Create a DES turbulence model from kOmegaSST RANS model

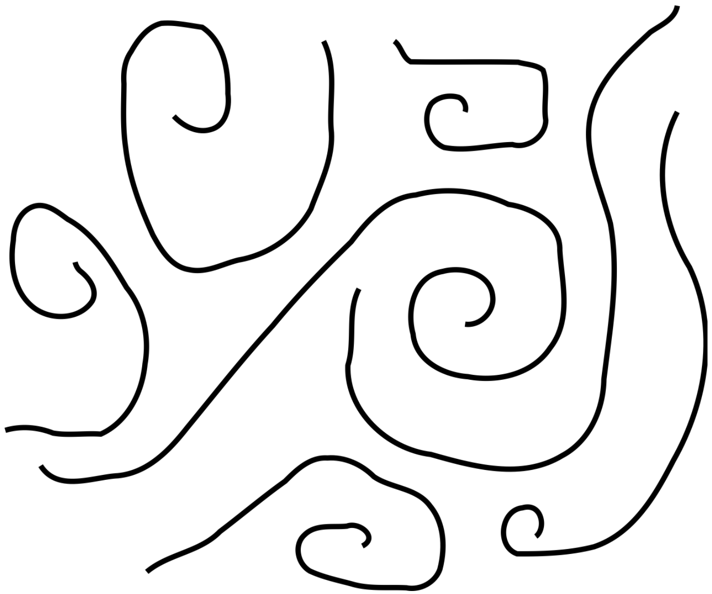
8th OFWS training by D. Chrisk

Revised by Y. Takagi

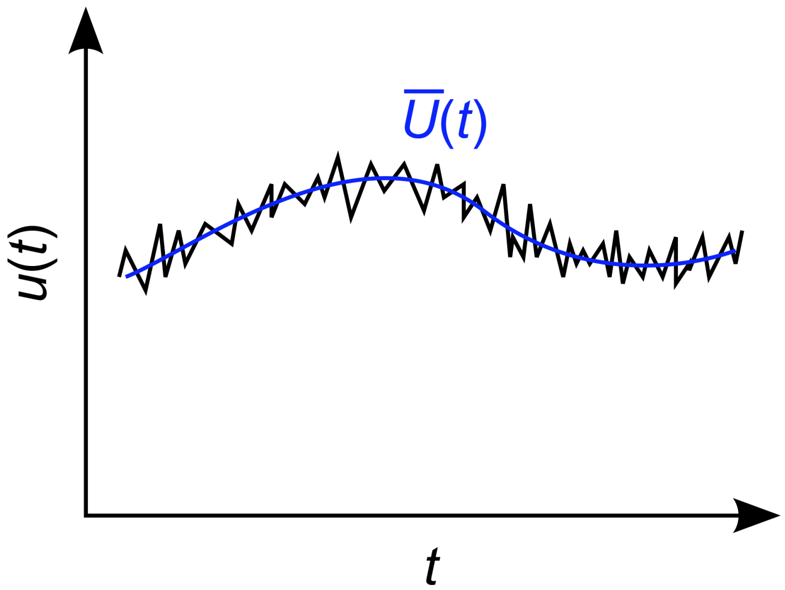
Open CAE seminar@Kansai, 2013.7.6

Turbulent flow simulation

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



