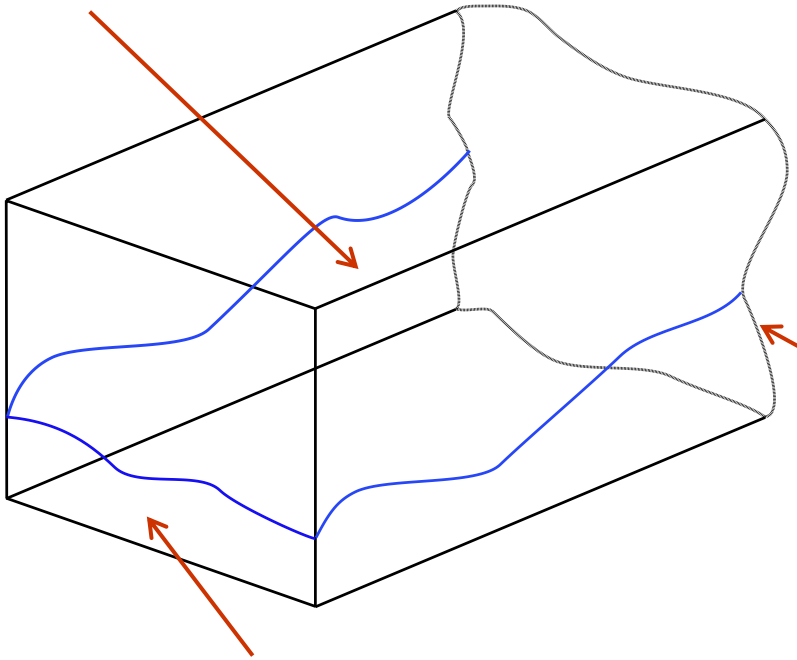
UPPSALA UNIVERSITET

A robust approach to generating inflow conditions for scale-resolving simulations of turbulent boundary layers

Timofey Mukha and Mattias Liefvendahl

Overview of the problem

Turbulent boundary layer (TBL)



Inflow patch

In case of **LES/DNS** we need a time-dependent Dirichlet-type boundary condition for the velocity field.

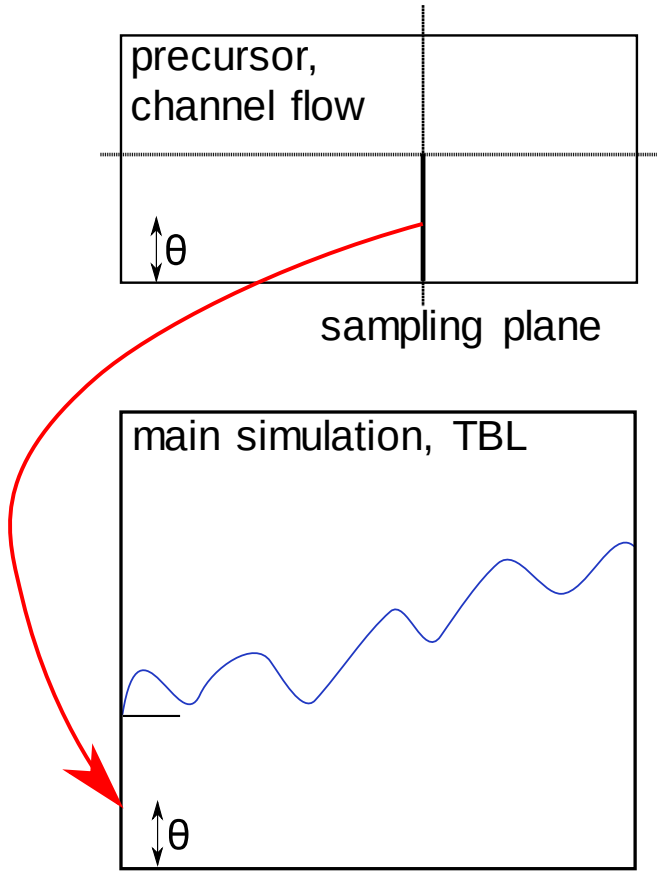
It should mimic the velocity distribution found in a TBL.

