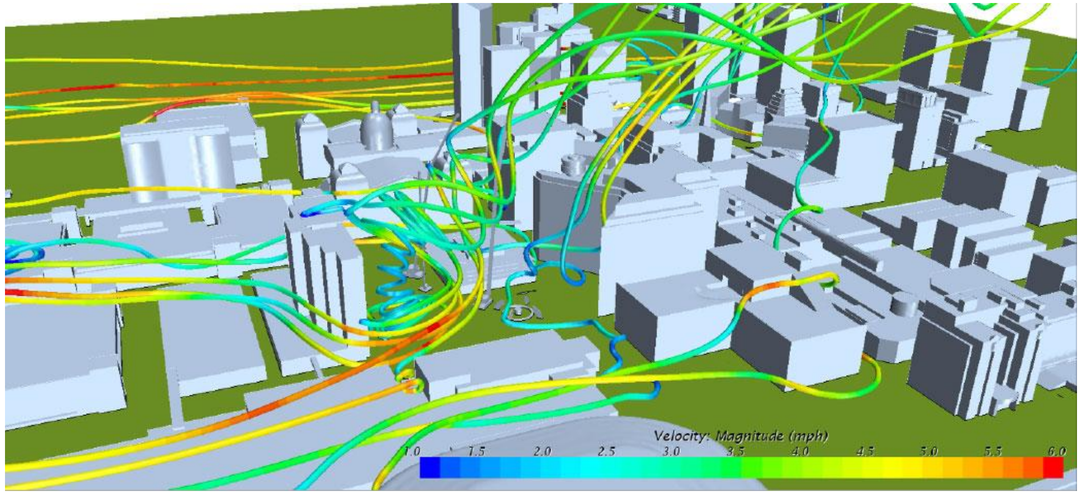
Atmospheric boundary layer simulations – wall boundary condition

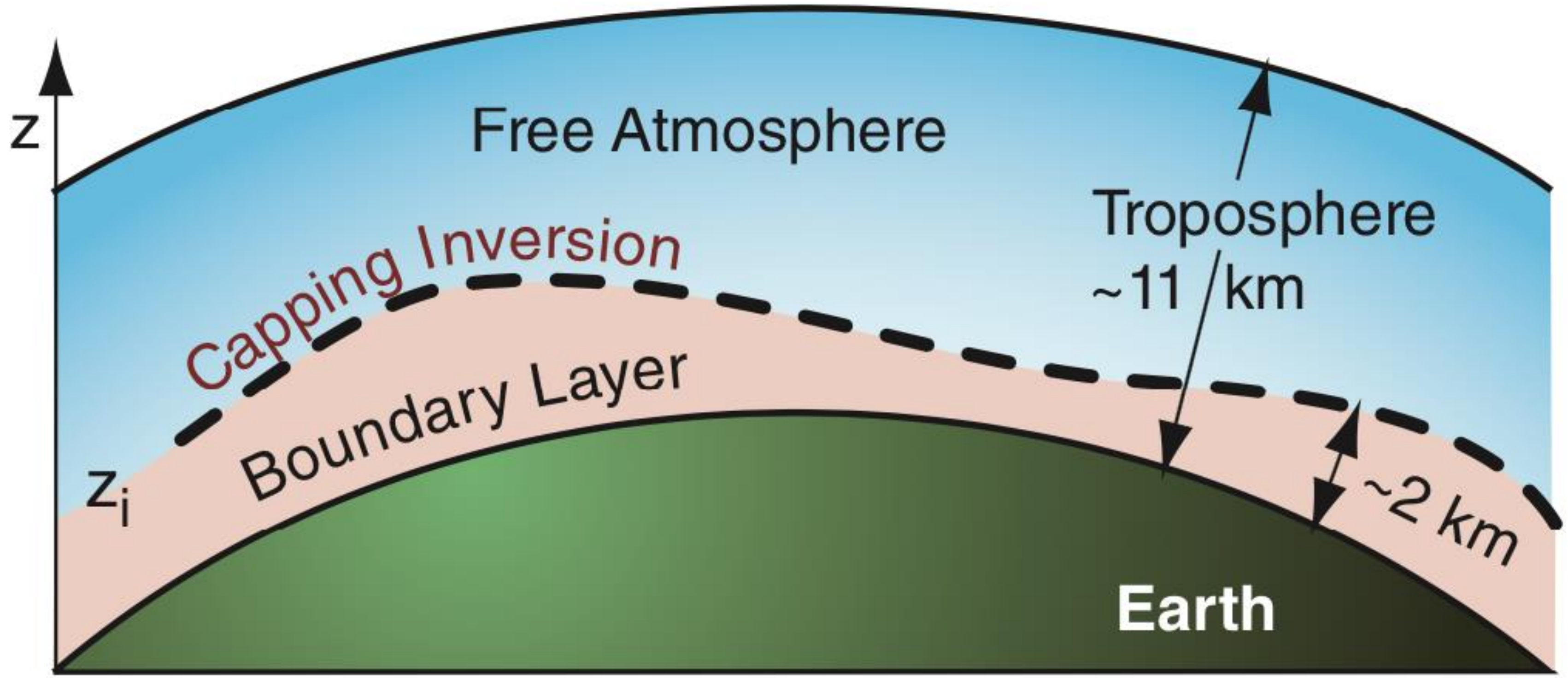
Mads Baungaard

September 9, 2024

Relevance



Atmospheric boundary layer (ABL)

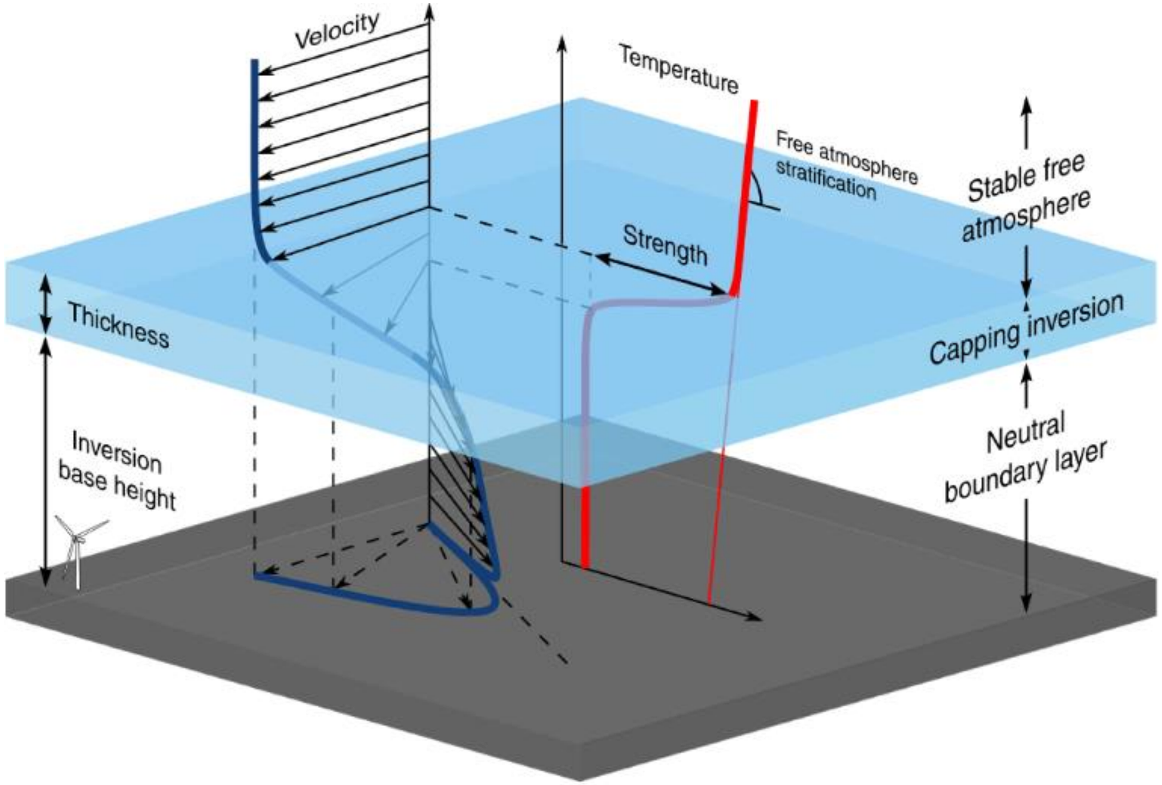


- $\nu \sim 10^{-5}\ \mathrm{m^2\ s^{-1}}$
- $\mathcal{L} \sim 10^{3}\ \mathrm{m}$
- $\mathcal{U} \sim 10^{1}\ \mathrm{m\ s^{-1}}$

→ **High-Re flow**

Horizontally homogeneous ABL

- Flat terrain + uniform surface + uniform atmospheric forcing
  → Mean flow and turbulence statistics are independent of *x* and *y*.



$$\frac{\partial U}{\partial t} = f_{\mathrm{c}}(V - V_{\mathrm{g}}) - \frac{\partial \overline{uw}}{\partial z},$$

$$\frac{\partial V}{\partial t} = f_{\mathrm{c}}(U_{\mathrm{g}} - U) - \frac{\partial \overline{vw}}{\partial z},$$

$$\frac{\partial \Theta}{\partial t} = -\frac{\partial \overline{w\theta}}{\partial z},$$

Mathematical formulation

Solve

$$\frac{\partial \mathbf{q}}{\partial t} = f\left(\mathbf{q}, \frac{\partial \mathbf{q}}{\partial z}, \frac{\partial^2 \mathbf{q}}{\partial z^2}, \ldots\right)$$

subject to

$$\mathbf{q}(z, t = 0) = \mathbf{q}_{\text{initial}}$$

$$b_{\text{top}}\left(\mathbf{q}, \frac{\partial \mathbf{q}}{\partial z}, \ldots\right)\bigg|_{(z=z_{\text{top}}, t)} = \mathbf{0}$$