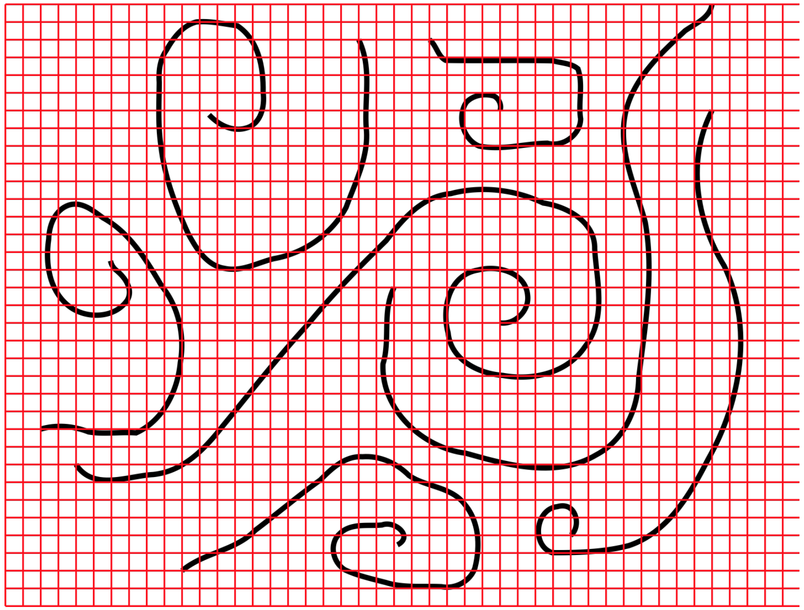
Vortex (eddy) field



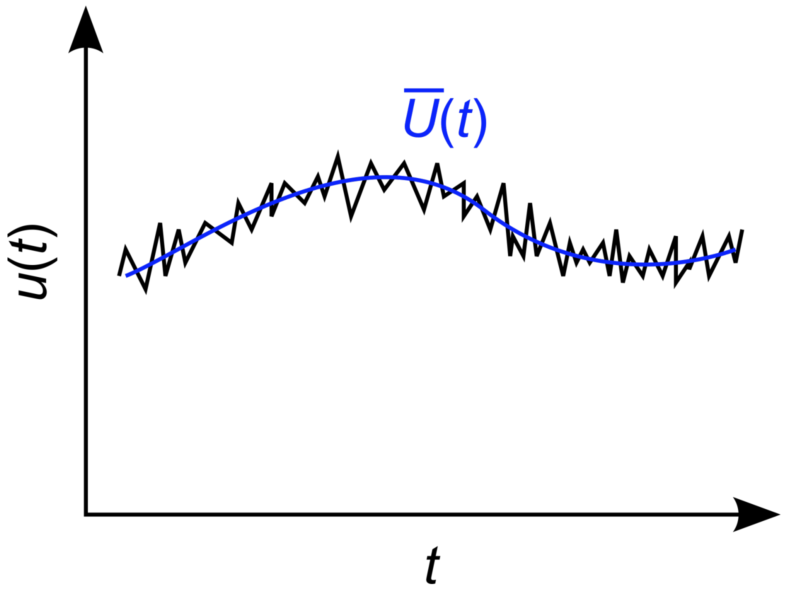
Reynolds average

Turbulent flow simulation

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



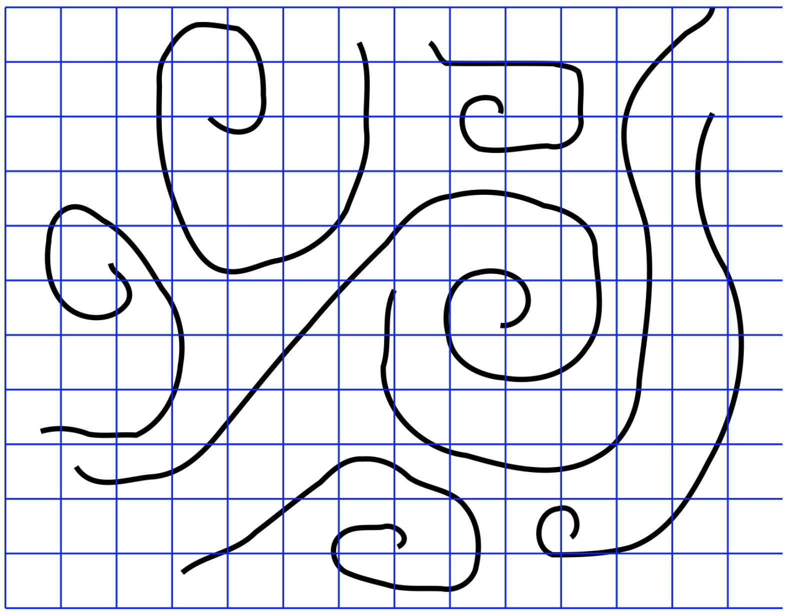
DNS grid, *u*



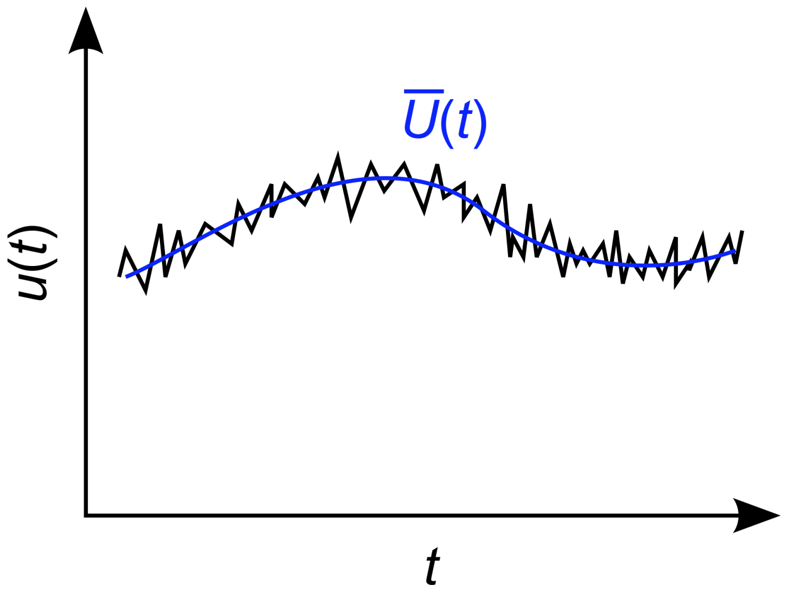
Reynolds average

