSM are extremely stiff numerically due to the coupling between the equations and suffer of the same near-wall problems of the k-ε

k-ω model

Turbulence Channel Flow

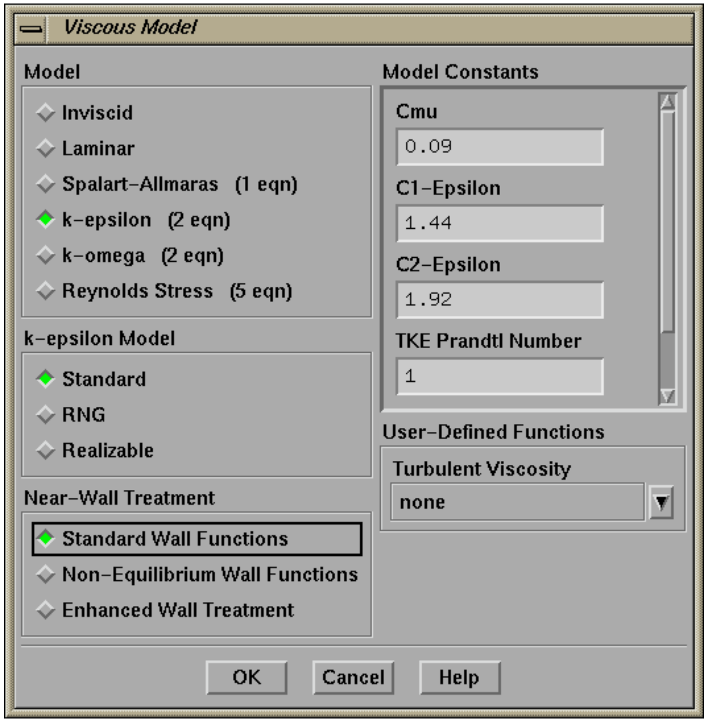
$Re_\tau = 590$



Velocity and k profiles

Models available in Fluent

Define → Models → Viscous



Viscous Model

Model

- ○ Inviscid
- ○ Laminar
- ○ Spalart-Allmaras (1 eqn)
- ● k-epsilon (2 eqn)
- ○ k-omega (2 eqn)
- ○ Reynolds Stress (5 eqn)

k-epsilon Model

- ● Standard
- ○ RNG
- ○ Realizable

Near-Wall Treatment

- ● Standard Wall Functions
- ○ Non-Equilibrium Wall Functions
- ○ Enhanced Wall Treatment

Model Constants

Cmu: 0.09

C1-Epsilon: 1.44

C2-Epsilon: 1.92

TKE Prandtl Number: 1

User-Defined Functions

Turbulent Viscosity: none

One-equation model

Two-equation model

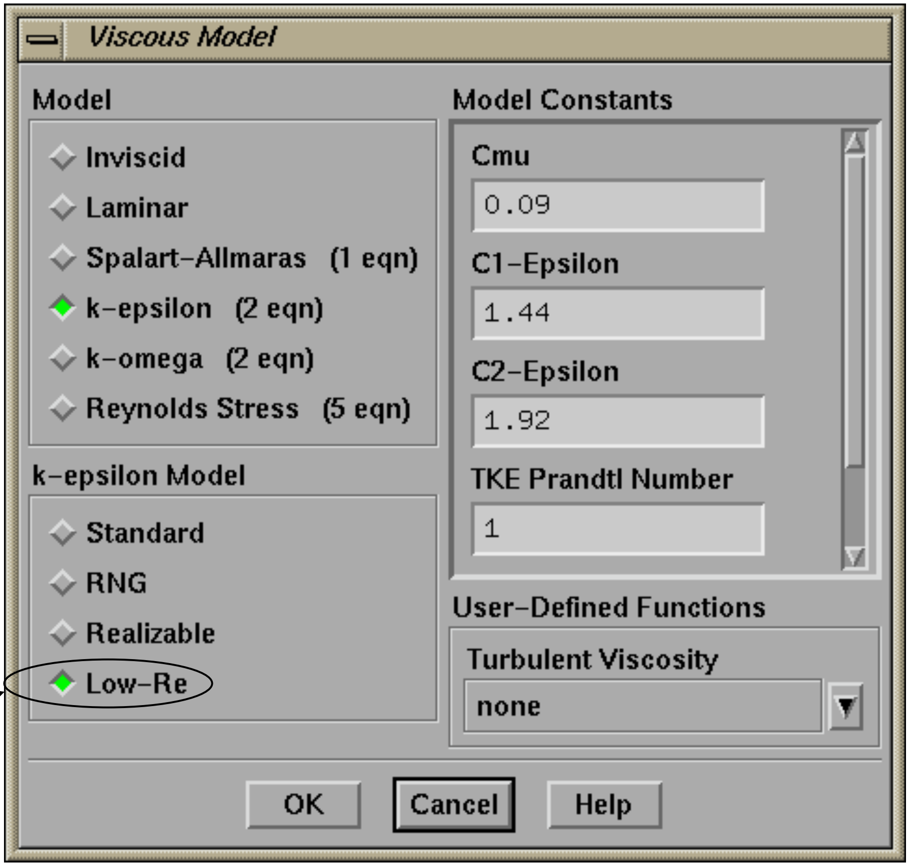
k-ε (3 versions + 3 wall treatments)
k-ω (2 versions)

Reynolds Stress model

Note that the coefficients might be adjusted!!!

Models in Fluent *hidden behind the GUI*

Some turbulence models are NOT directly available in the GUI!!!



Model
- ◇ Inviscid
- ◇ Laminar
- ◇ Spalart-Allmaras (1 eqn)
- ◆ k-epsilon (2 eqn)
- ◇ k-omega (2 eqn)
- ◇ Reynolds Stress (5 eqn)