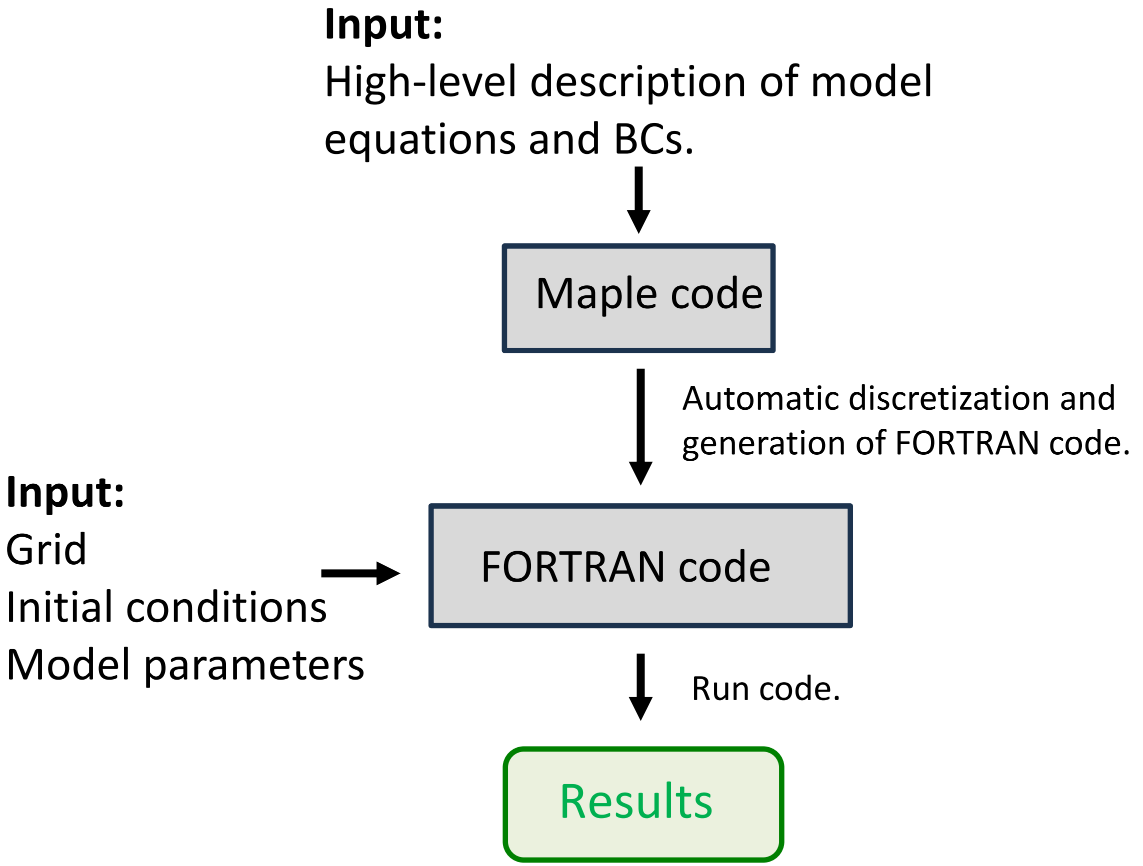
$$b_{\text{bot}}\left(\mathbf{q}, \frac{\partial \mathbf{q}}{\partial z}, \ldots\right)\bigg|_{(z=z_{\text{bot}}, t)} = \mathbf{0}$$

ABL with K- ε model:

$$\mathbf{q} = \begin{pmatrix} U \\ V \\ \Theta \\ K \\ \varepsilon \end{pmatrix}$$

Maple/FORTRAN 1d-solver

- KTH PhDs, Lazeroms and Zeli, used a Maple/FORTRAN 1d-solver for their ABL simulations (written by S. Wallin).




```
#
# The equation
#
> eq_U := Diff(U,t) = Diff(nut(y)*Diff(U,y),y) + Px;
  eq_k := Diff(k,t) = (1-fw(y))*(Pk(y) - Jk(y)*k +
      Diff((nut(y)/sigk)*Diff(k,y),y) + Gk_ex - Gk_im*k)
  + fw(y)*(Pklog(y) - Jk(y)*k);
  eq_eps := Diff(eps,t) = (1-fw(y))*(Peps(y) - Jeps(y)*
  eps +
      Diff((nut(y)/sigeps)*Diff(eps,y),y) + Geps_ex -
  Geps_im*eps) + fw(y)*cfl(y)*(epslog(y) - eps);
  eq_Theta := Diff(Theta,t) = Diff(kappat(y)*Diff(Theta,
  y),y);
  eq_kth := Diff(kth,t) = 0;#Pkth - Jkth*kth + Diff(nut
  (y)/sigkth*Diff(kth,y),y);
  eq := [eq_U, eq_k, eq_eps, eq_Theta, eq_kth];
```

New Python 1d-solver

- In the Autumn 2021, a similar code was written in Python (by S. Wallin).

Input:
High-level description of model equations and BCs.
Grid
Initial conditions
Model parameters

↓

Python code

↓ Run code.

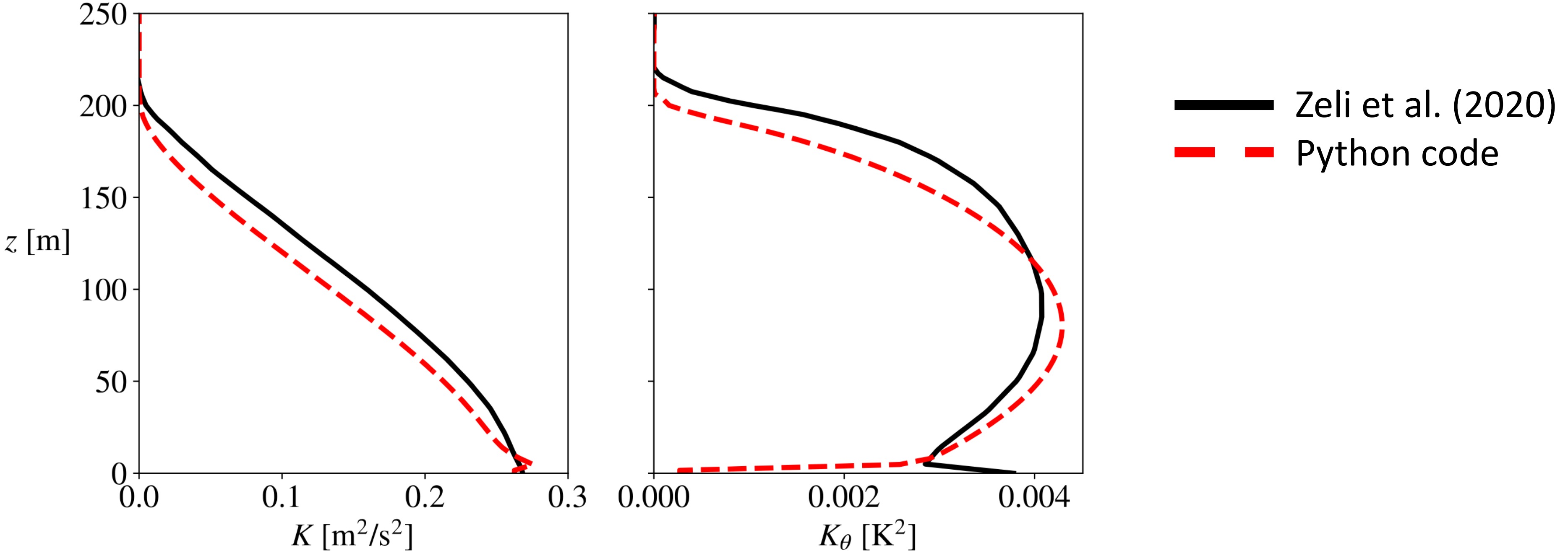
Results

```
eqd = {'U':   'Diff(U,t) - Diff(nut*Diff(U,y),y) = Px',
       'K':   'Diff(K,t) - Diff(nut/sigk*Diff(K,y),y) + fe*K = Pk',
       'eps': 'Diff(eps,t) - Diff(nut/sige*Diff(eps,y),y) + Ce2*fe*eps = Ce1*Pk*fe',
      }
BCd = {'U':   ['U = 0.0', 'Diff(U,y) = 0'],
       'K':   ['K = utau**2/sqrt(Cmu)', 'Diff(K,y) = 0'],
       'eps': ['eps = utau**3/(kappa*y0)', 'Diff(eps,y) = 0']
      }
eqt = {'utau':   'utau = kappa*sqrt(U_[1]**2.0)/(ln(yc_[1]/y0+1.0))'
      }
fyd = {'sqrt': sqrt,
       'ln': ln,
```

Python code verification

Can the Python code reproduce ABL results from Lazeroms (2015) and Zeli (2021)?