Turbulent flow simulation

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



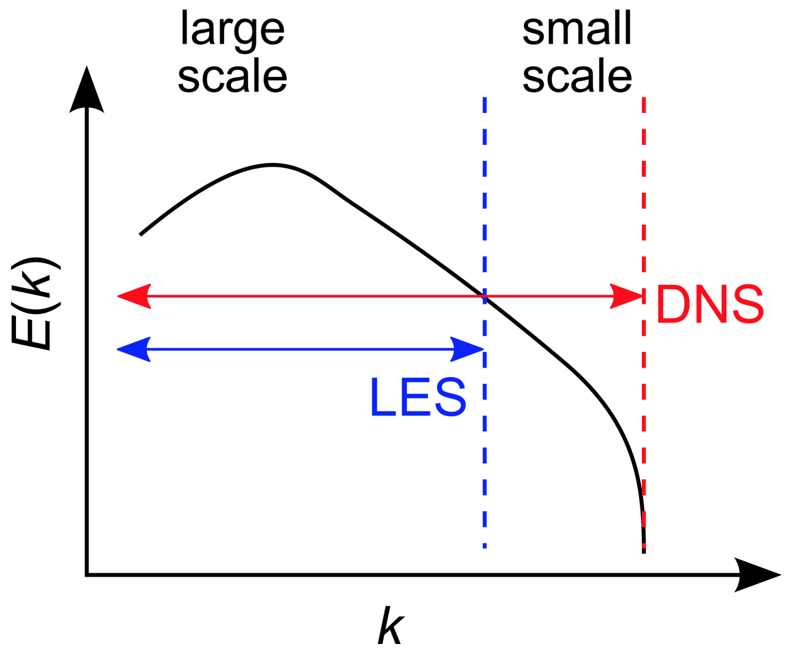
LES grid, $\overline{u} = u - u'$



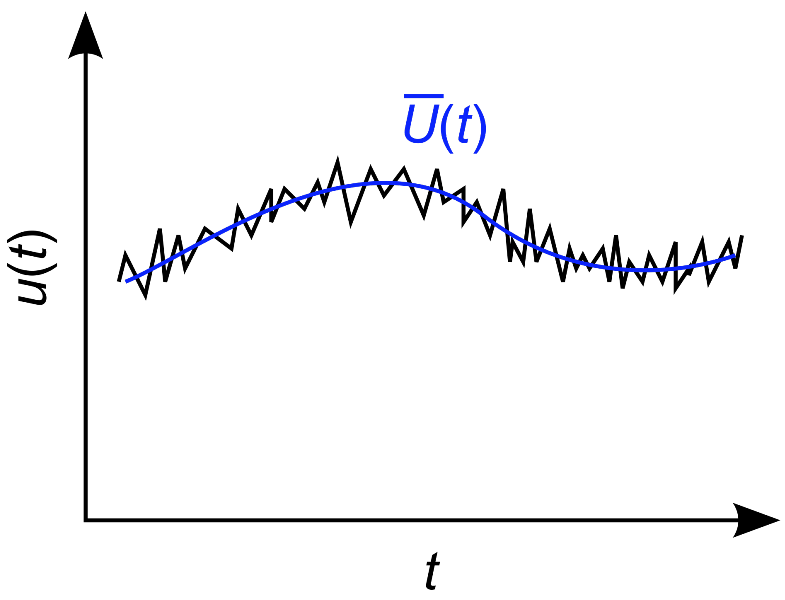
Reynolds average

Turbulent flow simulation

	DNS	LES	RANS
Modeling	No	Subgrid scale	Reynolds average
Accuracy	◎	○	△
Cost	×	○	◎



