Domain decomposition approach for near-wall turbulence modeling

Citation for published version (APA):
Utyuzhnikov, S. (2016). Domain decomposition approach for near-wall turbulence modeling. In *Domain decomposition approach for near-wall turbulence modeling* (Proceedings of ECCOMAS 2016 Congress). Springer Nature.

Published in:
Domain decomposition approach for near-wall turbulence modeling

Domain Decomposition Approach for Near-wall Turbulence Modeling

Sergei Utyuzhnikov

University of Manchester

Moscow Institute of Physics & Technology

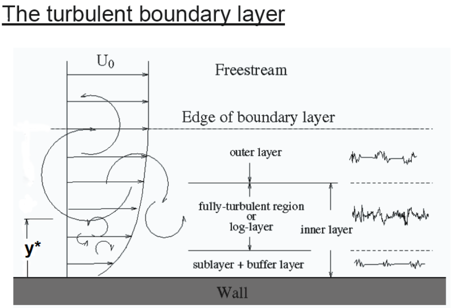
s.utyuzhnikov@manchester.ac.uk

ECCOMAS, Crete, Greece

5-10 June, 2016

1. Introduction to Near-wall Turbulence
2. Near-wall Domain Decomposition
3. Test cases

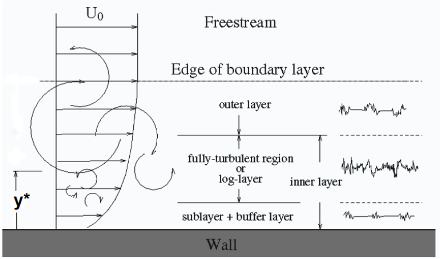
Near-wall Turbulence



©Fluent

Near-wall Turbulence

The turbulent boundary layer

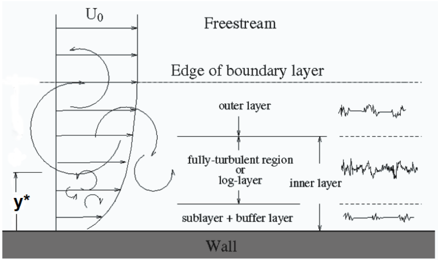


©Fluent

- near-wall sublayer significantly affects mean flow

Near-wall Turbulence

The turbulent boundary layer

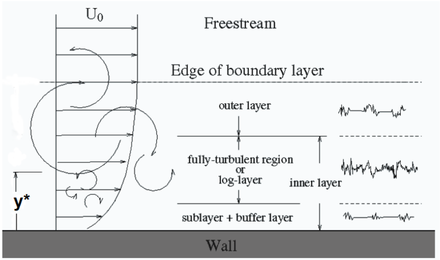


©Fluent

- near-wall sublayer significantly affects mean flow
- resolution of near-wall area requires up to 90% of CPU time

Near-wall Turbulence

The turbulent boundary layer

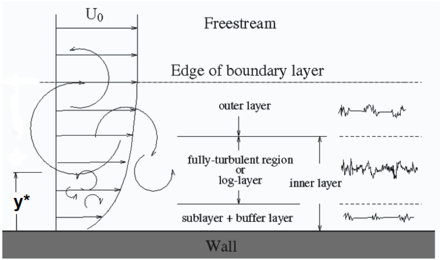


ⒸFluent

- near-wall sublayer significantly affects mean flow
- resolution of near-wall area requires up to 90% of CPU time
- it is a multiscale problem

Near-wall Turbulence

The turbulent boundary layer

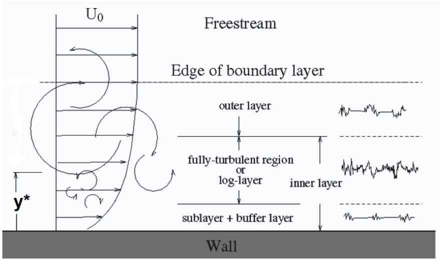


ⒸFluent

- near-wall sublayer significantly affects mean flow
- resolution of near-wall area requires up to 90% of CPU time
- it is a multiscale problem
- domain decomposition should be efficient

Near-wall Turbulence

The turbulent boundary layer



©Fluent

- near-wall sublayer significantly affects mean flow
- resolution of near-wall area requires up to 90% of CPU time
- it is a multiscale problem
- domain decomposition should be efficient
- standard approach: domain decomposition based on wall functions + HRN