Some preliminary results for the GABLS 1 case:



Wall boundary condition for ABLs

Example: neutral atmospheric channel flow

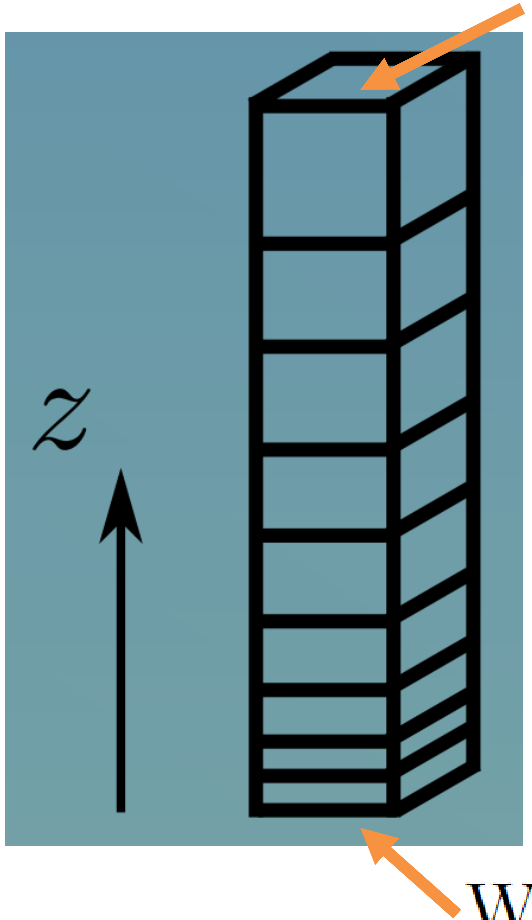
Assume

- No Coriolis
- Neutral ABL (→ no temperature)
- Regular K-ε model

$$\frac{\partial U}{\partial t} = -f_c V_g - \frac{\partial \overline{uw}}{\partial z},$$
$$\frac{\partial K}{\partial t} = \mathcal{P} - \varepsilon + \mathcal{D}_K,$$
$$\frac{\partial \varepsilon}{\partial t} = C_{\varepsilon 1}\frac{\varepsilon}{K}\mathcal{P} - C_{\varepsilon 2}\frac{\varepsilon^2}{K} + \mathcal{D}_\varepsilon$$

$$\overline{uw} = -\nu_t \frac{\partial U}{\partial z}$$
$$\nu_t = f_m \frac{K^2}{\varepsilon}$$

Top BC: $\frac{\partial \cdot}{\partial z} = 0$

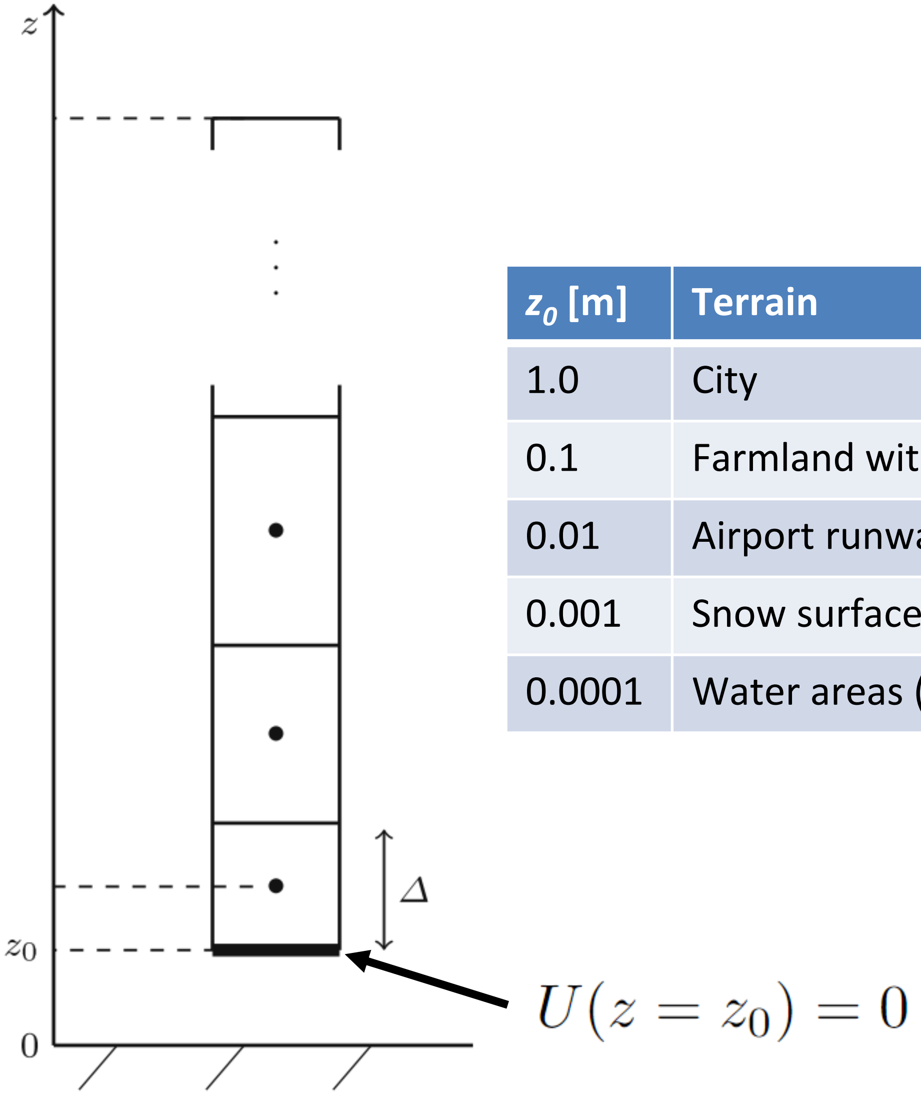


Wall BC: ??

Wall boundary condition (BC) for ABLs

- Model wall BC with neutral log-law.

- $U(z) = \frac{u_*}{\kappa} \ln\left(\frac{z}{z_0}\right)$
- $K(z) = \frac{u_*^2}{\sqrt{f_m}}$
- $\varepsilon(z) = \frac{u_*^3}{\kappa z}$

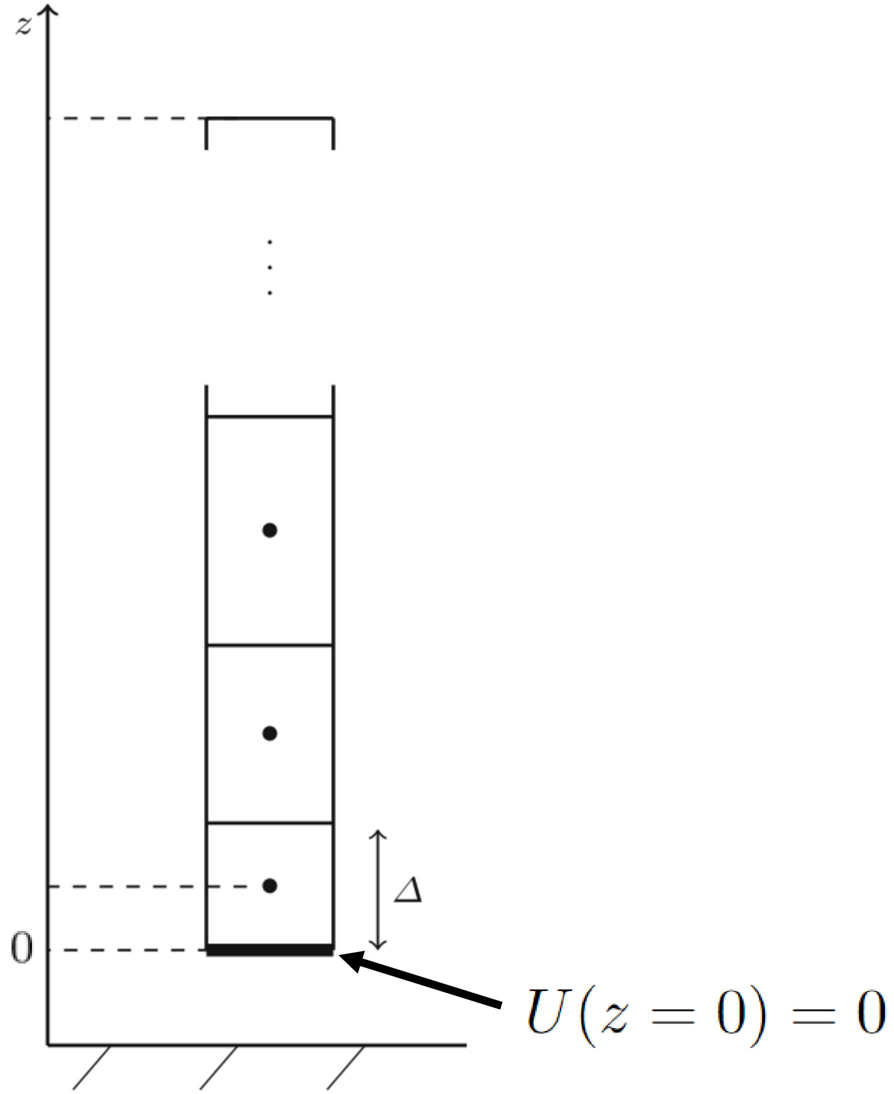


z_0 [m]	Terrain
1.0	City
0.1	Farmland with closed appearance
0.01	Airport runway areas
0.001	Snow surfaces
0.0001	Water areas (lakes, fjords, open sea)

Coordinate transformation

- Numerically convenient to re-define $z \rightarrow z + z_0$

  - $U(z) = \frac{u_*}{\kappa} \ln\left(\frac{z+z_0}{z_0}\right)$
  - $K(z) = \frac{u_*^2}{\sqrt{f_m}}$
  - $\varepsilon(z) = \frac{u_*^3}{\kappa(z+z_0)}$