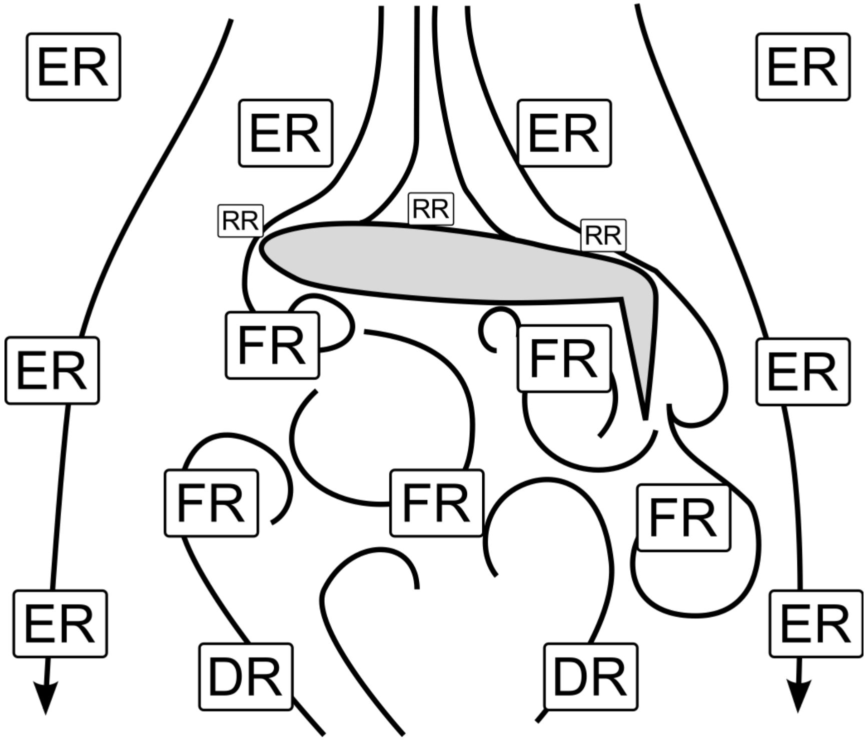
Filtering approach



Reynolds average

Detached-eddy simulation (DES)

- P. R. Spalart (1997)[1)]:
  - *We name the new approach "Detached-Eddy Simulation" (DES) to emphasize its distinct treatments of attached and separated regions.*

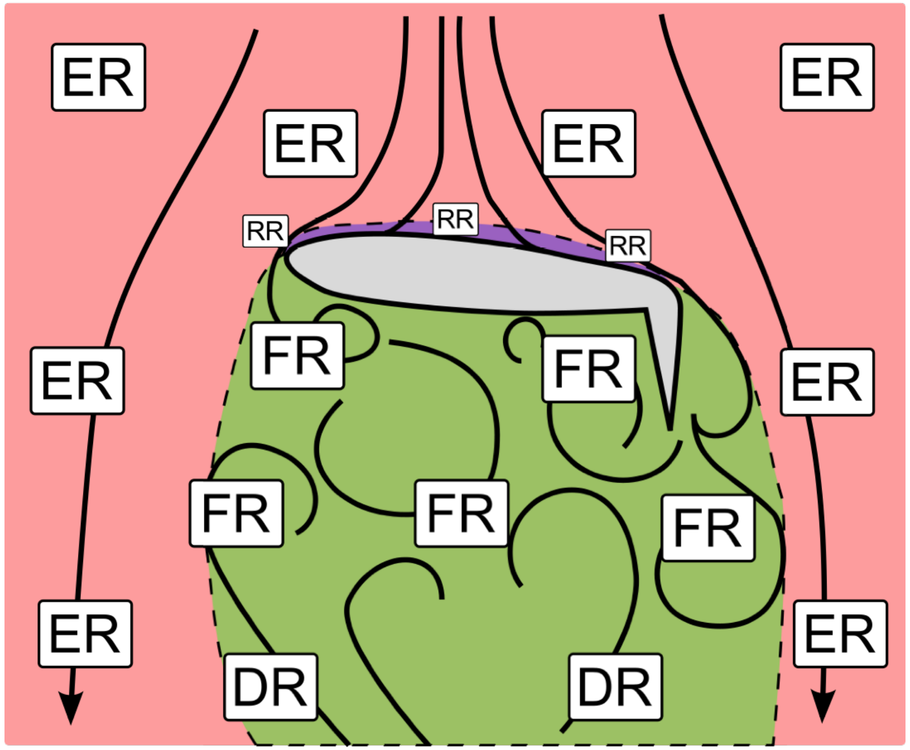


Ref. 2)