Near-wall Turbulence

- Low Reynolds Number models (LRN):
  governing equations include all terms of the Reynolds Averaged Navier-Stokes Equations (RANS)

Near-wall Turbulence

- Low Reynolds Number models (LRN):
  governing equations include all terms of the Reynolds Averaged Navier-Stokes Equations (RANS)

- High Reynolds Number models (HRN):
  governing equations ignore all near-wall damping terms

Near-wall Turbulence

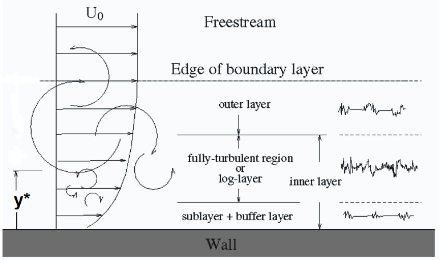
- Low Reynolds Number models (LRN):
  governing equations include all terms of the Reynolds Averaged Navier-Stokes Equations (RANS)

- High Reynolds Number models (HRN):
  governing equations ignore all near-wall damping terms

- Wall functions:
  Dirichlet boundary conditions are set for HRN at the nearest to the wall cell

Near-wall Turbulence

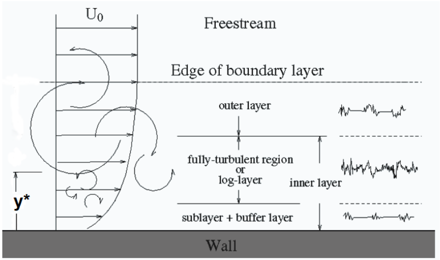
The turbulent boundary layer



©Fluent

Near-wall Turbulence

The turbulent boundary layer

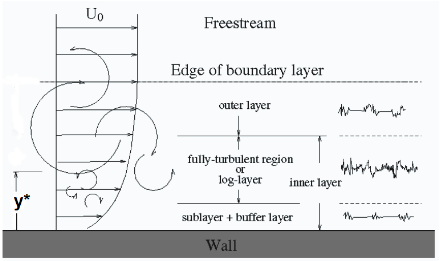


©Fluent

- non-overlapping domain decomposition:

Near-wall Turbulence

The turbulent boundary layer

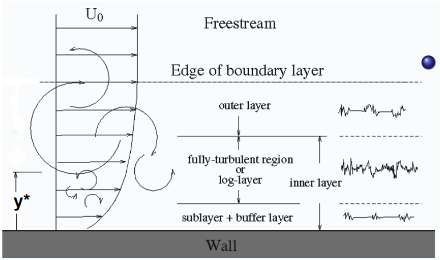


©Fluent

- non-overlapping domain decomposition:
- the b.c can be transferred from the wall to $y = y^*$

Near-wall Turbulence

The turbulent boundary layer

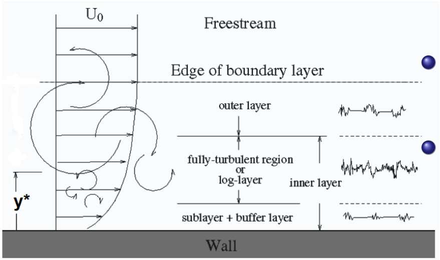


©Fluent

- non-overlapping domain decomposition:
- the b.c can be transferred from the wall to $y = y^*$
- the interface near-wall b.c. (INBC) is nonlocal

Near-wall Turbulence

The turbulent boundary layer



©Fluent

- non-overlapping domain decomposition:
- the b.c can be transferred from the wall to $y = y^*$
- the interface near-wall b.c. (INBC) is nonlocal
- general formulation of b.c.

$$\frac{\partial u}{\partial n}_{|y^*} = u(y^*)S_{y^*}(1) + f_y^*,$$

S_{y^*} is the Steklov-Poincaré operator

Utyuzhnikov, 2009

Near-wall domain decomposition. Sketch (linear problem)

- Original BVP (possible statement)

Near-wall domain decomposition. Sketch (linear problem)

- Original BVP (possible statement)

- Transfer of b.c. to an interface boundary

Near-wall domain decomposition. Sketch (linear problem)

- Original BVP (possible statement)

- Transfer of b.c. to an interface boundary

- R.b.c. does not depend on the outer region!