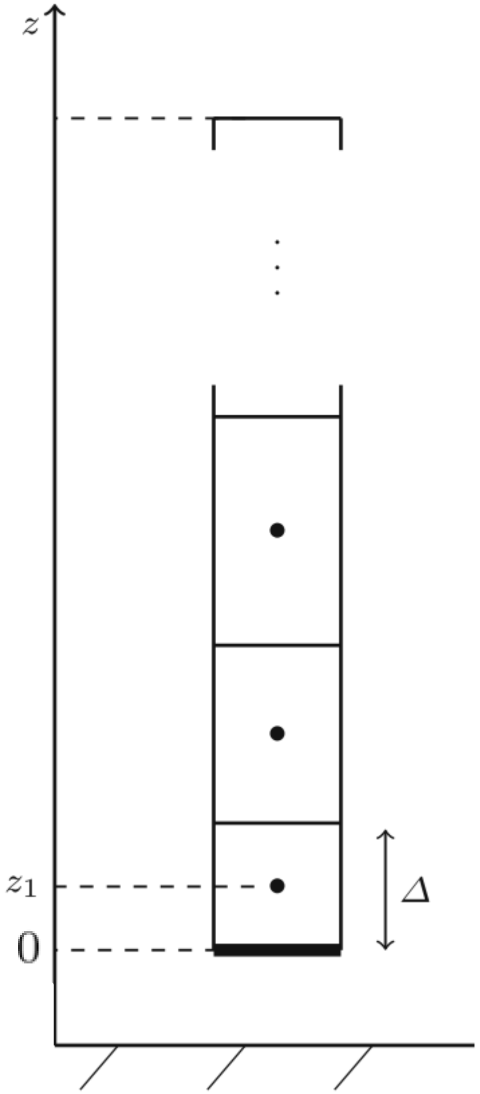


A naïve BC implementation 1/2

- Set BC:
  - $U(z=0) = 0$
  - $K(z=0) = \dfrac{u_*^2}{\sqrt{f_m}}$
  - $\varepsilon(z=0) = \dfrac{u_*^3}{\kappa z_0}$

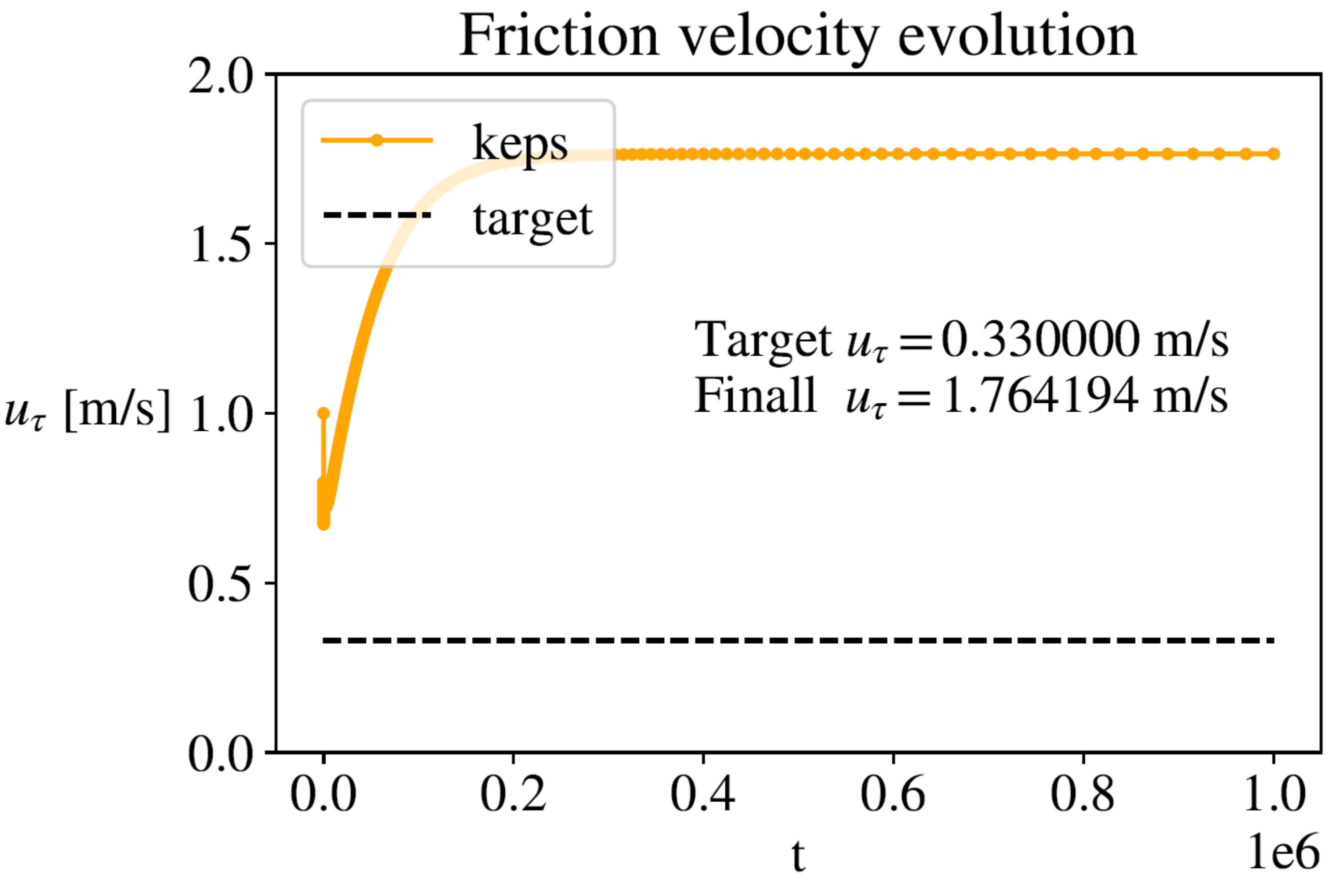
- Estimate friction velocity from first cell via log-law:

$$u_* = \frac{\kappa U_1}{\ln\left(\frac{z_1 + z_0}{z_0}\right)}$$



A naïve BC implementation 2/2

- Analytical steady-state value of u_*: $$u_*^{\text{target}} = \sqrt{-f_c V_g H}$$



u_* overestimated!

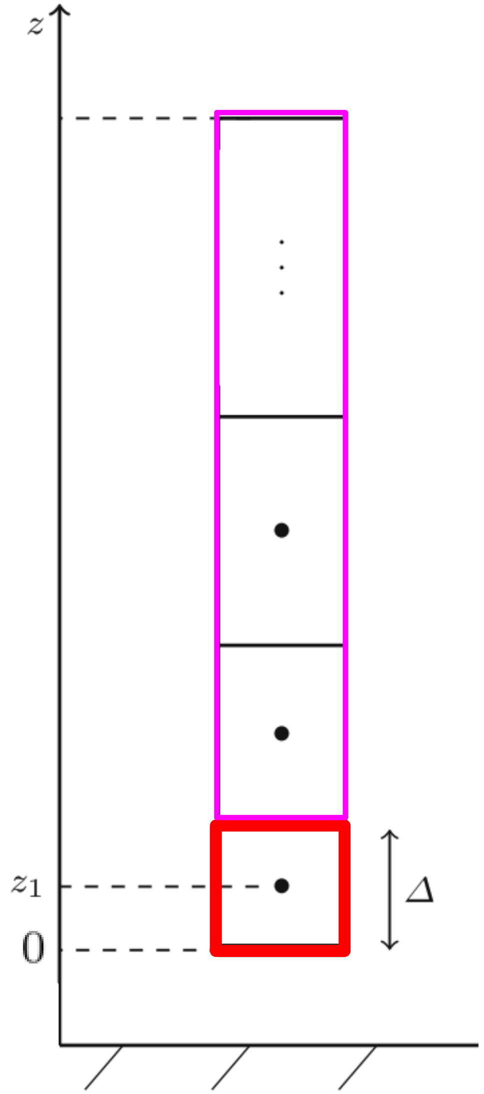
A more elaborate BC 1/4

- As suggested by Zeli et al. (2019):

$$\frac{\partial K}{\partial t} = b\boxed{(\mathcal{P}_{\log} - \varepsilon)} + (1-b)\boxed{(\mathcal{P} - \varepsilon + \mathcal{D}_K)},$$
$$\frac{\partial \varepsilon}{\partial t} = b\boxed{(\varepsilon_{\log} - \varepsilon)\, p} + (1-b)\boxed{\left(C_{\varepsilon 1}\frac{\varepsilon}{K}\mathcal{P} - C_{\varepsilon 2}\frac{\varepsilon^2}{K} + \mathcal{D}_\varepsilon\right)}$$

$$\mathcal{P}_{\log} = \frac{u_*^4}{\kappa f_m^{0.25} K_1^{0.5}(z_1 + z_0)}$$
$$\varepsilon_{\log} = \frac{f_m^{0.75} K_1^{1.5}}{\kappa (z_1 + z_0)}$$

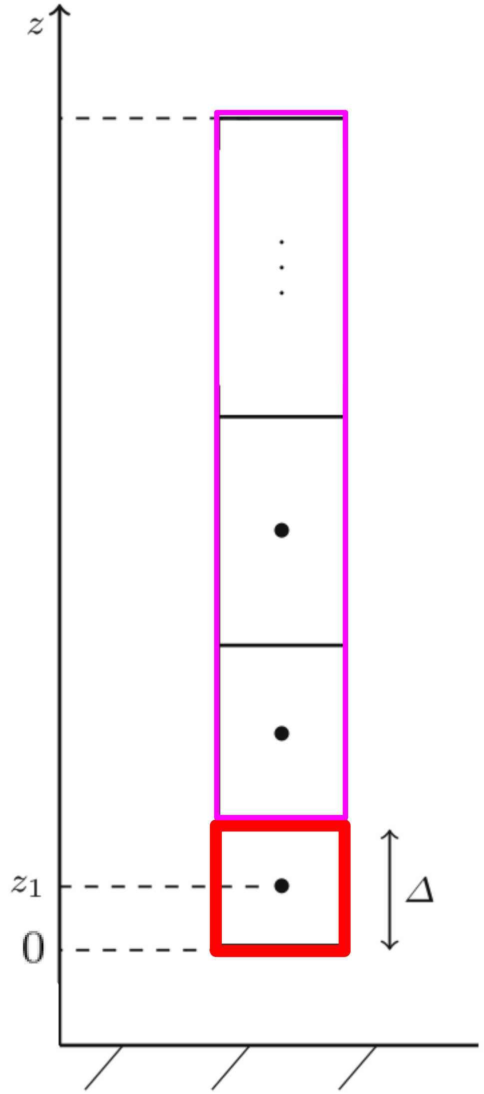
- Idea: relax ε_1 towards log-law value. Set D_K=0 in first cell-center to be consistent with log-law.
  P_{log}= ε_{log}, when K_1 attains log-law value.