Super-Region	Region
Euler (ER)	
RANS (RR)	Viscous (VR)
	Outer (OR)
LES (LR)	Viscous (VR)
	Focus (FR)
	Departure (DR)

Detached-eddy simulation (DES)

- P. R. Spalart (1997)[1)]:
  - *We name the new approach "Detached-Eddy Simulation" (DES) to emphasize its distinct treatments of attached and separated regions.*



Ref. 2)

Super-Region	Region
Euler (ER)	
RANS (RR)	Viscous (VR)
	Outer (OR)
LES (LR)	Viscous (VR)
	Focus (FR)
	Departure (DR)

Coupling with momentum equation through viscosity

- RANS

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\mathbf{U}\mathbf{U}\right) - \nabla \cdot \left(\left(\nu + \underline{\nu_t}\right)\left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}}\right)\right) = \nabla p$$

Turbulent viscosity

- LES

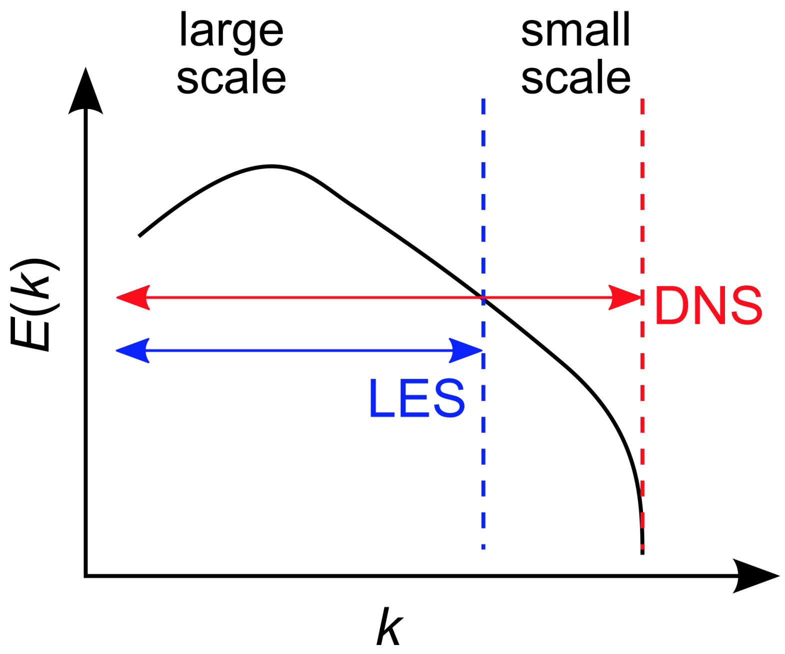
$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \left(\mathbf{U}\mathbf{U}\right) - \nabla \cdot \left(\left(\nu + \underline{\nu_{\mathrm{SGS}}}\right)\left(\nabla \mathbf{U} + (\nabla \mathbf{U})^{\mathrm{T}}\right)\right) = \nabla p$$

Sub-grid scale viscosity

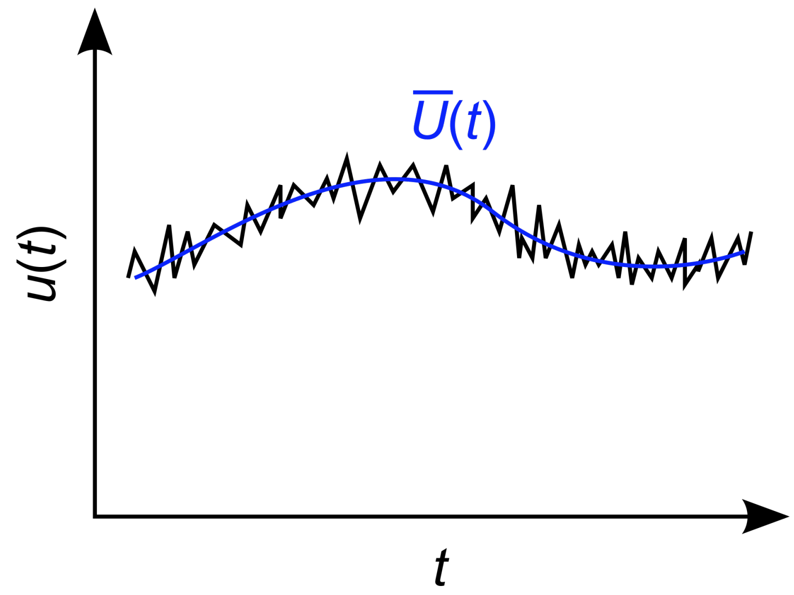
Only change viscosity!