Near-wall domain decomposition. Sketch (cont.)

- Solution of BVP in the outer region

Near-wall domain decomposition. Sketch (cont.)

- Solution of BVP in the outer region

- Solution of BVP in the inner region (if needed)

The near-wall domain decomposition (NDD)

Consider equation

$$(\mu u_y)_y = R(y)$$

in $D = [0 \ y_e]$ with boundary condition at $y = 0$:

$$u(0) = 0.$$

INBC is set at y^*

$$0 < y^* < y_e, \ \ D^- := [0 \ y^*].$$

Interface Near-wall Boundary Condition (Utyuzhnikov, 2005)

Consider

$$(\mu u_y)_y = R(y),$$
$$u(0) = 0.$$

INBC at $y = y^*$:

$$u(y^*) = u'(y^*) \int_0^{y^*} \frac{\mu(y^*)}{\mu(y)} dy - \frac{1}{\mu(y^*) y^*} \int_0^{y^*} \left(\frac{\mu(y^*)}{\mu(y)} \int_y^{y^*} R(y') dy' \right) dy.$$

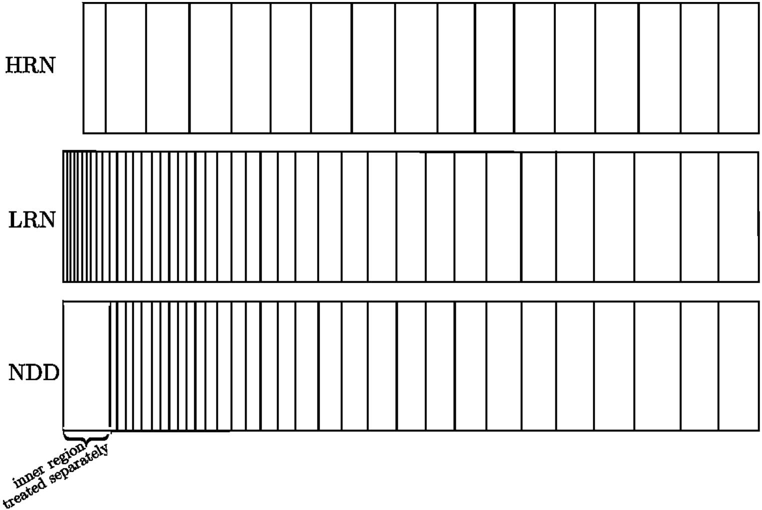
1D Equation. Wall flux

The wall flux can be obtained without resolution of the inner layer

$$C_f = (\mu u_y)_{|y=y^*} - \int_0^{y^*} R(y)dy,$$

where $C_f = (\mu u_y)_{|y=0}$.

Near-wall non-overlapping domain decomposition



Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case

Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case
- The IBCs are mesh independent

Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case
- The IBCs are mesh independent
- The IBCs completely replace the inner region for the outer region

Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case
- The IBCs are mesh independent
- The IBCs completely replace the inner region for the outer region
- Treatment of all functions is basically the same

Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case
- The IBCs are mesh independent
- The IBCs completely replace the inner region for the outer region
- Treatment of all functions is basically the same
- Can be used for both LRN and HRN models

Interface boundary conditions (IBCs)

- The interface boundary condition is exact even in non-linear case
- The IBCs are mesh independent
- The IBCs completely replace the inner region for the outer region
- Treatment of all functions is basically the same
- Can be used for both LRN and HRN models

Advantages of NDD

- Interface boundary y^* provides a clear trade-off between accuracy and time consumption

Advantages of NDD

- Interface boundary y^* provides a clear trade-off between accuracy and time consumption
- The LRN solution is recovered as $y^* \to 0$

Advantages of NDD

- Interface boundary y^* provides a clear trade-off between accuracy and time consumption
- The LRN solution is recovered as $y^* \to 0$
- Meshes in the inner and outer regions completely independent

Advantages of NDD

- Interface boundary y^* provides a clear trade-off between accuracy and time consumption
- The LRN solution is recovered as $y^* \to 0$
- Meshes in the inner and outer regions completely independent
- No free parameters