k-epsilon Model
- ◇ Standard
- ◇ RNG
- ◇ Realizable
- ◆ Low-Re

Model Constants

Cmu: 0.09

C1-Epsilon: 1.44

C2-Epsilon: 1.92

TKE Prandtl Number: 1

User-Defined Functions

Turbulent Viscosity: none

Mixing Length

```
define/model/viscous/mixing-length y
```

Two-equation model

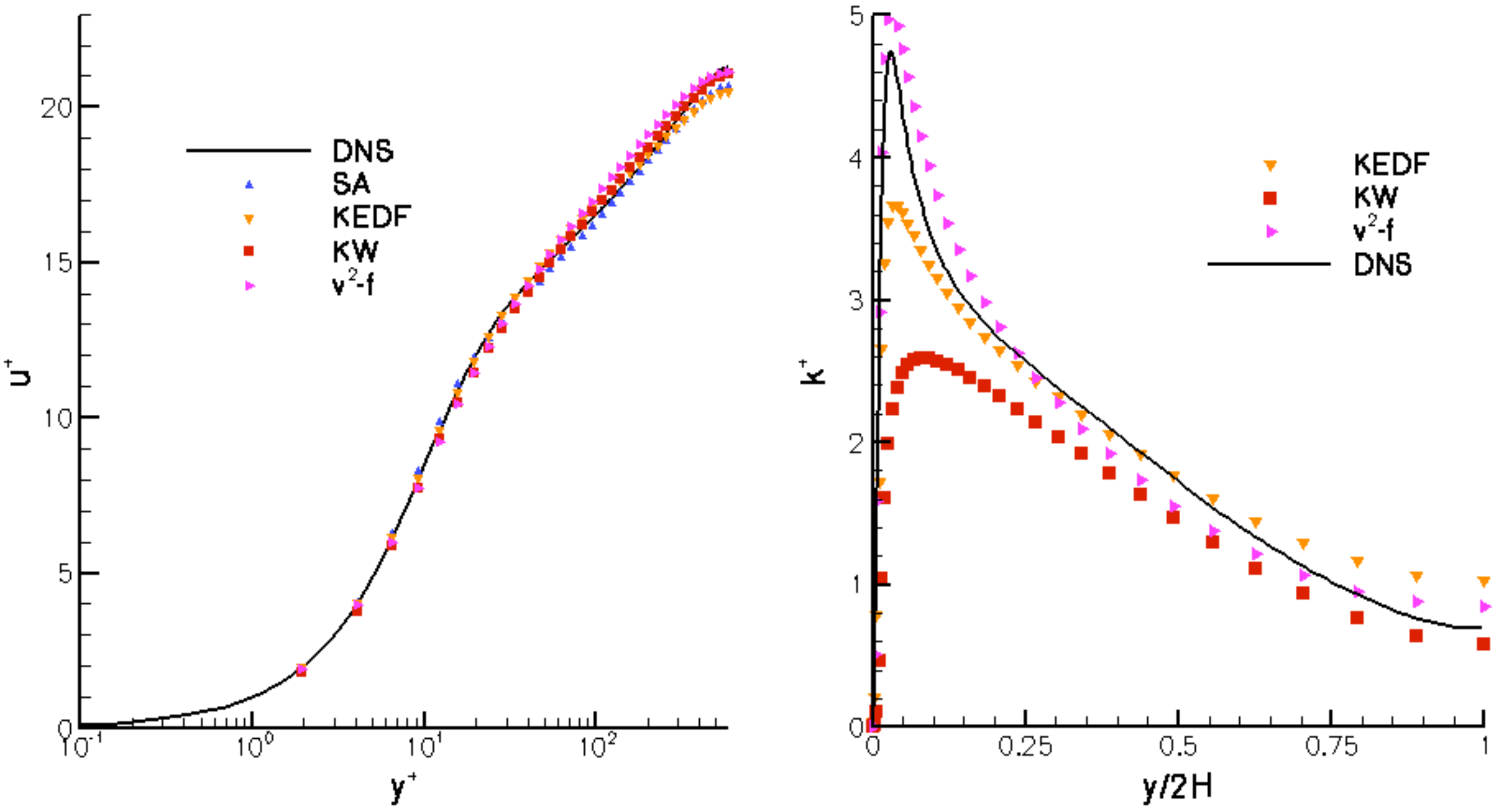
k-ε Low-Re (6 versions)

```
define/model/viscous/turbulence-expert/low-re-ke y
define/model/viscous/turbulence-expert/low-re-ke-index 2
```

Comparison of Models

Turbulence Channel Flow

$Re_\tau = 590$



Velocity and k profiles

Channel Flow Summary

Wall functions are accurate if the first grid point is in the logarithmic layer

Grid Convergence Study with wall functions approach FAILS

Damping Functions (and Two-Layer approaches) are accurate for the velocity profiles
But the turbulent kinetic energy peak is underpredicted

k-ω model is a viable alternative to k-ε and has less sensitivity to the grid clustering

SA model and v^2-f model are equivalent in capturing the velocity profile

v^2-f model is accurate in predicting the peak of turbulence kinetic energy near the wall

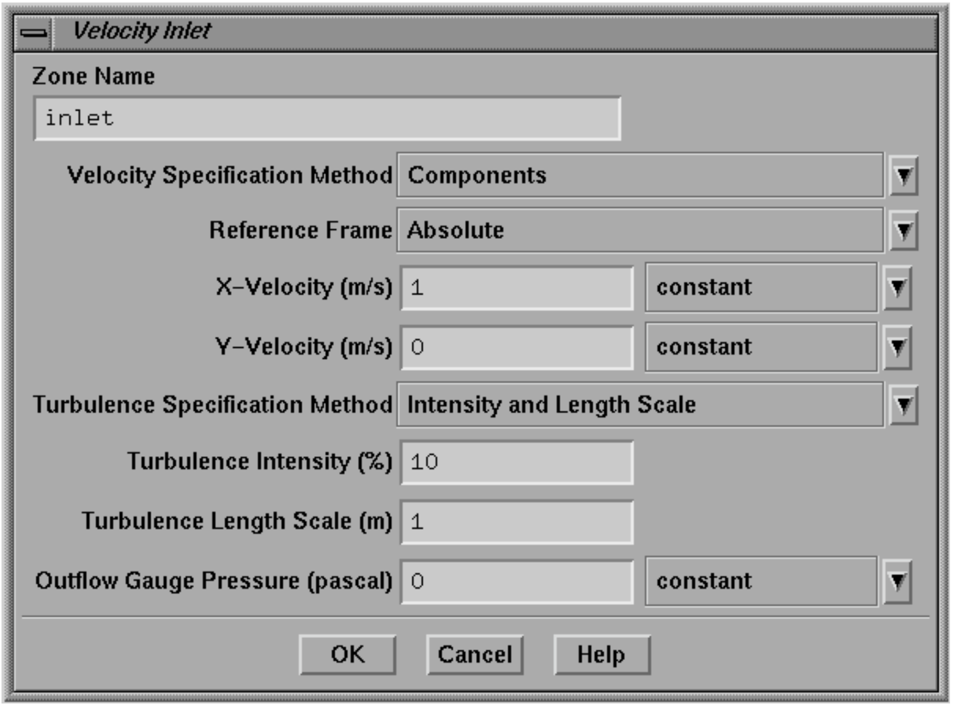
Inlet boundary conditions for turbulent quantities

At inlet boundary conditions additional quantities have to be specified in turbulent flows depending on the turbulence model selected. Typically there are three options:

1) k-ε

2) Turbulence Intensity and Turbulence Length Scale

3) Turbulence Intensity and Turbulent Viscosity



Zone Name: inlet

Velocity Specification Method: Components

Reference Frame: Absolute

X-Velocity (m/s): 1 constant

Y-Velocity (m/s): 0 constant

Turbulence Specification Method: Intensity and Length Scale

Turbulence Intensity (%): 10

Turbulence Length Scale (m): 1

Outflow Gauge Pressure (pascal): 0 constant

Example: Asymmetric Diffuser

Problem set-up

Material Properties:

$\rho = 1 \text{kg/m}^3$

$\mu = 0.0001 \text{kg/ms}$

Reynolds number:

h = 2m,

$Re = \rho Uh/\mu = \mathbf{20{,}000}$

Boundary Conditions:

Inlet profiles available from experiments

No-slip top/bottom walls

Initial Conditions:

$u = 1; v = p = 0$

Turbulence model:

Two-equation models

Solver Set-Up

Segregated Solver

Discretization:

2nd order upwind

SIMPLE

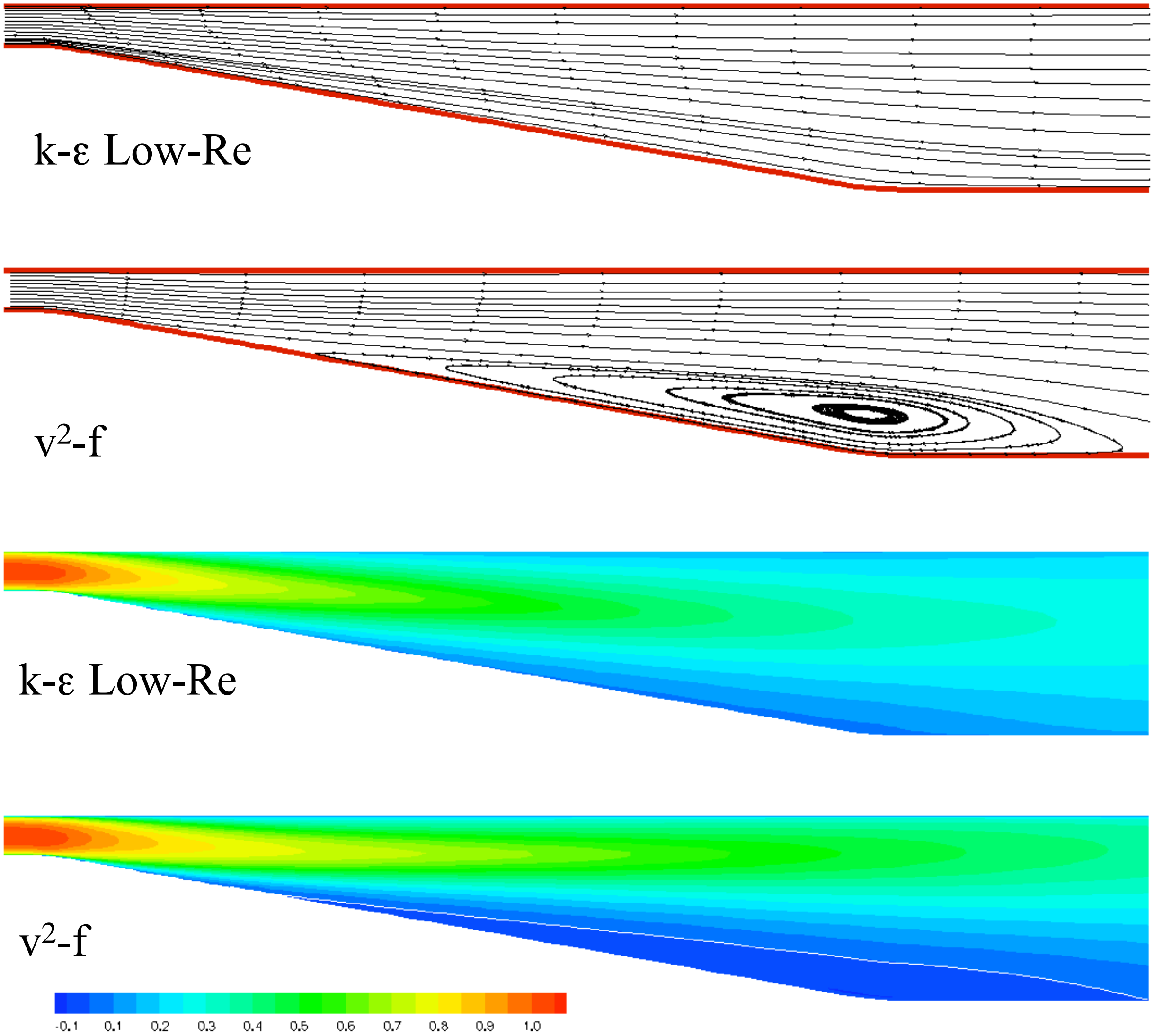
Multigrid

V-Cycle

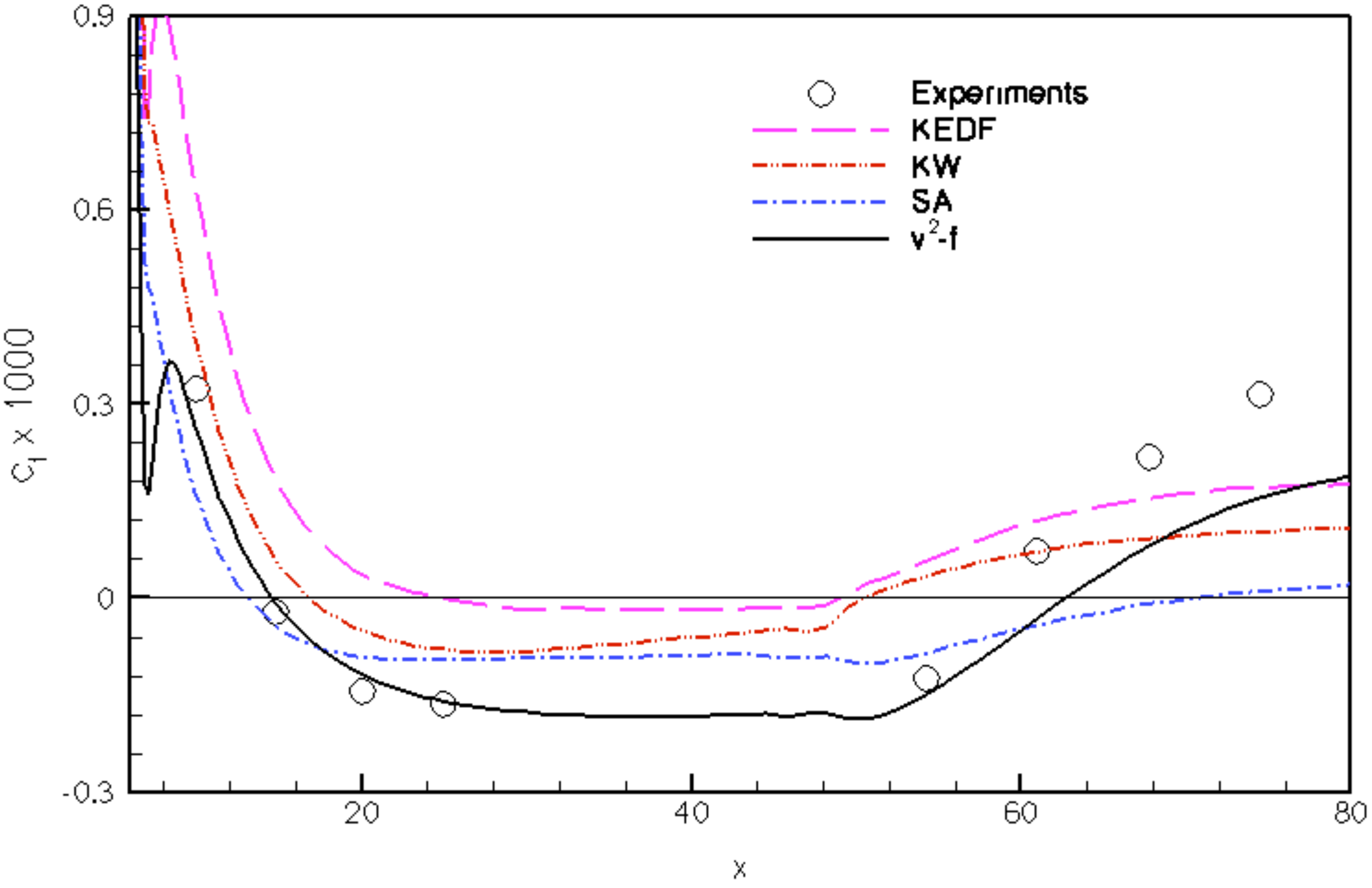
Flow in Asymmetric Diffuser

Experiments indicated the presence of a large recirculation region

k-ε models with damping function do NOT capture it



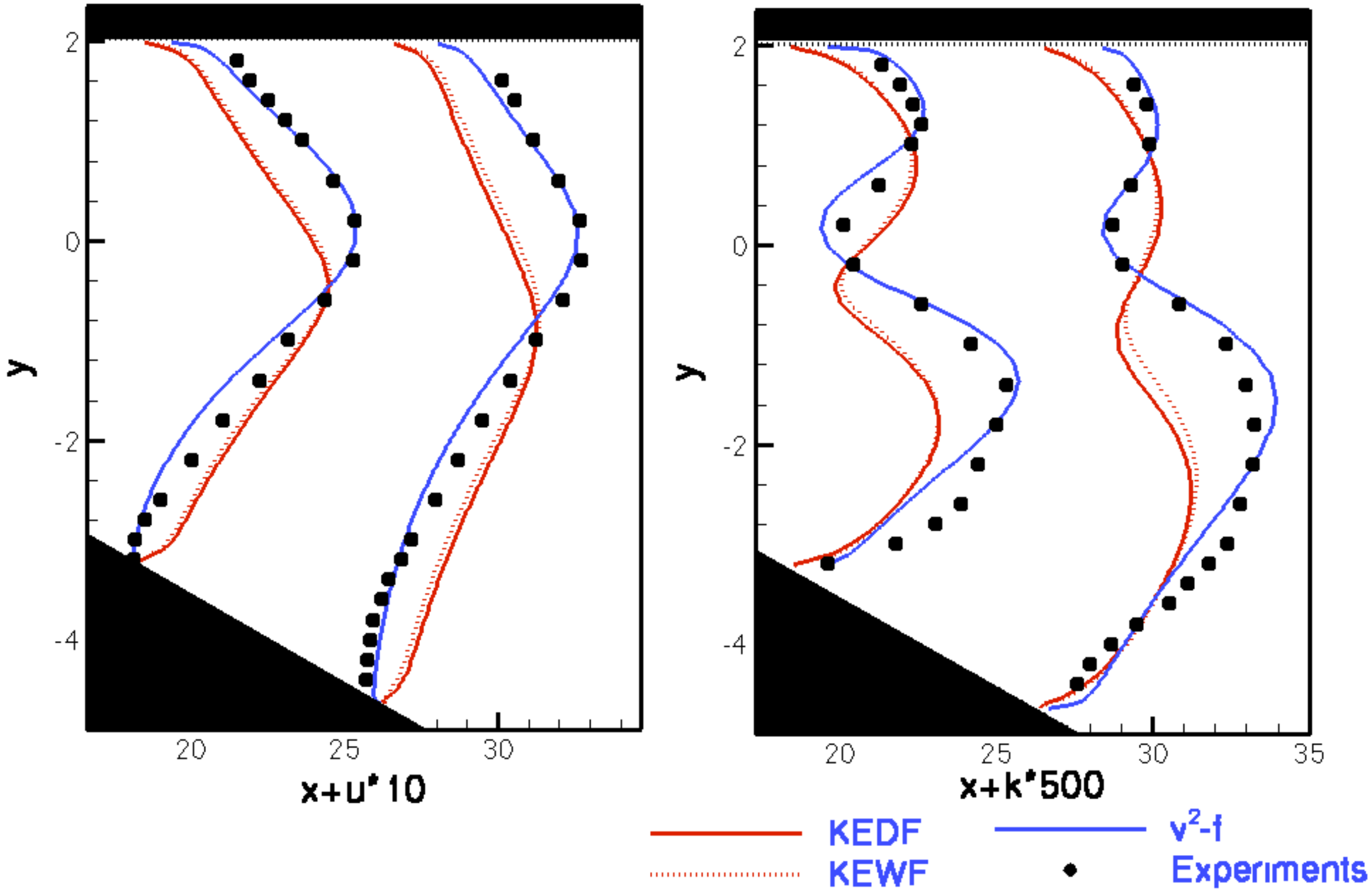
Flow in Asymmetric Diffuser



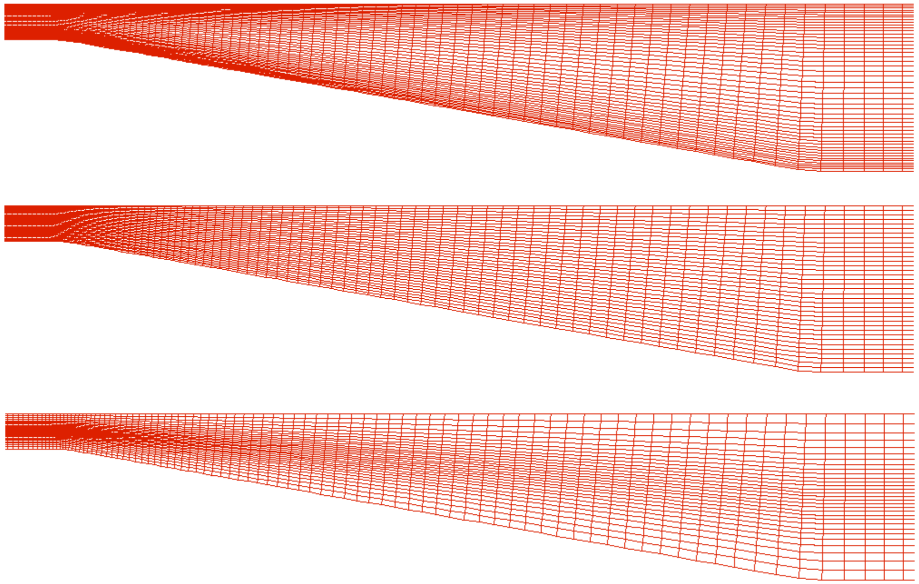
Skin friction on the bottom wall