A more elaborate BC 2/4

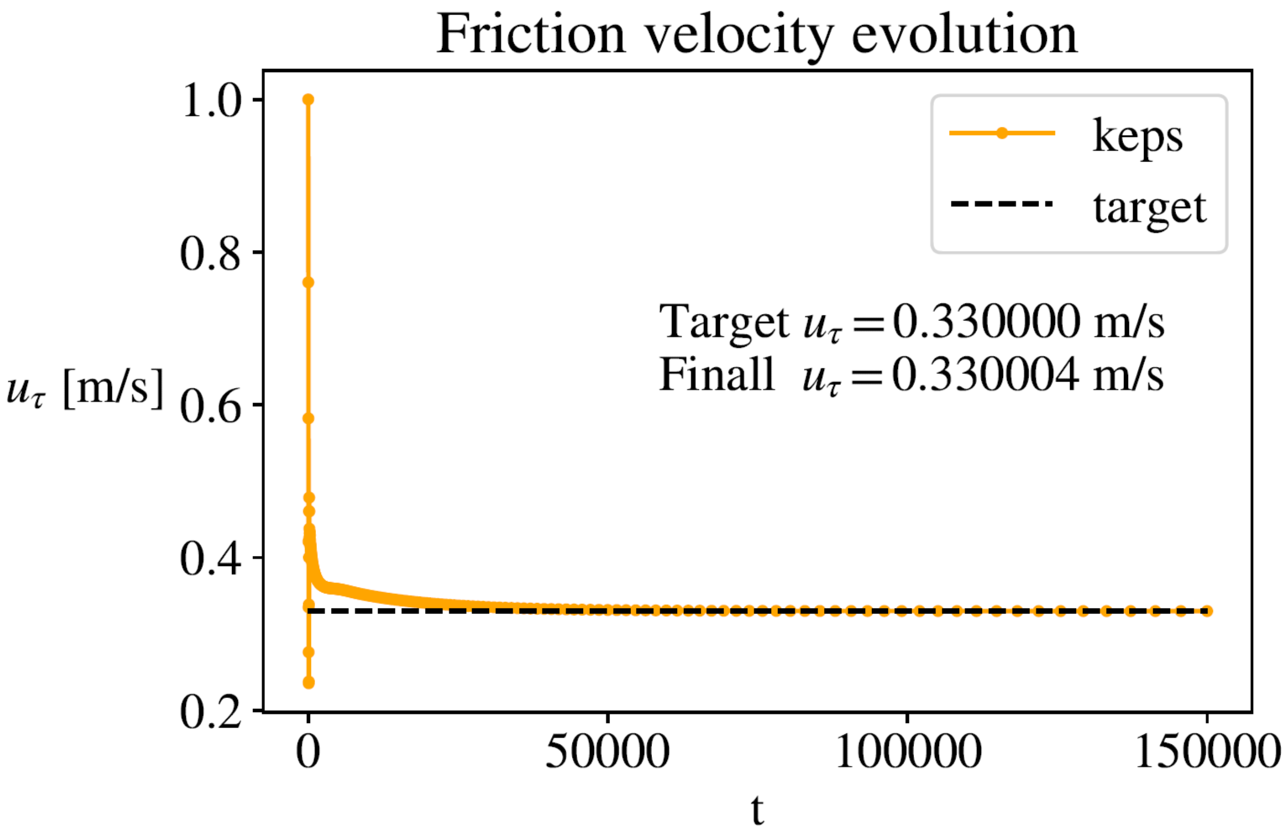
- Zeli et al. (2019) suggested the BCs:
  - $U(z=0)=0$
  - $K(z=0)=\frac{u_*^2}{\sqrt{f_m}}$
  - $\varepsilon(z=0)=\frac{u_*^3}{\kappa z_1}\ln\left(\frac{z_1+z_0}{z_0}\right)$
- I found the U=0 BC to be problematic for numerical convergence → changed to flux BC*:

$$\left(\nu_t \frac{\partial U}{\partial y}\right)\Bigg|_{z=0} = u_*^2$$



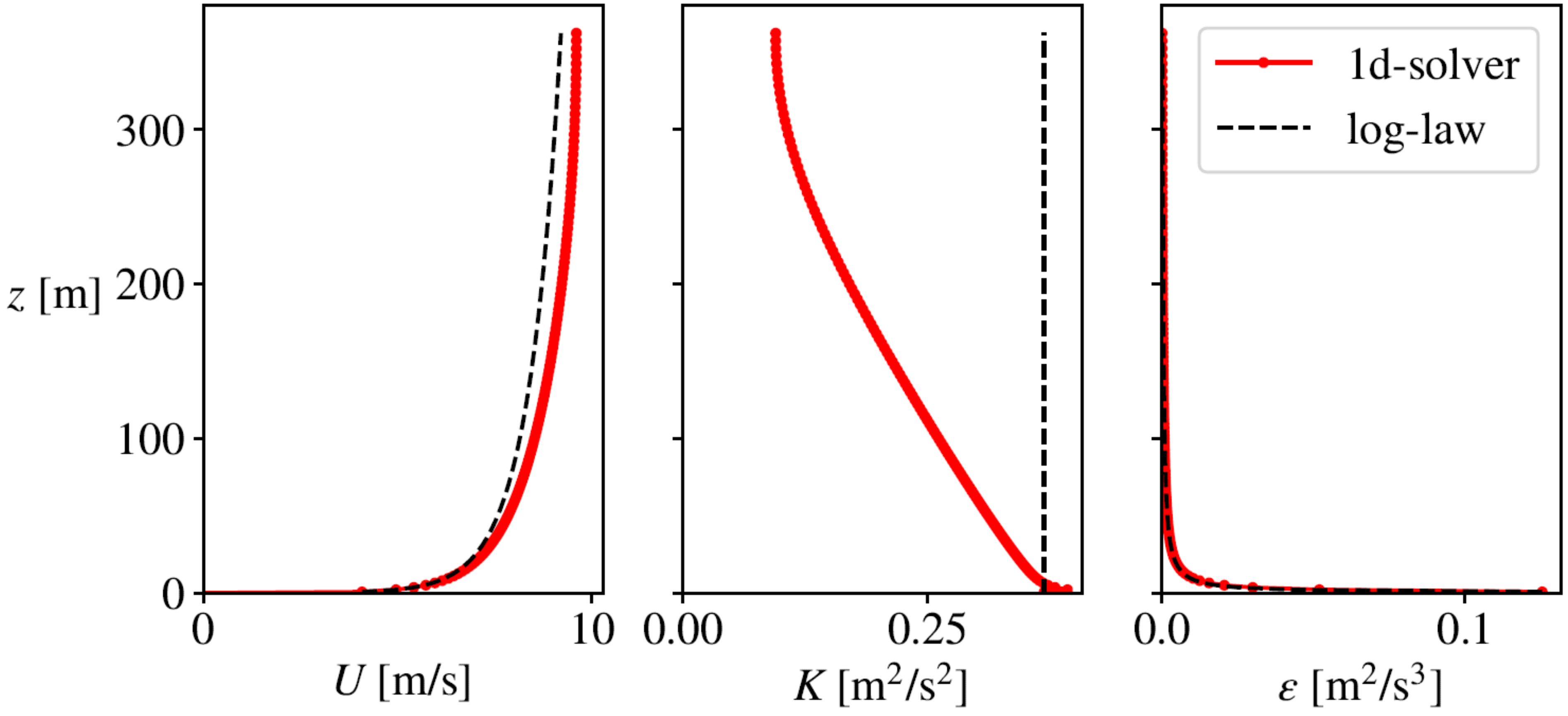
*(and set an artificial wall eddy viscosity to still obtain U(0)=0; could then set dK/dy=deps/dy=0)

A more elaborate BC 3/4



Converge to the correct value!