Advantages of NDD

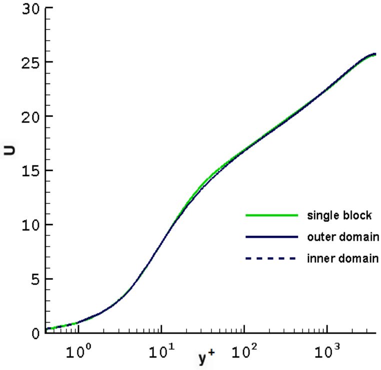
- Interface boundary y^* provides a clear trade-off between accuracy and time consumption
- The LRN solution is recovered as $y^* \to 0$
- Meshes in the inner and outer regions completely independent
- No free parameters
- Implementation into industrial codes easy since IBCs computed with external subroutine

Advantages of NDD

- Interface boundary y^* provides a clear trade-off between accuracy and time consumption
- The LRN solution is recovered as $y^* \to 0$
- Meshes in the inner and outer regions completely independent
- No free parameters
- Implementation into industrial codes easy since IBCs computed with external subroutine
- Robin boundary condition is robust, converges fast since both Φ and $\partial_y \Phi$ taken at the same iteration:

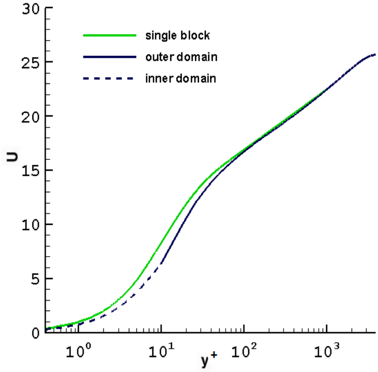
$$\Phi\big|_{y^*} = f_1 \left.\frac{\partial \Phi}{\partial y}\right|_{y^*} + f_2$$

Low-Re Velocity Profile. $Re = 3950$



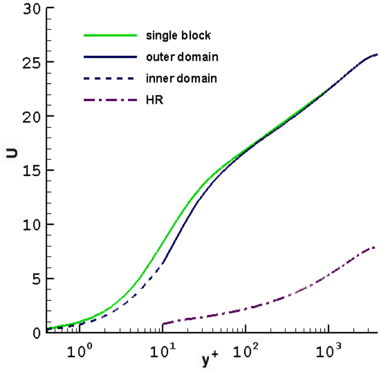
$$y^{+*} = u_\tau y^*/\nu = 1$$

Low-Re Velocity Profile. $Re = 3950$



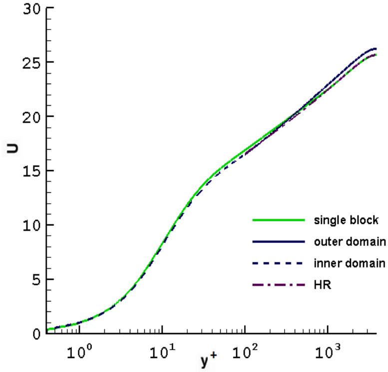
$$y^{+*} = u_\tau y^*/\nu = 10$$

Low-Re Velocity Profile. $Re = 3950$



$$y^{+*} = u_\tau y^*/\nu = 10$$

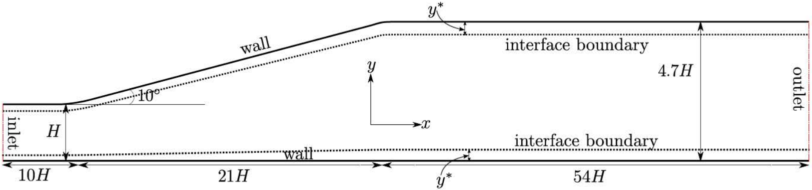
Low-Re Velocity Profile. $Re = 3950$



$$y^{+*} = u_\tau y^*/\nu = 100$$

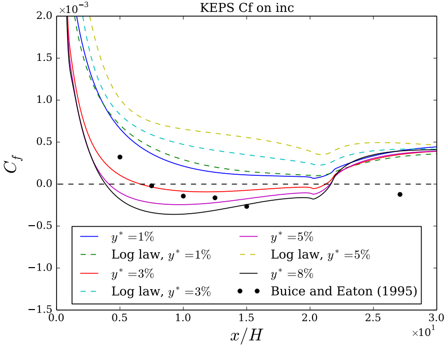
Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$