Downstream

- Diffuser
- Ramp
- BFS
- etc

Proposed framework

Channel flow as a precursor



Set-up

- Fully developed channel flow is used as a precursor simulation.
- Velocity is sampled from a cross-plane and prescribed at the inlet of the TBL without change.

Advantages

- Simple and robust.
- Only requires Re-number at the inlet as input.
- Minimal coding to implement.

Implementation
Introducing `eddylicious`

`Eddylicious` is a small python package with big ambition!

- Become a placeholder for various inflow generation methods.
- Support different CFD solvers.
- Be easily extendable.
- Be very well-documented, provide tutorials.
- Be open source.

"The most valuable study of inlet conditions for LES would be one which implements all the… methodologies in the same code…"
(Tabor and Baba-Ahmadi, Computers & Fluids, 39, 2010)

UPPSALA UNIVERSITET

Implementation
Introducing `eddylicious`

`Eddylicious` is split in four different parts

- **Readers.** Read stored velocity distributions from the hard-drive.
- **Writers.** Write the generated fields to the hard-drive.
- **Generators.** Actually generate the inflow.
- **Executables.** Configurable scripts/utilities.

UPPSALA UNIVERSITET

Implementation
Introducing `eddylicious`

`Eddylicious` is split in four different parts

- **Readers.**
  - `foamFile`-format from `OpenFOAM`.
  - A single HDF5 file.

- **Writers.**
  - The `timeVaryingMappedFixedValue` b.c. in `OpenFOAM`.
  - A single HDF5 file.

  A modification of `timeVaryingMappedFixedValue` that reads from HDF5 can be found on bitbucket.

- **Generators.**
  - The proposed method + rescaling procedure of Lund et al*.
  - Precursor can either be a channel flow or a TBL.

*Lund et al, J. Comp. Phys., 140:233-58, 1998.

Numerical experiments

Set-up

Solver set-up

- `OpenFOAM 2.3.1`
- `pimpleFoam`
- `linear` scheme for convective fluxes.
- `backward` scheme for time-stepping.
- LES, no SGS modelling.

Mesh

- $\Delta x^+ \approx 50, \Delta z^+ \approx 20, y_1^+ \approx 1.$
- Grids match up to $y = \delta$.
- $N_{prec} \approx 1.7 \cdot 10^6,\ N_{main} \approx 1.7 \cdot 10^7$