Significant problem: difference of filtering (average) approaches

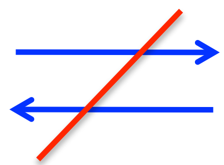
LES Filtering



Reynolds average



Spatial



Temporal

Inconsistency at the interface between LES and RANS regions

DES starts with RANS model

- Dimensional analysis of turbulent viscosity, velocity and length scale:

$$\left[\nu_{\mathrm{t}}\right] = \frac{\mathrm{m}^2}{\mathrm{s}}, \left[u\right] = \frac{\mathrm{m}}{\mathrm{s}}, \left[l\right] = \mathrm{m} \qquad \therefore \nu_{\mathrm{t}} = C_{\mu} u l$$

- Dimensions of turbulent kinetic energy:

$$\left[k\right] = \frac{\mathrm{m}^2}{\mathrm{s}^2}$$

- Therefore,

$$\nu_t = C_{\mu} \sqrt{k} l$$

Estimation of length scales for k-ε and k-ω

- Standar k-ε model:

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

$$\nu_t = C_\mu \sqrt{k}\, l = C_\mu \frac{k^2}{\varepsilon} \Rightarrow l = \frac{k^{3/2}}{\varepsilon}$$

- Wilcox's k-ω model

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

$$\nu_t = C_\mu \sqrt{k}\, l = \frac{k}{\omega} \Rightarrow l = \frac{\sqrt{k}}{C_\mu \omega}$$

Advantages of k-ω SST model with respect to DES

- Use with wall-functions on high-*Re* grids ($y^+ >> 12$)
- Use without wall-functions on low-*Re* grids ($y^+ \approx 1$)
- No additional low-*Re* term necessary
- Convection to DES model widely discussed in literature, see Ref. 3,4)
- (Re-use of blending functions for DDES, IDDES possible)

Governing equations of *k*-ω SST model

- Transport equation of *k*

$$\frac{\partial k}{\partial t} + \nabla\cdot\left(\mathbf{U}k\right) - \nabla\cdot\left(\left(\nu + \alpha_\kappa \nu_t\right)\nabla k\right)$$
$$= \min\left(\nu_t \nabla\mathbf{U}\cdot\left(2S - \frac{1}{3}\nabla\cdot\mathbf{UI}\right), C_1\beta^* k\omega\right) - \frac{2}{3}k\nabla\cdot\mathbf{U} - \beta^* k\omega\eta$$

- Transport equation of ω

$$\frac{\partial \omega}{\partial t} + \nabla\cdot\left(\mathbf{U}\omega\right) - \nabla\cdot\left(\left(\nu + \alpha_\kappa \nu_t\right)\nabla \omega\right)$$
$$= \gamma\nabla\mathbf{u}\cdot\left(2S - \frac{1}{3}\nabla\cdot\mathbf{UI}\right) - \frac{2}{3}\gamma\omega\nabla\cdot\mathbf{U} - \beta\omega\eta - \left(F_1 - 1\right)CD_{k\omega}$$

SST model is made "length-scale aware"

- Replace ω in dissipation term with length scale from Wilcox's *k*-ω SST model:

$$\frac{\partial k}{\partial t} + \nabla \cdot \left(\mathbf{U}k\right) - \nabla \cdot \left(\left(\nu + \alpha_\kappa \nu_t\right)\nabla k\right)$$

$$= \min\left(\nu_t \nabla \mathbf{U} \cdot \left(2S - \frac{1}{3}\nabla \cdot \mathbf{UI}\right), C_1 \beta^* k\omega\right)$$

$$-\frac{2}{3}k\nabla \cdot \mathbf{U} - \beta^* k\omega\eta$$

$$\omega = \frac{\sqrt{k}}{C_\mu l_{\mathrm{DES}}}$$

Length scale switching between RANS/LES modes

- Characteristic turbulent length scale:

$$l_{\mathrm{DES}} = \min\left(l_{\mathrm{RANS}}, l_{\mathrm{LES}}\right)$$

where

$$l_{\mathrm{RANS}} = \frac{\sqrt{k}}{C_\mu \omega}, \quad l_{LES} = C_{\mathrm{DES}} \min\left(\Delta_x, \Delta_y, \Delta_z\right)$$