- Compute inlet conditions with a separate LRN calculation



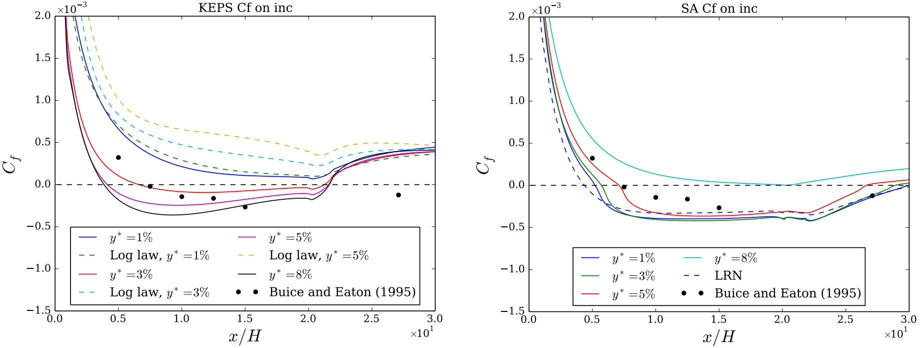
- Cut boundary layers off both walls
- Compute C_f along the inclined wall (Jones, Utyuzhnikov, 2015)

Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. Different y^*/H



$k - \epsilon$ model (HRN)

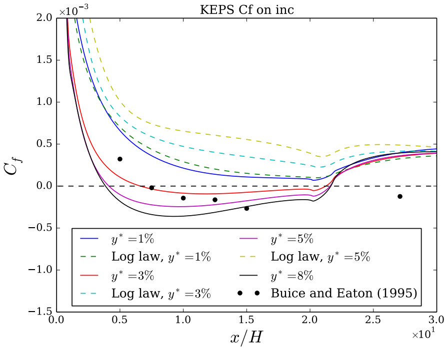
Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. Different y^*/H



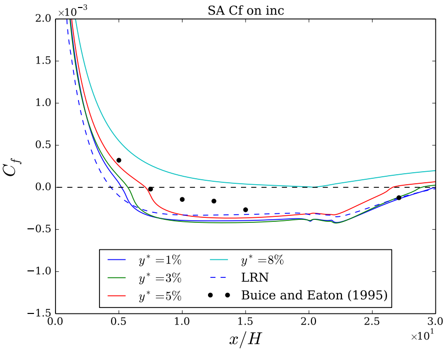
$k-\epsilon$ model (HRN)

Spalart-Almaras model

Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. Different y^*/H



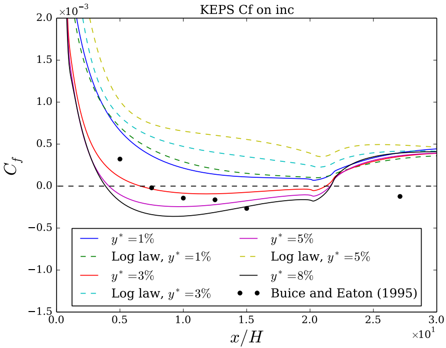
$k-\epsilon$ model (HRN)



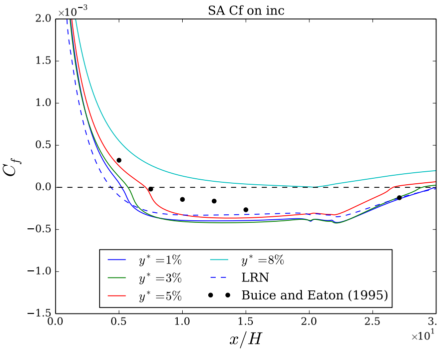
Spalart-Almaras model

- Clear convergence to LRN solution as $y^* \to 0$ for SA model

Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. Different y^*/H



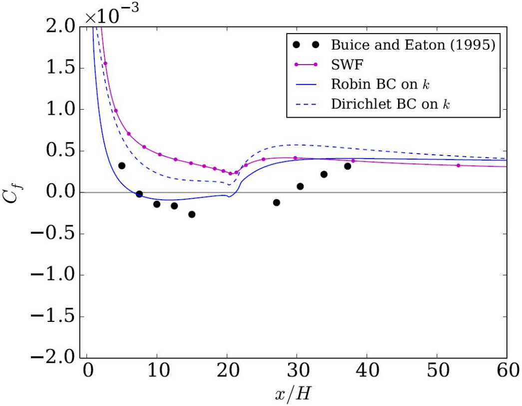
$k - \epsilon$ model (HRN)