Performed simulations

- Precursor channel flow, $\mathrm{Re}_b = 10000.$

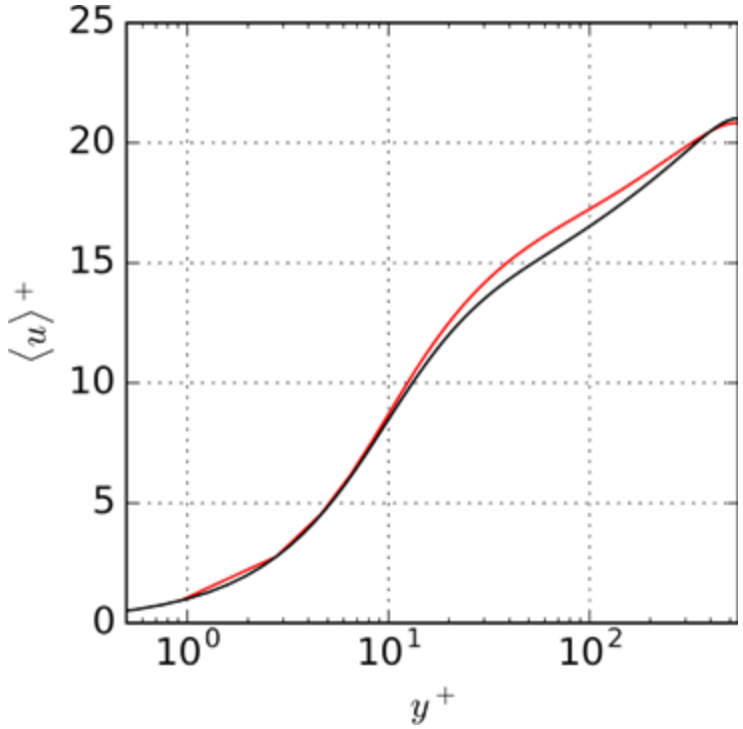
Gives $\mathrm{Re}_\theta \approx 833.$

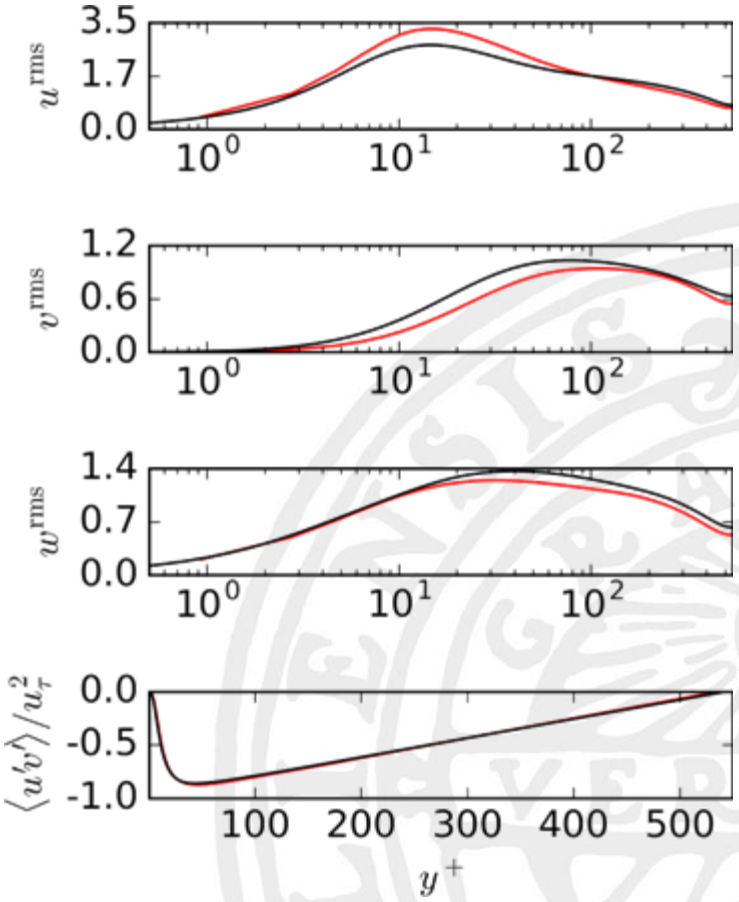
- Zero pressure gradient flat-plate TBL, for range $\mathrm{Re}_\theta \in [833, 2400]$.