Flow in Asymmetric Diffuser

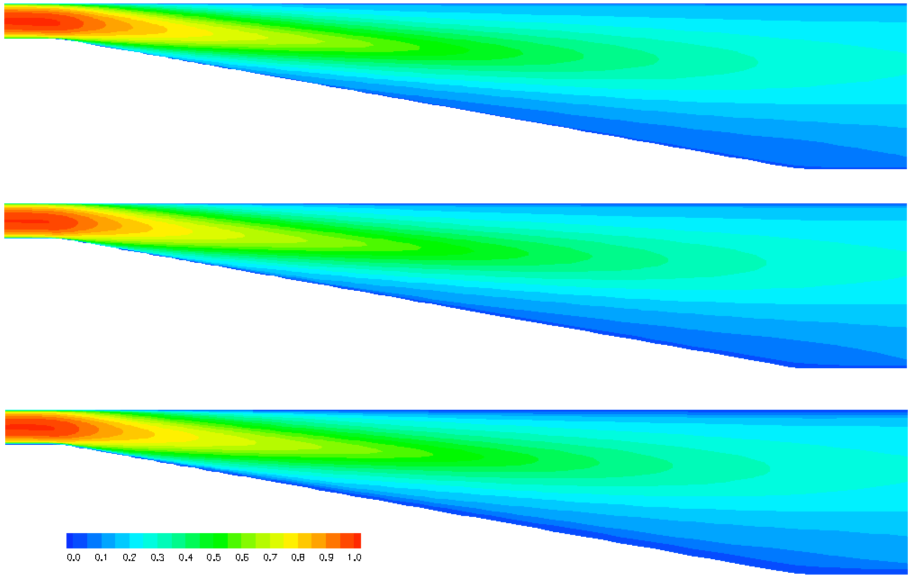
Streamwise Velocity Turbulence Kinetic Energy



k-ε with wall functions



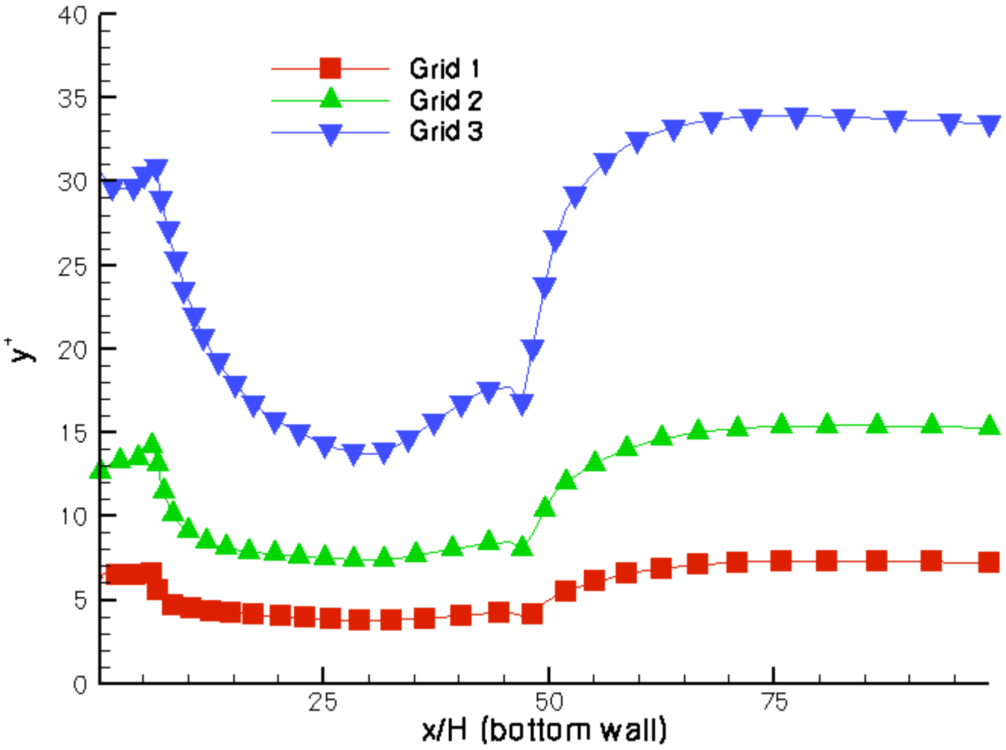
Series of grids generated
with different clustering at wall



No separation is captured!

k-ε with wall functions

In "complex" configuration it is impossible to generated a grid with the first grid point in the logarithmic layer…..