Spalart-Almaras model

- Clear convergence to LRN solution as $y^* \to 0$ for SA model
- DD produces recirculation region with both models

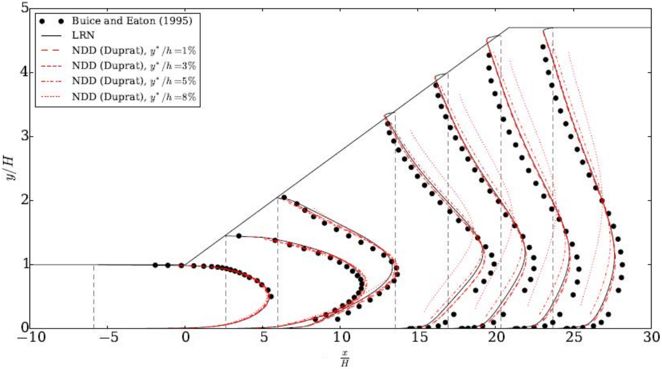
Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. $y^*/h = 0.03$



$k - \epsilon$ model (HRN)

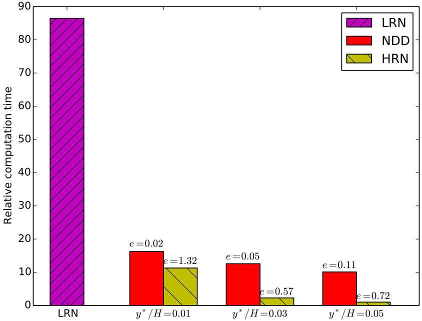
- Only Robin boundary condition for k predicts the separation

Turbulent asymmetric diffuser, $Re = 1.8 * 10^4$. Velocity profile



Spalart-Almaras model. $10U/U_b$

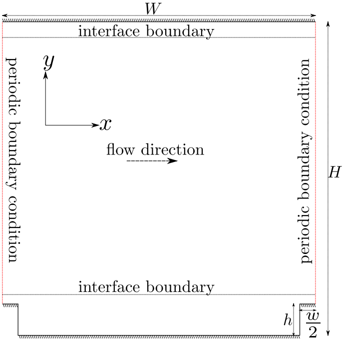
Comparison of simulation time for *SST* model



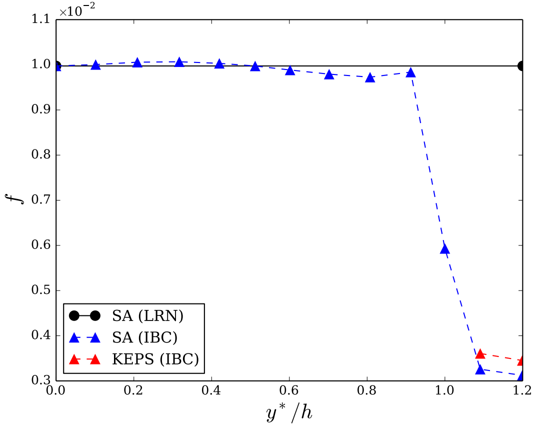
- Max error compared to LRN solution is shown

$$e = \frac{\max \left| C_f^{\mathrm{LRN}}(x) - C_f^{\mathrm{NDD\ /\ HRN}}(x) \right|}{C_{f0}}$$

Plane ribbed channel flow (2D). SA model (Jones, Utyuzhnikov, 2015)



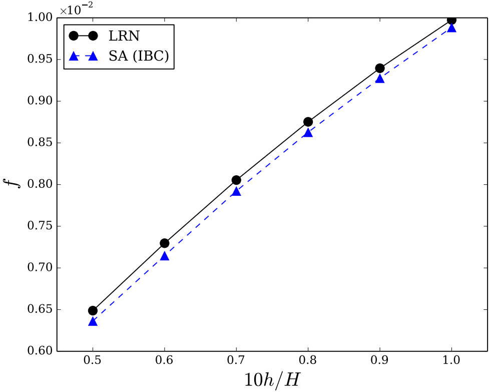
- Periodic flow, both walls heated with constant heat flux
- $h/H = 0.1$
- Remove same amount from upper and lower walls
- Interesting limit is $y^*/h = 1$
- For $y^*/h < 1$, some of rib tops is included in mesh