For *k*-ω SST model:

$$C_{\mathrm{DES}} = 0.7$$

Correct DES case setup

- 3D
- Time-resolved
- Convection discretization scheme
- Appropriate grid parameters
- Cautious use of symmetry boundary condition

Convection discretization with "localBlended" scheme

- LES area: Central differencing
- RANS area: (High order) upwind
- Blending used for smooth transition

$$\phi_f = (1-\sigma)\phi_{f,CD} + \sigma\phi_{f,LUD}$$

- Blending function σ implemented as function object

  →can be plugged into any solver without code change

- Syntax in fvSchems:

```
div(phi,U)  Gauss localBlended linear linearUpwind grad(U);
```

5 stages for implementation of SST-DES model

- Stage 1: Copy files and rename classes
- Stage 2: Convert from RASModel to LESModel
- Stage 3: Include DES length scale
- Stage 4: Compile function object for localBlended scheme
- Stage 5: Create and run test case (pitzDaily)

Stage 1: Copy files and rename class

- Make local copy of kOmegaSST model from
  $FOAM_SRC/turbulenceModels/incompressible/RAS/kOmegaSST to
  $FOAM_RUN/../lib
- Rename directory kOmegaSST/ to SST-DES/
- In SST-DES/, rename kOmegaSST.H and kOmegaSST.C to SSTDES.H and SSTDES.C
- Edit SSTDES.H and SSTDES.C and with search&replace change all "kOmegaSST" to "SSTDES"
- Make copy of $FOAM_SRC/turbulenceModels/incompressible/RAS/Make directory to SST-DES/
- Edit Make/files: remove all *.C files, enter SSTDES.C,
  change lib-name to LIB = $(FOAM_USER_LIBBIN)/libDESModels
- Edit Make/option: add line
  -I$(LIB_SRC)/turbulenceModels/incompressible/RAS/lnInclude \
- Compile with "wmake libso"

Stage 2a: Convert from RASModel to LESModel

- Using search&replace in `SSTDES.H` and `SSTDES.C`, change all:
  - '`nut`' to '`nuSgs`'
  - '`RAS`' to '`LES`'
  - '`R()`' to '`B()`'
- In `SSTDES.H`:
  - In line 137 (after `wallDist y_;`), add: `dimensionedScalar omegaMin_;`

    1.6-ext: `omega0`
  - 1.6-ext: Change constructor declaration:

```
SSTDES::SSTDES
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& lamTransportModel
)
```

```
SSTDES::SSTDES
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& transport,
    const word& modelName = typeName
)
```

Stage 2b: Convert from RASModel to LESModel

- In SSTDES.C:
  - Replace line 29 (`#include “backwardsComp…`) with
    `#include “wallDist.H”`
  - 1.6-ext `Change constructor:`

```
SSTDES::SSTDES
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& lamTransportModel
)
:
LESModel(typeName,U,phi,lamTransportModel),
```

→

```
SSTDES::SSTDES
(
    const volVectorField& U,
    const surfaceScalarField& phi,
    transportModel& transport,
    const word& modelName
)
:
LESModel(modelName,U,phi,transport),
```

Stage 2c: Convert from RASModel to LESModel

- In SSTDES.C:
  - In line 244 (after `y_(mesh_),`), add:

    `omegaMin_("omegaMin", dimless/dimTime, SMALL),`

    1.6-ext omega0
  - In line 257, 269, 281;

    Change '`autoCreate*("*", mesh_)`' to '`mesh_`'
  - Remove lines 402-405: `if(!turbulence_)` …
  - In `IOojbect` of k_, omega_, nuSgs:

```
IOobject
    (
        "k",
        runTime.timeName(),
        mesh_,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
```

→

```
IOobject
    (
        "k",
        runTime.timeName(),
        mesh_,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
```