…in addition, for complicated flows with recirculation the Universal Law is inaccurate

Example: NLR Two Component Airfoil

Problem set-up

Material Properties:
$\rho = 1 kg/m^3$
$\mu = 3.98E\text{-}7 Kg/ms$

Reynolds number:
h = 1m,
Re = $\rho Uh/\mu$ = **2,512,600**

Boundary Conditions:
Constant velocity at AOA=α
No-slip walls

Initial Conditions:

$u = \cos\alpha;\ v = \sin\alpha,\ p = 0$

Turbulence model:

SA and k-ε models

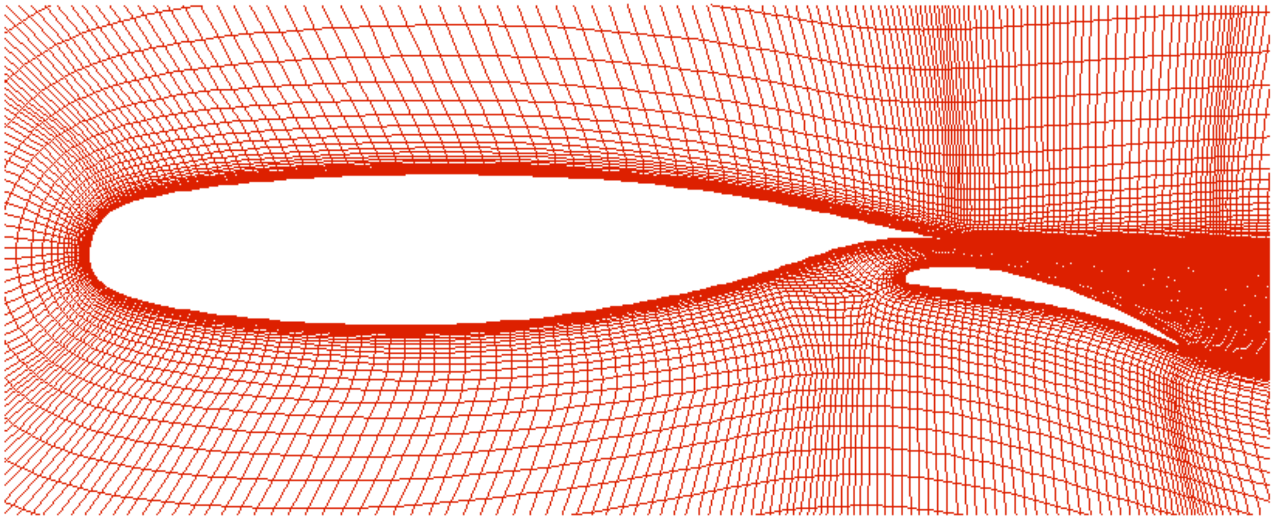
Solver Set-Up

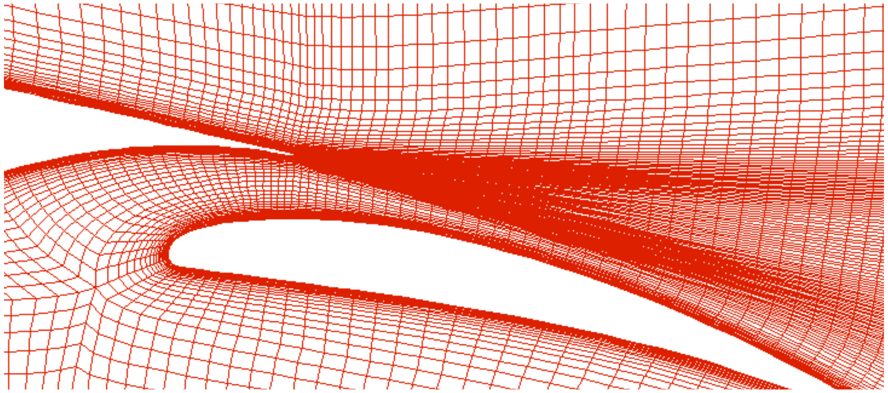
Segregated Solver

Discretization:
2nd order upwind
SIMPLE

Multigrid
V-Cycle

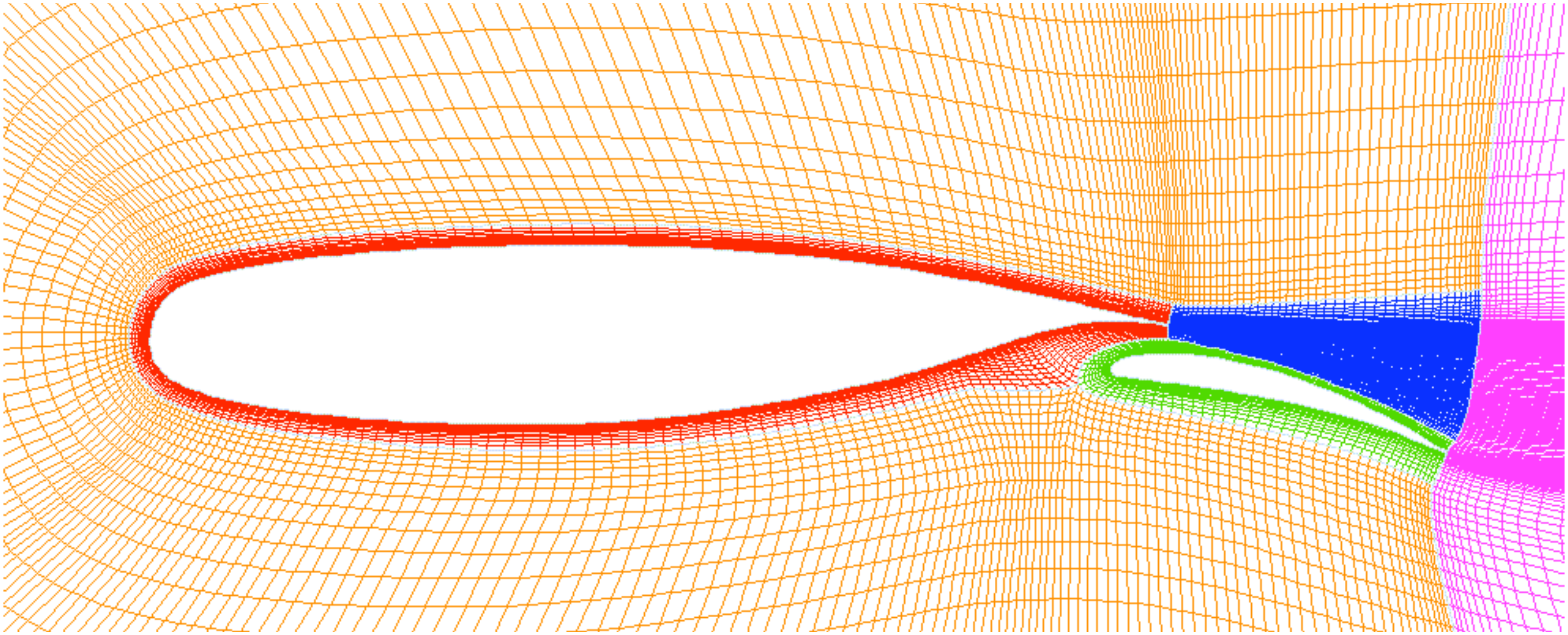