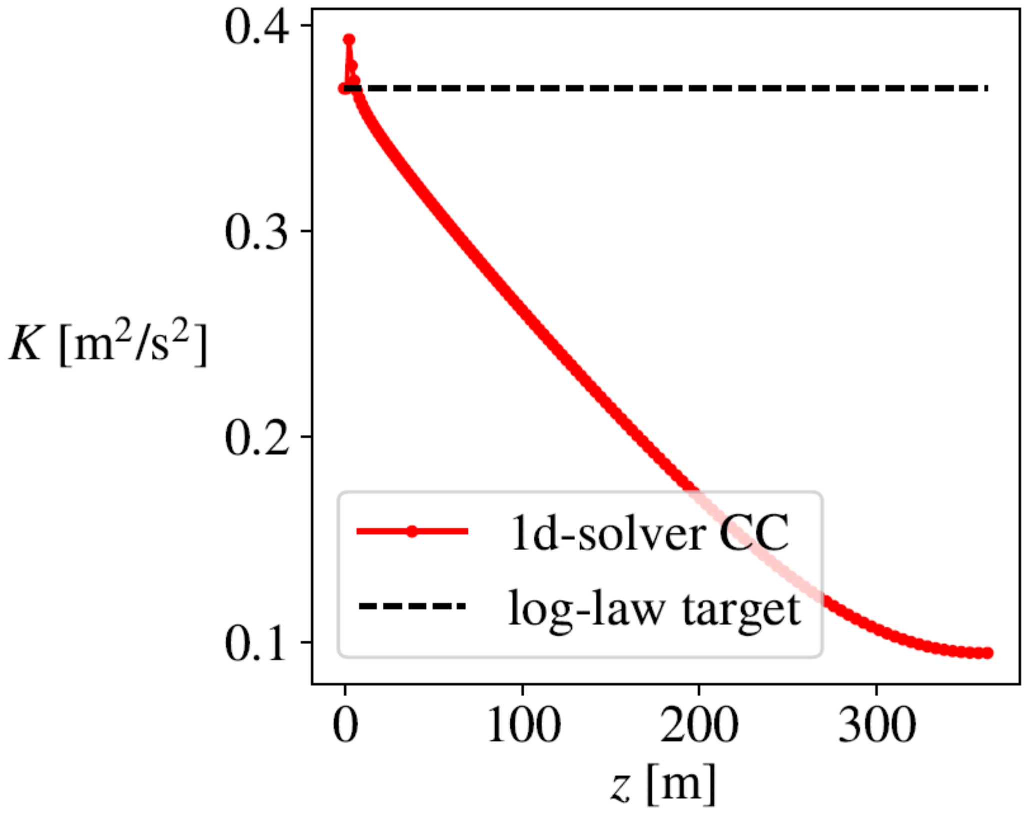
A more elaborate BC 4/4

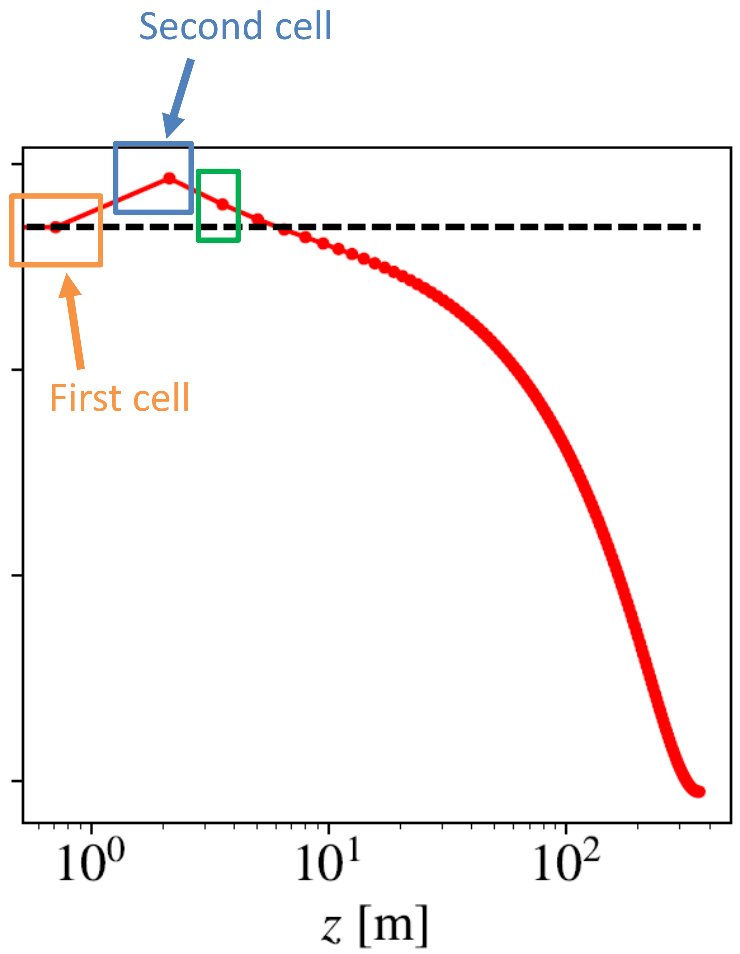


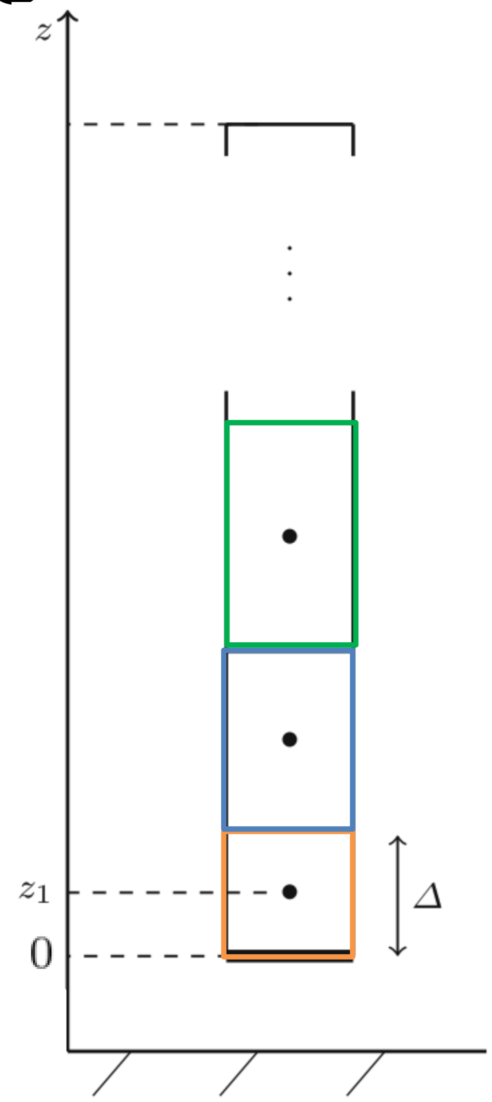
The profiles obey log-law at bottom of domain,
except a small overshoot of TKE.

The "TKE overshoot"-problem 1/3

- Overshoot occurs in *second* cell.



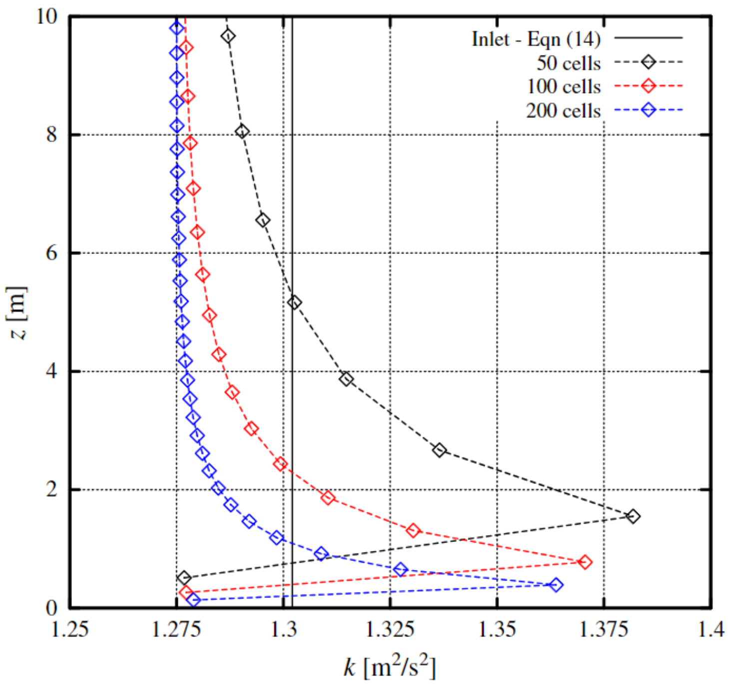




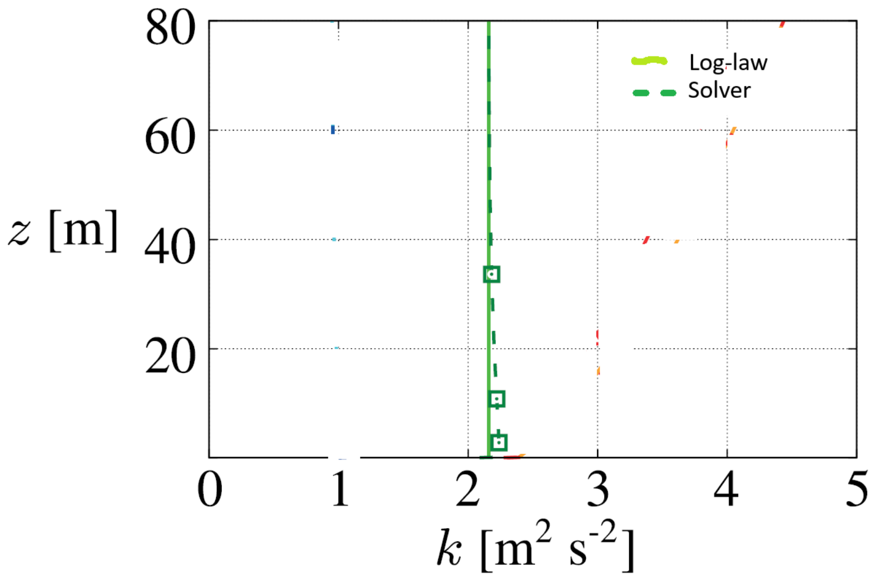
The “TKE overshoot”-problem 2/3

- Not only for channel flow; occurs for any ABL type.
- A commonly observed problem in various codes.