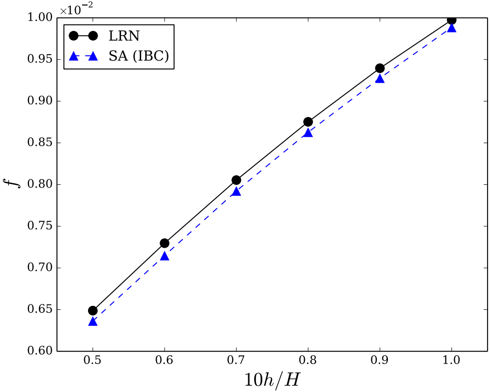
Friction factor for different y^*, the rib height is fixed

- For *SA*, error is less than 2.5% until $y^*/h = 0.9$

Plane ribbed channel flow (2D). f for different h



Plane ribbed channel flow (2D). f for different h



Varying the rib height is trivial since only one parameter needs adjusting:

$$h = h_{resolved} + y^*$$

Conclusions

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows
- NDD represents a clear trade-off between accuracy and computational resources that is easily managed

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows
- NDD represents a clear trade-off between accuracy and computational resources that is easily managed
- Heat fluxes and friction factor show little sensitivity to y^*

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows
- NDD represents a clear trade-off between accuracy and computational resources that is easily managed
- Heat fluxes and friction factor show little sensitivity to y^*
- For ribbed channels a large portion of a rib can be effectively removed

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows
- NDD represents a clear trade-off between accuracy and computational resources that is easily managed
- Heat fluxes and friction factor show little sensitivity to y^*
- For ribbed channels a large portion of a rib can be effectively removed
- Changing the rib height is simple and the results are reasonably accurate

Conclusions

- NDD proved to be an efficient and quite accurate method for simulating near-wall regions in turbulent flows
- NDD represents a clear trade-off between accuracy and computational resources that is easily managed
- Heat fluxes and friction factor show little sensitivity to y^*
- For ribbed channels a large portion of a rib can be effectively removed
- Changing the rib height is simple and the results are reasonably accurate
- NDD can be especially efficient in optimal engineering design

Literature

1. Utyuzhnikov, S.V., "Generalized wall-functions and their application for simulation of turbulent flows", Int. J. Numerical Methods in Fluids, 2005, 47.

2. Utyuzhnikov, S.V., "Some new approaches to building and implementation of wall-functions for modeling of near-wall turbulent flows", J. Computers & Fluids, 2005, 34.

3. Utyuzhnikov, S.V., "Robin-type wall functions and their numerical implementation", J. Appl. Numer. Math., 2008, 58.

4. Utyuzhnikov, S.V., "Domain decomposition for near-wall turbulent flows", J. Computers & Fluids, 2009, 38.

5. Utyuzhnikov, S.V., "Generalized Calderon-Ryaben'kii's potentials", IMA J. of Appl. Math., 2009, 74.

Literature (cont.)

6. Utyuzhnikov, S.V., "Interface boundary conditions in near-wall turbulence modeling", J. Computers & Fluids, 2012, 68.

7. Aleksin, V.A., Utyuzhnikov, S.V., "Implementation of near-wall boundary conditions for modelling boundary layers with free-stream turbulence", Appl. Math. Modelling, 2014, 38 (14).

8. Utyuzhnikov, S.V., "Towards development of unsteady near-wall interface boundary conditions for turbulence modelling", Comp. Phys. Comm., 2014, 185 (11).

9. Jones, A., Utyuzhnikov, S., "Application of a near-wall domain decomposition method to turbulent flows with heat transfer", Computers & Fluids, 2015, 119.

10. Jones, A., Utyuzhnikov, S., "A near-wall domain decomposition approach in application to turbulent flow in a diffuser", Appl. Math. Modelling, 2016, 40 (1).