UPPSALA UNIVERSITET

Numerical experiments

Precursor channel flow





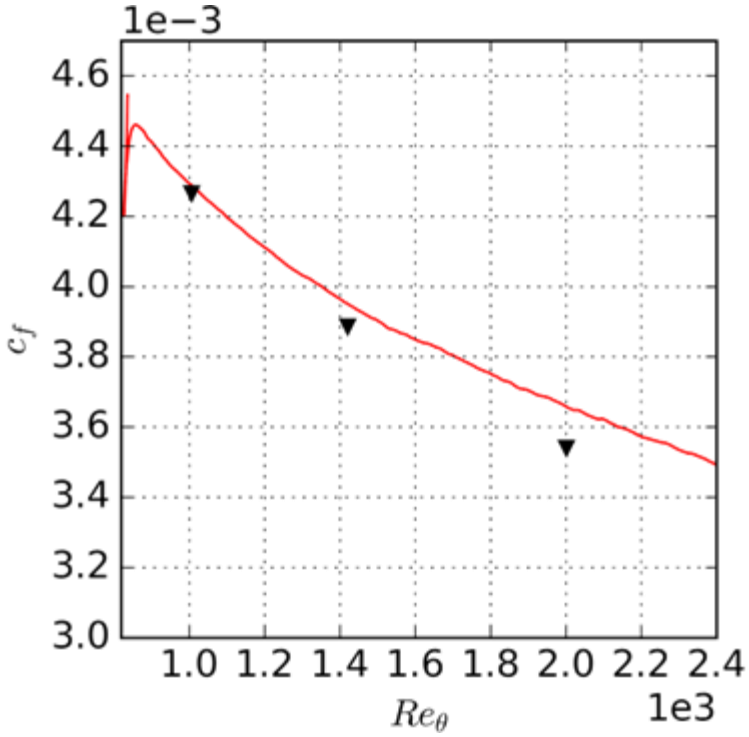
proposed method

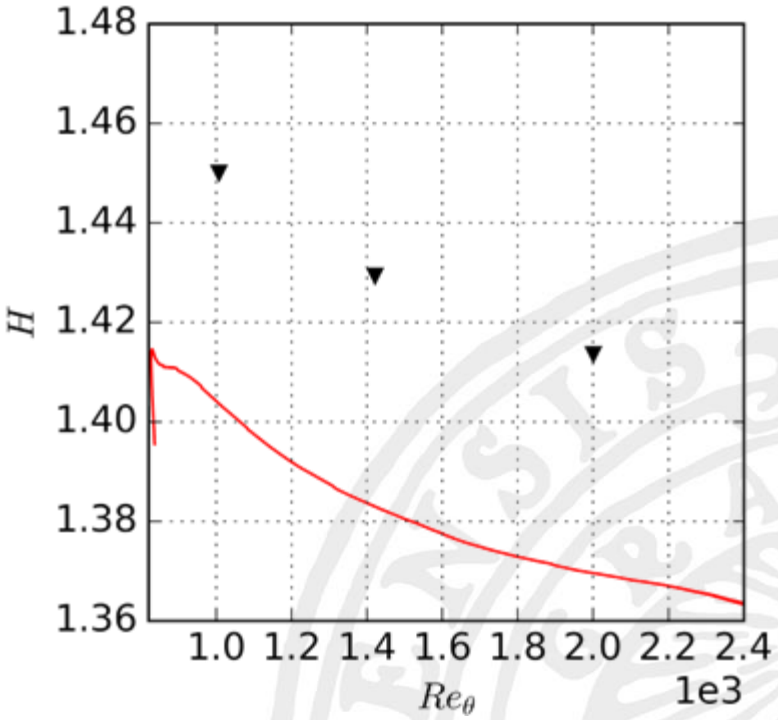
DNS. Lee and Moser, J. Fluid Mech., 774, 395-415, 2015