Computational Grid





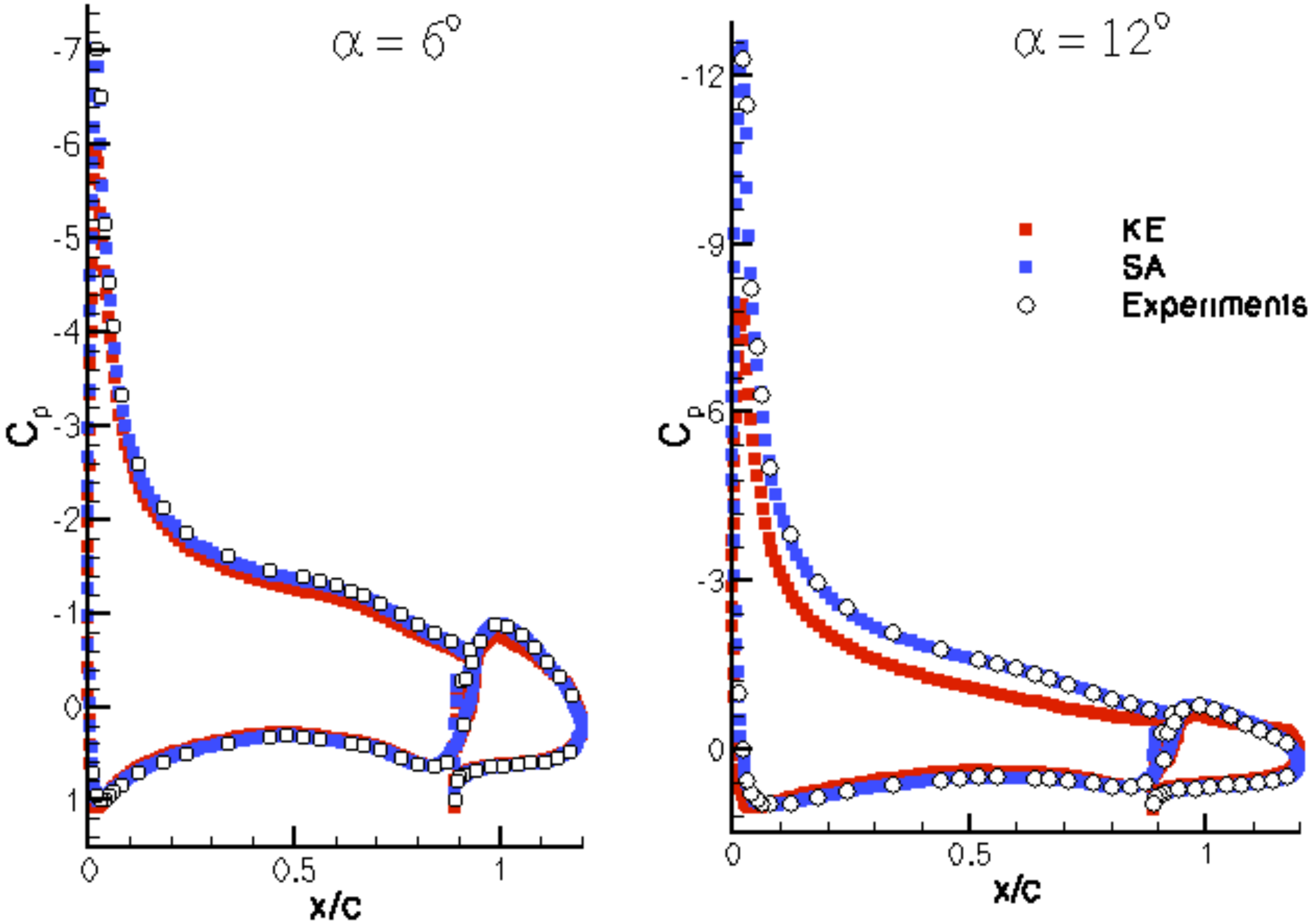
Due to the high Reynolds number resolution of the boundary layers requires extreme clustering

How this grid is generated?



Multiblock Structured Grid

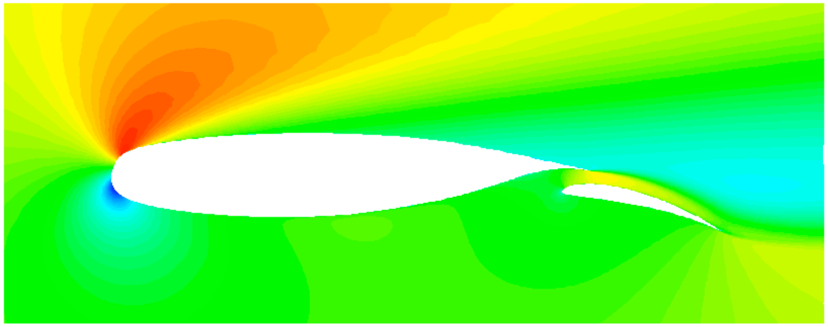
Pressure distributions



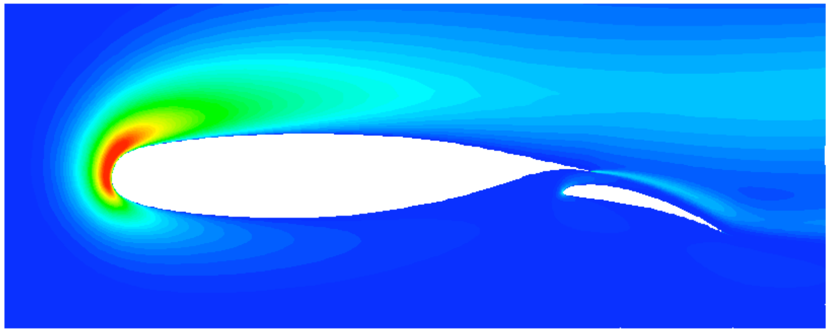
Pressure Distribution at Low and High Angle of Attack

Why k-ε fails?

Streamwise
Velocity



Turbulent
kinetic
energy



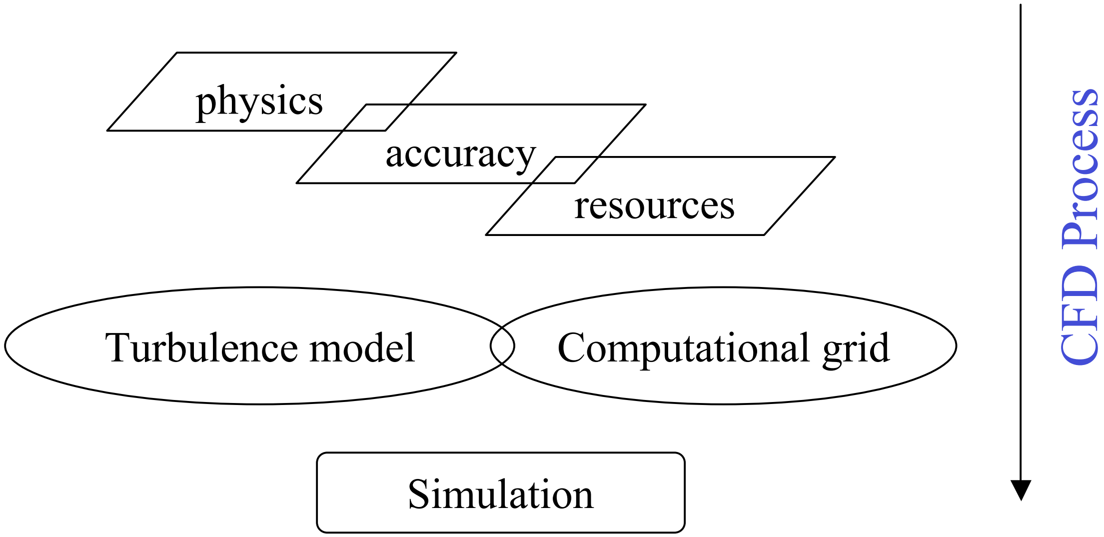
Spurious production of k in the stagnation regions
Fix: Use of a production limiter:

```
define/model/viscous/turbulence-expert/kato-launder-model y
```