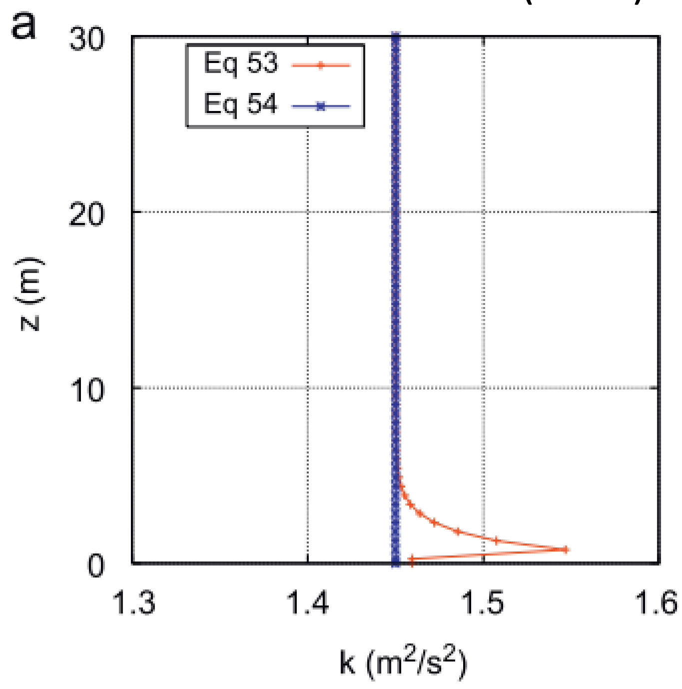
Sumner & Masson (2012)



van der Laan et al. (2017)



Richards & Norris (2011)



The “TKE overshoot”-problem 3/3

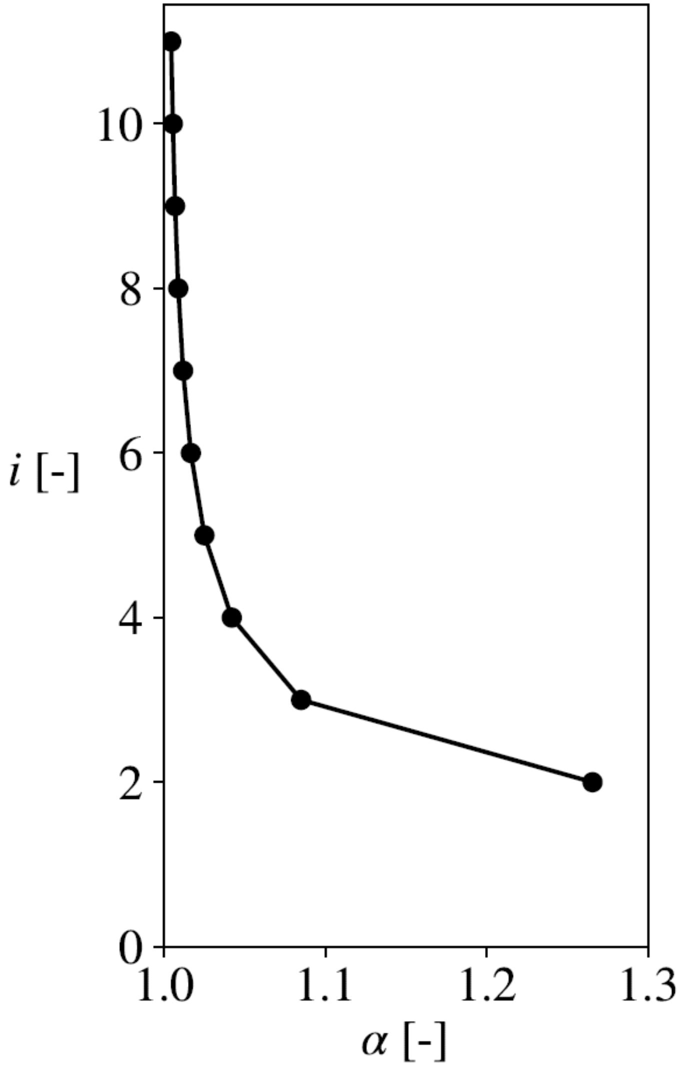
- It is a discretization problem connected to the shear stress.

$$\mathcal{D}_U = -\frac{\partial \overline{uw}}{\partial z} \Rightarrow \mathcal{D}_U\big|_{z_i} = -\frac{\overline{uw}_{i+\frac{1}{2}} - \overline{uw}_{i-\frac{1}{2}}}{\Delta z_i}$$

$$\mathcal{P} = -\overline{uw}\frac{\partial U}{\partial z} \Rightarrow \mathcal{P}\big|_{z_i} = -\overline{uw}_i\left(\frac{U_{i+\frac{1}{2}} - U_{i-\frac{1}{2}}}{\Delta z_i}\right)$$

- Richards & Norris (2011) derived an analytical estimate of the production overestimation:

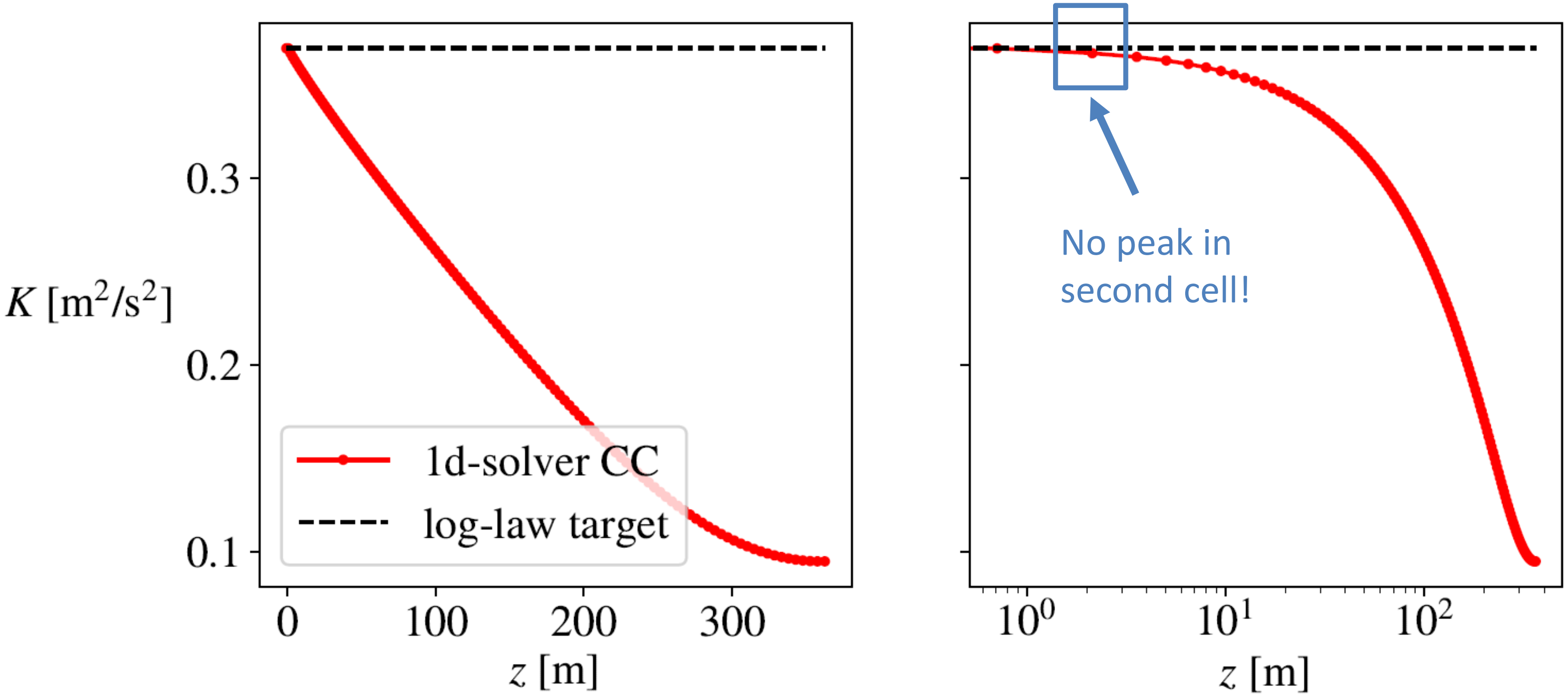
$$\mathcal{P}\big|_{z_i} = \frac{u_*^3}{\kappa z_i}\underbrace{\left(\frac{1}{1-\frac{1}{4}\left(\Delta z/z_i\right)^2}\right)^2}_{\alpha}$$



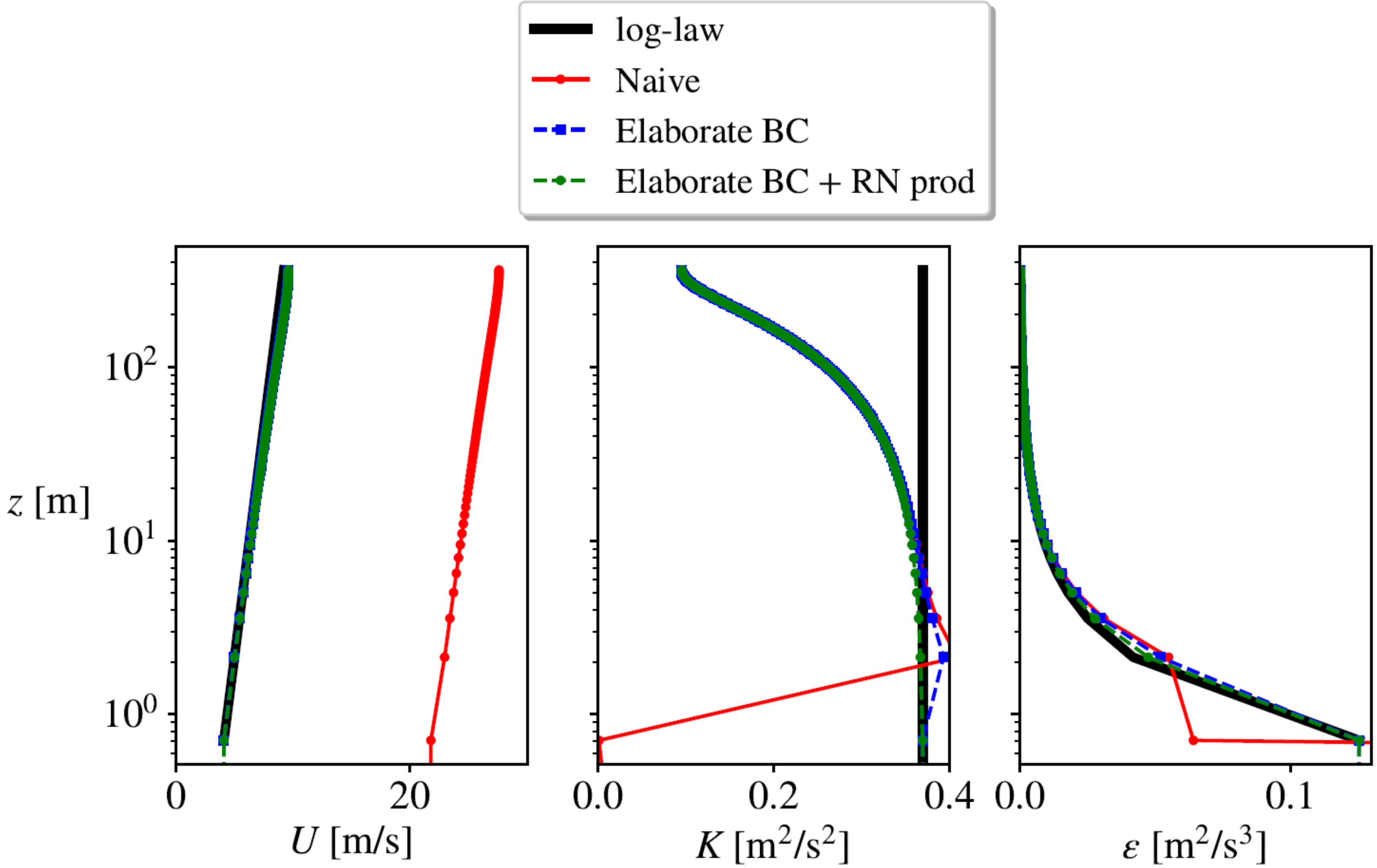
A solution to the “TKE overshoot”-problem