UPPSALA UNIVERSITET

Numerical experiments

Main TBL simulations





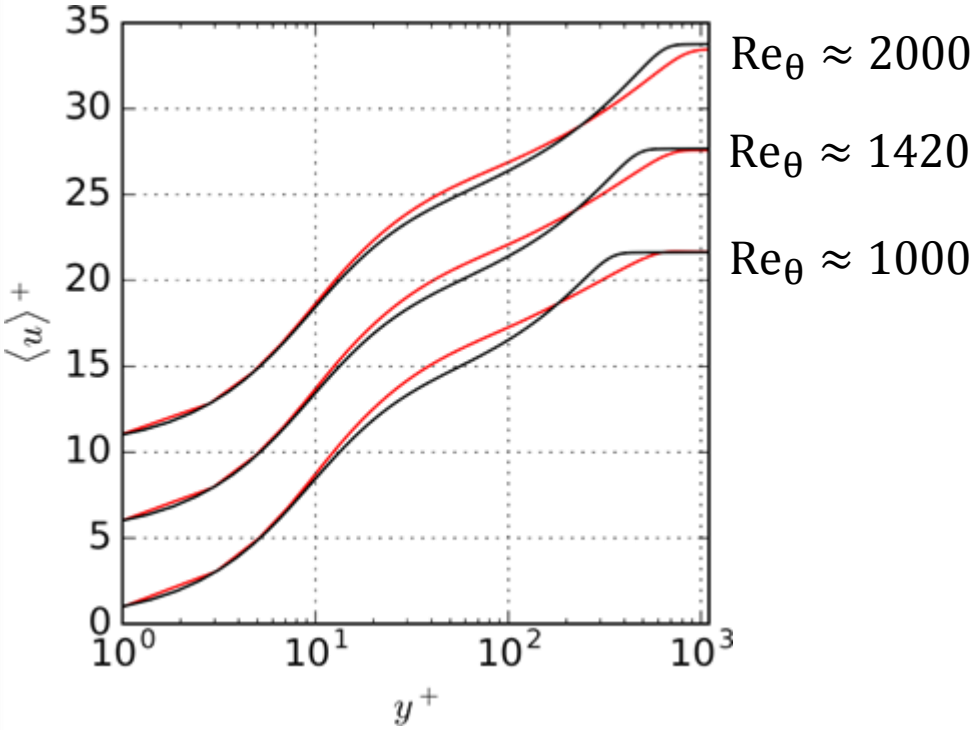
proposed method

DNS. Schlatter and Örlü, J. Fluid Mech., 659, 116-126, 2010.

UPPSALA UNIVERSITET

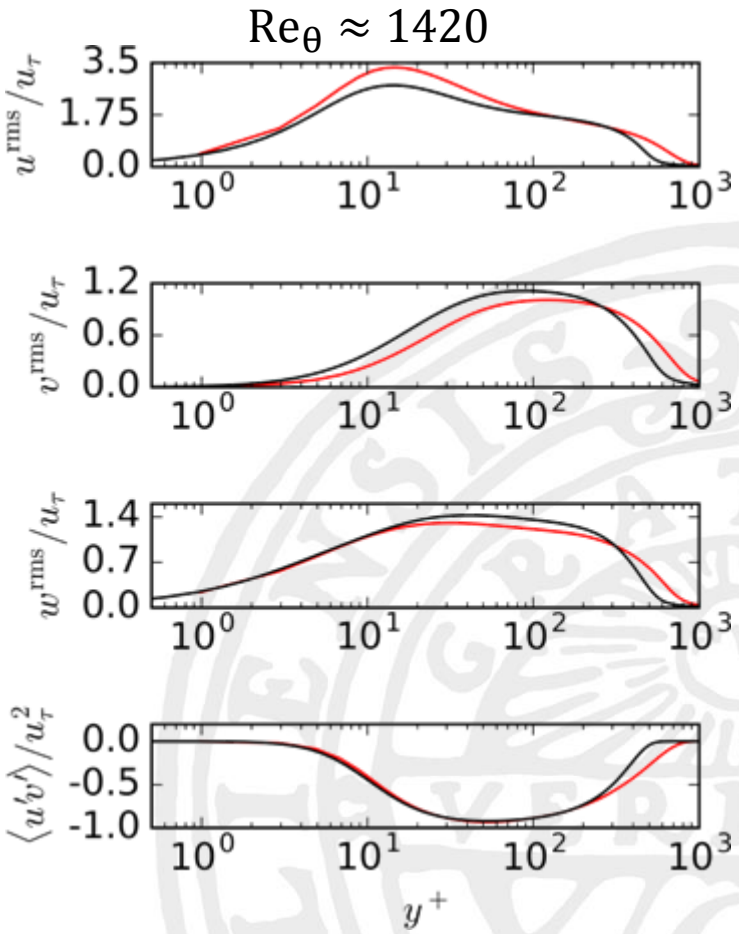
Numerical experiments

Main TBL simulation



proposed method

DNS. Schlatter and Örlü, J. Fluid Mech., 659, 116-126, 2010.



UPPSALA UNIVERSITET

Conclusions

- Using channel flow for generating inflow for simulations involving a TBL is considered.
- The observed adaption length of the flow is $\approx 30\delta_{99}^{inflow}$.
- A new python package, `eddylicious`, is available. The purpose is uniting existing inflow generation methods under one roof.
- The modular system of the package allows to easily extend it for using with new CFD codes and with new generation methods.

UPPSALA UNIVERSITET

Thank **you** for listening!

And the **PDC Centre for High Performance Computing** for making these computations possible!