1.6-ext
  - In '`//Re-calculate viscosity`' section, remove

    `nuSgs_ = min(nuSgs_,nuRatio()*nu());`

Stage 2d: Convert from RASModel to LESModel

- In Make/options:
  - Change

    `-I$(LIB_SRC)/turbulenceModels/incompressible/RAS/lnInclude/` to

    `-I$(LIB_SRC)/turbulenceModels/incompressible/LES/lnInclude`
  - Add to EXE_INC:

    ```
    -I$(LIB_SRC)/turbulenceModels/LES/LESdeltas/lnInclude \
    -I$(LIB_SRC)/turbulenceModels/LES/LESfilters/lnInclude \
    ```
  - Add to LIB_LIBS:

    ```
    -lLESdeltas \
    -lLESfilters \
    ```
- Compile

Stage 3a: Include DES length scale

- In `SSTDES.H`:
  - In line 138 (before `wallDist y_;`), add:

```
dimensionedScalar CDES_;
```

- In `SSTDES.C`:
  - In constructor initialization list (line 245), insert;

```
CDES_
(
    dimensioned<scalar>::lookupOrAddToDict
    (
        “CDES”,
        coeffDict_,
        0.7
    )
 ),
```

Stage 3b: Include DES length scale

- In `SSTDES.C`:
  - In `SSTDES::correct()` (line 411, before `volScalarField S2 =` …)

```
//calculate DES length scale
volScalarField lRANS = sqrt(k_)/(betaStar_*omega_);
volScalarField lLES = CDES_*delta();
volScalarField lDES = Foam::min(lLES,lRANS);
```

- Modify k transport equation, change line 457:

from: `- fvm::Sp(betaStar_*omega_,k_)`

to: `- fvm::Sp(sqrt(k_)/lDES,k_)`

- Compile

Stage 4: Compile function object for localBlended scheme

- Download `functionObject.tgz`
- (Version in OFWS8 `training_materials/` is for OF-1.6-ext)
- Unpack
- Compile with "`wmake libso`"
- And you're ready to go …

Stage 5a: Create and run test case (pitzDaily)

- Copy test case from tutorials section:
  - Change to run-directory, just type "run"
  - Copy pitzDaily test case from `$FOAM_TUTORIALS/incompressible/pisoFoam/les/pitzDaily` to $FOAM_RUN
- Add dynamic loading of DES library to `controlDict`, edit `system/controlDict`:
  - Insert line: libs ("libDESModels.so");
  - Insert function object for `localBlended` scheme:

```
blendingParameter
{
    type UBlendedField;
    functionObjectLibs ("libDESFunctionObjects.so");
    outputControl outputTime;
    URef 10;
    lRef 0.0025;
    enabled true;
}
```

Stage 5b: Create and run test case (pitzDaily)

- Enable SSTDES model, edit `constant/LESModels`:
  - In line 18, change 'oneEqEddy' to 'SSTDES'
  - Add:

```
DESCoeffs
{
    CDES 0.7;
}
```

- Set convection discretization scheme, edit system/fvSchemes:
  - Change line 33 to

```
div(phi,U)  Gauss localBlended linear linearUpwind grad(U);
```

  - If you use TVD scheme,

```
div(phi,U)  Gauss limited linear 1;
```

Stage 5c: Create and run test case (pitzDaily)

- Run test case:
  - `blockMesh`
  - `pisoFoam`

Further steps (homework)

- Introduce blended C_{DES} constant, see Ref. 3)
- Expand SSTDES to SST-DDES
- Expand SST-IDDES
- Run proper test case:
  - 3D
  - appropriate mesh resolution, see Ref. 2)
  - IDDES: Test WMLES and DES capability with either RANS-type or LES-type boundary conditions

References

1) P. R. Spalart et al.: "Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach", 1st ASOSR CONFERENCE on DNS/LES, 1997
2) P. R. Spalar: "Young-Person's Guide to Detached-Eddy Simulation Grids", NASA CR-2001-211032
3) M. S. Gritskevich et al.: "Development of DDES and IDDES Formulations for the k-w Shear Stress Transport Model", Flow Turbulence and Combustion, 2011
4) F. R. Menter et al.: "Ten Years of Industrial Experience with the SST Turbulence Model", Turbulence, Heat and Mass Transfer, 2003
5) C. Mockett: "A Comprehensive Study of Detached-Eddy Simulation", PhD thesis, 2009.
6) W. Hasse et al.: "DESider – A European Effort on Hybrid RANS-LES Modeling", Springer, 2009