Guidelines

Simulations of turbulence flows require "decisions" based on:

1) Flow Physics
   to characterize the flow features (turbulence, high gradients, etc.)
2) Computational requirements
   to evaluate the grids resolution required for a certain accuracy
3) Project Requirement
   to evaluate the need for sophisticated turbulence models



Guidelines

Modeling procedure:

- Determine relevant Reynolds number to estimate if the flow is turbulent
- Select a turbulence model option and a near-wall treatment
- Estimate the physical dimension of the first grid point off the wall (y^+)
- Generate the grid
- Perform the simulation
- "Reality" check (experiments, literature, model consistency, grid resolution)

Estimating y^+

Definition: $y^+ = \rho\, y_p\, u_\tau / \mu$

$$y^+ = \rho\, y_p\, u_\tau / \mu$$

$$u_\tau = U_e\, (c_f/2)^{1/2}$$

To estimate u_τ we can use "classic" relationships for the skin friction:

- Flat Plate: $c_f/2 \sim 0.052\, (Re_x)^{-0.142}$
- Pipe: $c_f/2 \sim 0.046\, (Re_x)^{-0.2}$

Note that there are different laws for different Re number ranges

“Final” Guidelines

For k-ε simulations:

- Two-layer is preferable over wall-functions (grid dependence + accuracy)
- Realizable k-ε or Kato&Launder limiter have to be used

For k-ω simulations:

- SST is usually better than standard
- Grid should be clustered at wall

SA is usually a good compromise between accuracy and simplicity. It also has very good convergence properties (as compared to the two-equation models)

Reynolds stress model is expensive and it require a good initial guess (typically a k-ε-type simulation)

Summary of turbulence models in Commercial Codes

	Zero equation	**One equation**	**Two equation**	**RSM**	**Non-Linear Models**	**Custom**
FLUENT	y	y	y	y	n	y
StarCD	n	n	y	n	y	y
CFX	y	y	y	y	n	y

An example of User Defined Programming

Low-Re k-ε model

Development of a custom turbulence model can be accomplished using the UDFs.

k-ε model with damping functions formulation:

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\frac{(\mu + \mu_t)}{\rho}\frac{\partial U_i}{\partial x_j}\right]$$

RANS

$$\begin{aligned} u_i \frac{\partial k}{\partial x_i} &= P - \epsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \\ u_i \frac{\partial \epsilon}{\partial x_i} &= \frac{\epsilon}{k}\left(f_1 C_{\epsilon_1} P - f_2 C_{\epsilon_2}\epsilon\right) + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon}\right)\frac{\partial \epsilon}{\partial x_j}\right] \end{aligned}$$

Additional
Transport
Equations

An example of User Defined Programming

Low-Re k-ε model

$$\nu_t = C_\mu f_\mu \frac{k^2}{\epsilon}$$ Eddy Viscosity

$$P = \frac{\mu_t}{\rho} S^2$$ Turbulent Kinetic Energy Production

$$S = \sqrt{2 S_{ij} S_{ij}}$$ Strain Rate Magnitude $$S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)$$

$$\begin{aligned} f_1 &= 1 \\ f_2 &= \left(1 - \frac{2}{9} e^{-Re_T^2/36}\right)\left(1 - e^{-Re_y/12}\right) \\ f_\mu &= tanh(0.008 Re_y)\left(1 + \frac{4}{Re_t^{0.75}}\right) \end{aligned}$$

Damping functions

$$Re_T = k^2/\nu\epsilon$$
$$Re_y = \sqrt{k}y/\nu$$

Turbulent Reynolds number definitions

$$k_{wall} = 0$$
$$\epsilon_{wall} = 2\nu \frac{d}{dy}\sqrt{k}$$

Wall boundary conditions

Required UDF Routines

Source Terms	`DEFINE_SOURCE(k_source, t, eqn)` `DEFINE_SOURCE(d_source, t, eqn)`
Diffusivity	`DEFINE_PROPERTY(ke_diffusivity, c, t, eqn)`
Boundary Conditions	`DEFINE_PROFILE(wall_d_bc, domain)`
Eddy Viscosity	`DEFINE_TURBULENT_VISCOSITY(ke_ mut, c, t)`
Adjust Routine	`DEFINE_ADJUST(turb_adjust, domain)`
Initialization Routine	`DEFINE_INIT(turb_adjust, domain)`

Required field variables

Density	`C_R(cell,thread)`
Molecular viscosity	`C_MU(cell,thread)`
Eddy viscosity	`C_MU_T(cell,thread)`
Strain Rate Magnitude	`Strain_rate(cell,thread)`
Wall distance	`C_WALL_DIST(cell,thread)`

Remark: **`C_WALL_DIST(c,t)`** is only computed for special cases as it is only used by few turbulence models.

UDF Header

```
#include "udf.h"
```

Required: includes all Fluent macros

```
/* Turbulence model constants */
#define  C_MU    0.09
#define  SIG_TKE 1.0
#define  SIG_TDR 1.3
#define  C1_D    1.44
#define  C2_D    1.92
```

Constant definitions (global)

```
/* User-defined scalars */
enum
{
  TKE,
  TDR,
  N_REQUIRED_UDS
};
```

Assign a number to each scalar

Damping Functions

These are defined on a cell-by-cell basis

```
/* Reynolds number definitions */
real Re_y(cell_t c, Thread *t)
{ return C_R(c,t)*sqrt(C_UDSI(c,t,TKE))*C_WALL_DIST(c,t)/C_MU_L(c,t);}

real Re_t(cell_t c, Thread *t)
{ return C_R(c,t)*SQR(C_UDSI(c,t,TKE))/C_MU_L(c,t)/C_UDSI(c,t,TDR);}

/* Damping Functions */
real f_mu(cell_t c, Thread *t)
{ return tanh(0.008*Re_y(c,t))*(1.+4/pow(Re_t(c,t),0.75));}

real f_1(cell_t c, Thread *t)
{ return 1.;}

real f_2(cell_t c, Thread *t)
{ return (1.-2/9*exp(-Re_t(c,t)*Re_t(c,t)/36))*(1.-exp(-Re_y(c,t)/12));}
```

Source Term Routines

The production term in the k equation is: $P - \epsilon$

ε is obtained from the definition of the eddy viscosity to increase the coupling between the equations and to define an implicit term

$$\nu_t = C_\mu f_\mu \frac{k^2}{\epsilon}$$

```
DEFINE_SOURCE(k_source, c, t, dS, eqn)
{
  real  G_k;

  G_k = C_MU_T(c,t)*SQR(Strainrate_Mag(c,t));

  dS[eqn] = -2.*C_R(c,t)*C_R(c,t)*C_MU*f_mu(c,t)*C_UDSI(c,t,TKE)/C_MU_T(c,t);
  return G_k - C_R(c,t)*C_R(c,t)*C_MU*f_mu(c,t)*SQR(C_UDSI(c,t,TKE))/C_MU_T(c,t);
}
```

Source Term Routines

The production term in the ε equation is: $\frac{\epsilon}{k}(f_1 C_{\epsilon_1} P - f_2 C_{\epsilon_2}\epsilon)$

It contains already both k and ε. No need for manipulations!

```
DEFINE_SOURCE(d_source, c, t, dS, eqn)
{
  real G_k;

  G_k = C_MU_T(c,t)*SQR(Strainrate_Mag(c,t));

  dS[eqn] = C1_D*f_1(c,t)*G_k/C_UDSI(c,t,TKE)
     - 2.*C2_D*f_2(c,t)*C_R(c,t)*C_UDSI(c,t,TDR)/C_UDSI(c,t,TKE);
  return C1_D*f_1(c,t)*G_k*C_UDSI(c,t,TDR)/C_UDSI(c,t,TKE)
     - C2_D*f_2(c,t)*C_R(c,t)*SQR(C_UDSI(c,t,TDR))/C_UDSI(c,t,TKE);
}
```