- Richards & Norris (2011) suggested an alternative discretization of the shear production:

$$\mathcal{P}\big|_{z_i} = \frac{1}{2}\left(\overline{uw}^2_{i-\frac{1}{2}} + \overline{uw}^2_{i+\frac{1}{2}}\right)\frac{1}{\nu_{t,i}}$$

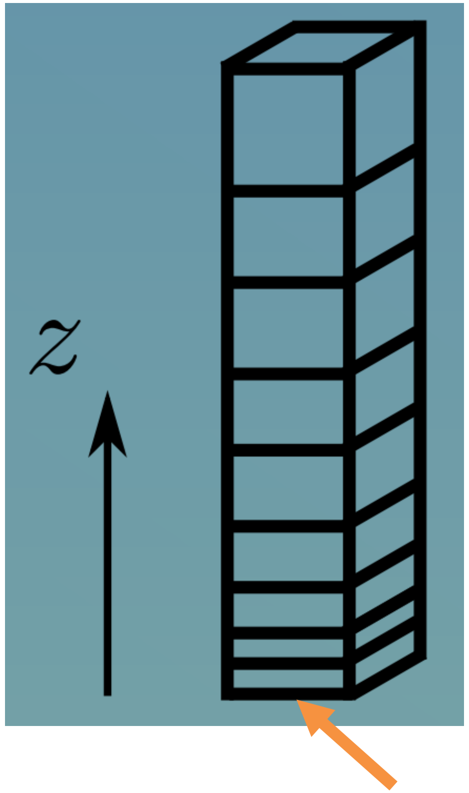


Comparison of methods



Summary

1

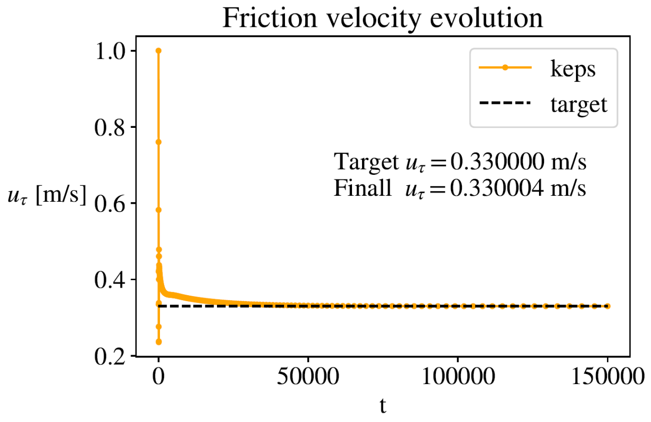


Wall BC: log-law

- All ABL simulations are based on log-law BC.

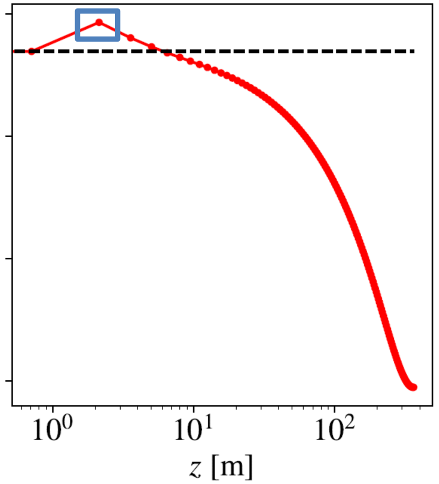
2

- Modify K and ε eqs.
- Use flux BC for U.



3

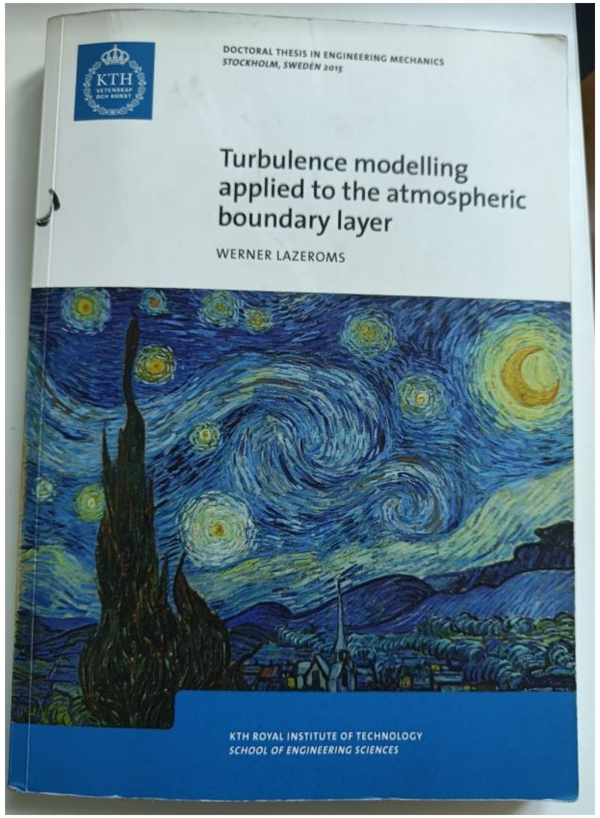
- TKE overshoot often observed in 2nd cell.
- Can be removed by consistent discretization.

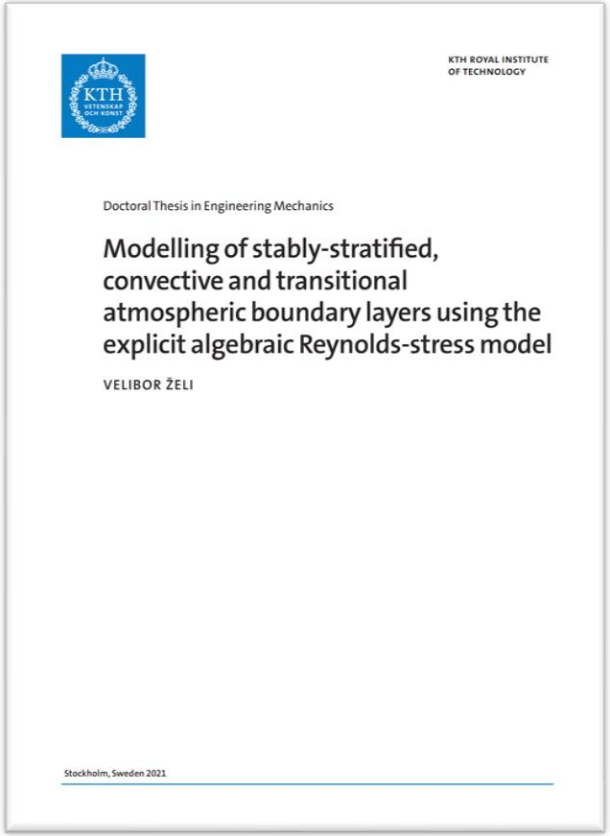


Extra slides

Turbulence modelling for ABL

- Need to model $\overline{uw}$, $\overline{vw}$ and $\overline{w\theta}$.
- Recent work by Lazeroms (2015) and Zeli (2021) investigated using Explicit Algebraic Reynolds Stress models (EARSMs).





Explicit Algebraic Reynolds Stress Model (EARSM)

- Three algebraic expressions:

$$\overline{uw} = f_1\left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}, \frac{\partial \Theta}{\partial z}, K, K_\theta, \varepsilon\right)$$
$$\overline{vw} = f_2\left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}, \frac{\partial \Theta}{\partial z}, K, K_\theta, \varepsilon\right)$$
$$\overline{w\theta} = f_3\left(\frac{\partial U}{\partial z}, \frac{\partial V}{\partial z}, \frac{\partial \Theta}{\partial z}, K, K_\theta, \varepsilon\right)$$

- Three transport equations:

$$\frac{\partial K}{\partial t} = \mathcal{P} - \varepsilon + \mathcal{G} + \mathcal{D}_K,$$
$$\frac{\partial K_\theta}{\partial t} = \mathcal{P}_\theta - \varepsilon_\theta + \mathcal{D}_{K_\theta},$$
$$\frac{\partial \varepsilon}{\partial t} = C_{\varepsilon 1}\frac{\varepsilon}{K}\mathcal{P} - C_{\varepsilon 2}\frac{\varepsilon^2}{K} + C_{\varepsilon 3}\frac{\varepsilon}{K}\mathcal{G} + \mathcal{D}_\varepsilon$$

Shear stresses

- Should plot the shear stresses at the faces or use a nodes-average:

$$\overline{uw}_i = \frac{1}{2}\left(\overline{uw}_{i+\frac{1}{2}} + \overline{uw}_{i-\frac{1}{2}}\right)$$