Diffusivity

The diffusion terms in the scalar equations are set-up together

```
DEFINE_DIFFUSIVITY(ke_diffusivity, c, t, eqn)
{
 real diff;
 real mu = C_MU_L(c,t);
 real mu_t = C_R(c,t)*C_MU*f_mu(c,t)*SQR(C_UDSI(c,t,TKE))/C_UDSI(c,t,TDR);

 switch (eqn)
  {
  case TKE:
   diff = mu_t/SIG_TKE + mu;
   break;
  case TDR:
   diff = mu_t/SIG_TDR + mu;
   break;
  default:
   diff = mu_t + mu;
  }
 return diff;
}
```

But each equation can have a different value

Eddy Viscosity

The eddy viscosity is set in the adjust routine (called at the beginning of each Iteration) and it is used in the mean flow and in the scalar equations

```
DEFINE_TURBULENT_VISCOSITY(ke_mut, c, t)
{

       return C_R(c,t)*C_MU*f_mu(c,t)*SQR(C_UDSI(c,t,TKE))/C_UDSI(c,t,TDR);
}
```

Wall Boundary Conditions

Only the boundary condition for ε is complicated because it requires the value of the derivative of k (the square root of k)

$$k_{wall} = 0$$
$$\epsilon_{wall} = 2\nu \frac{d}{dy}\sqrt{k}$$

```
DEFINE_PROFILE(wall_d_bc, t, position)
{
  face_t f;
  cell_t c0;
  Thread *t0 = t->t0; /* t0 is cell thread */
  real xw[ND_ND], xc[ND_ND], dx[ND_ND], rootk, dy, drootkdy;

  begin_f_loop(f,t)
    {
      c0 = F_C0(f,t);
      rootk = sqrt(MAX(C_UDSI(c0,t0,TKE), 0.));
      F_CENTROID(xw,f,t);
      C_CENTROID(xc,c0,t0);
      NV_VV(dx, =, xc, -, xw);
      dy = ND_MAG(dx[0], dx[1], dx[2]);
      drootkdy = rootk/dy;
      F_PROFILE(f,t,position) = 2.*C_MU_L(c0,t0)/C_R(c0,t0)*drootkdy*drootkdy;
    }
  end_f_loop(f,t)
}
```

The derivative is rootk/dy
rootk is the sqrt of k in
the adjacent cell center
dy is the distance between
cell and face center

Set-Up the Problem

Set-Up a case using one of the standard models (SA uses wall distance)

Define two scalars (TKE, TDR)

Compile and Attach the UDFs

Hook the various functions
(eddy viscosity, scalar diffusivity, sources, bc, init/adjust)

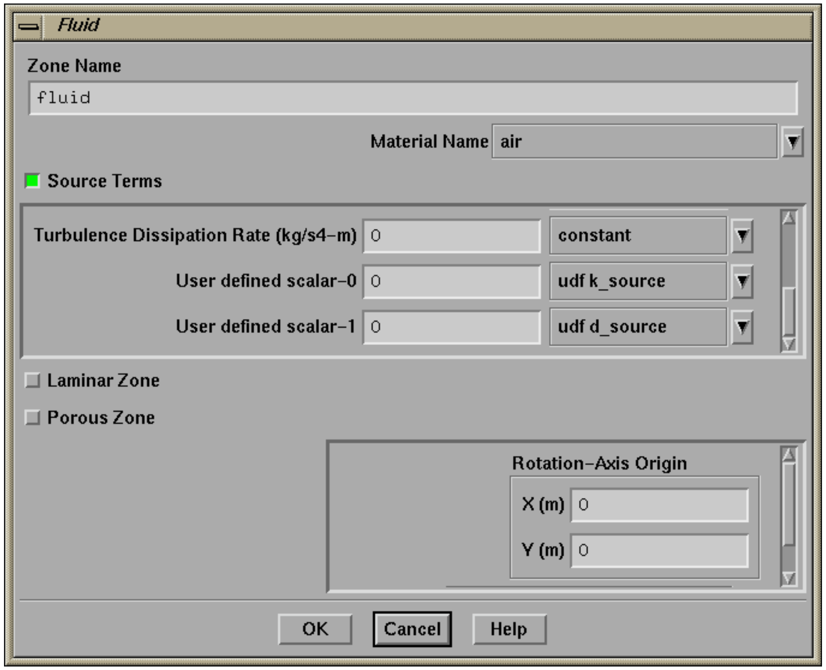
Deactivate the equations for the standard turbulence model

Set the under-relaxation factors for the scalars (<1)

Initialize and solve the RANS + TKE and TDR

Attach the UDF

Source terms



Zone Name: fluid

Material Name: air

☑ Source Terms

Turbulence Dissipation Rate (kg/s4-m): 0, constant

User defined scalar-0: 0, udf k_source

User defined scalar-1: 0, udf d_source

☐ Laminar Zone

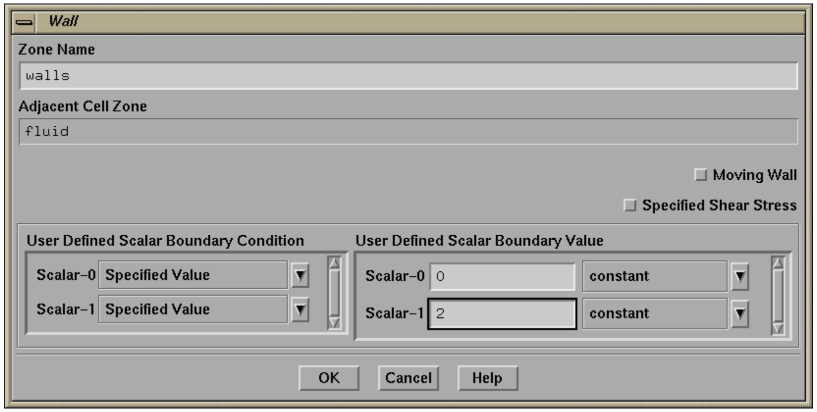
☐ Porous Zone

Rotation-Axis Origin

X (m): 0

Y (m): 0

Boundary conditions



Zone Name: walls

Adjacent Cell Zone: fluid

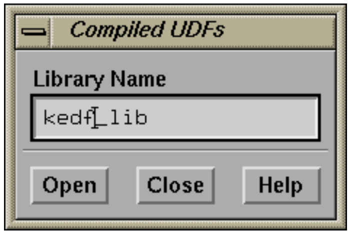
☐ Moving Wall

☐ Specified Shear Stress

User Defined Scalar Boundary Condition		User Defined Scalar Boundary Value		
Scalar-0	Specified Value	Scalar-0	0	constant
Scalar-1	Specified Value	Scalar-1	2	constant

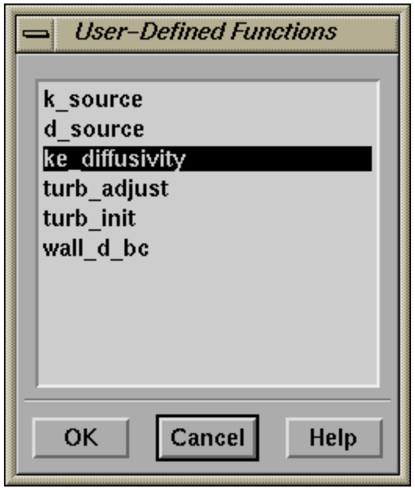
Open UDF Library

After compiling the library
It can be opened in Fluent



Library Name: kedf_lib

Output in the text window
Are the functions available



k_source
d_source
ke_diffusivity
turb_adjust
turb_init
wall_d_bc