[Airbus Amber]

Learning wall models for CFD

OpenTURNS users day 2021

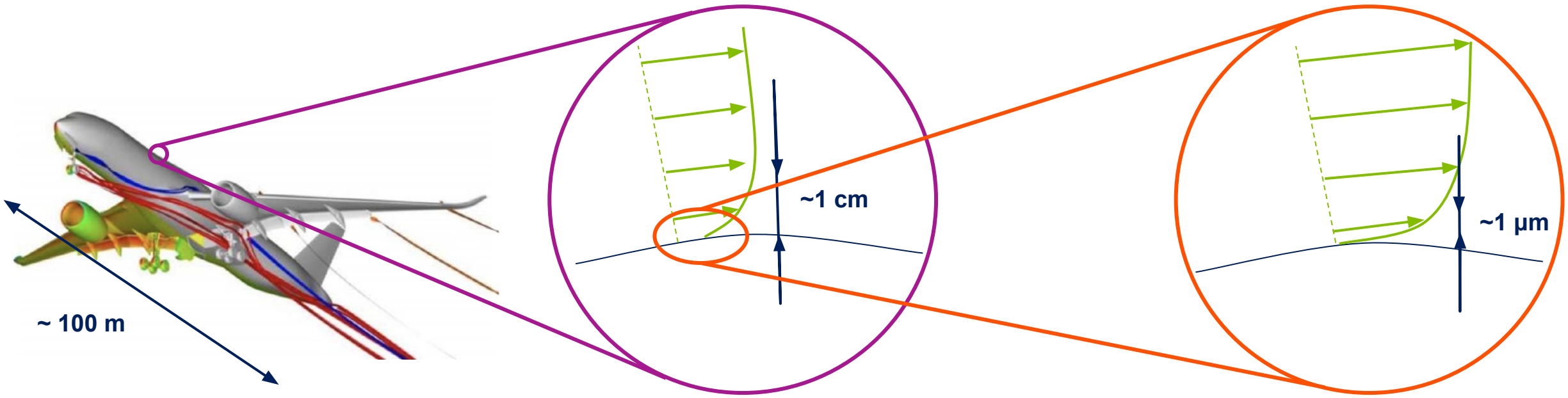
XRV

11/06/2021

AIRBUS

Turbulent boundary layers

Length scales in time-averaged turbulent boundary layers



Boundary layer thickness
E.g. 99% of the free stream velocity

Inner layer thickness
Linear increase of the velocity with the wall distance

Is the near wall behavior universal / independent of the configuration?

AIRBUS

Turbulent boundary layer profiles

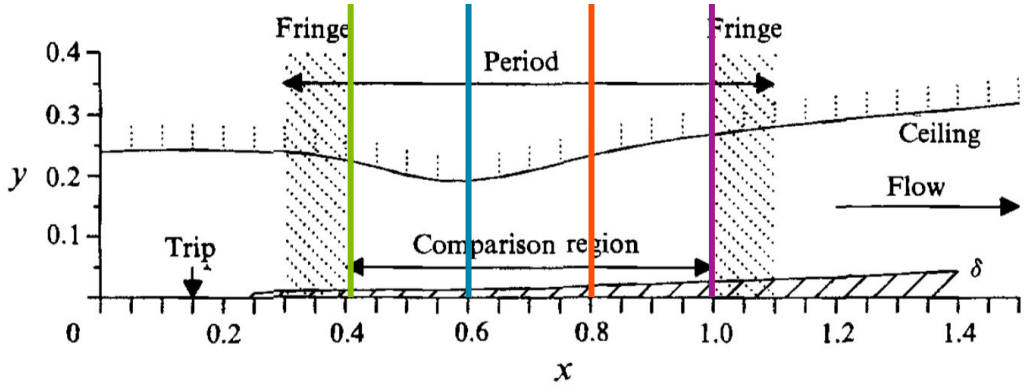
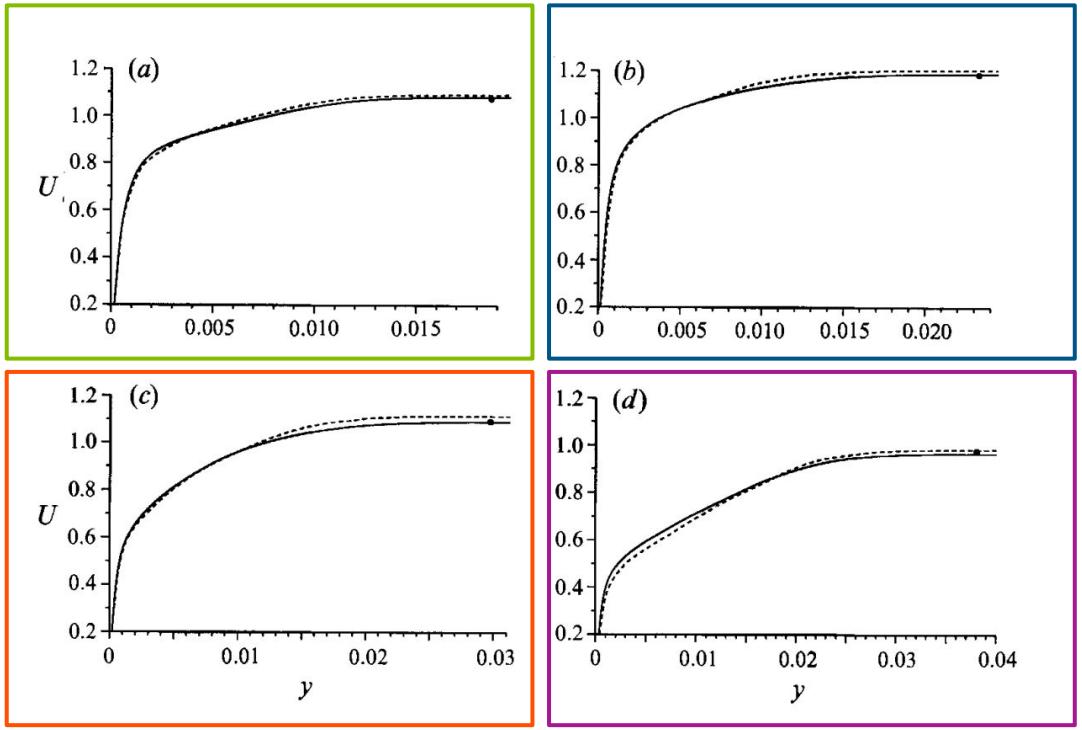


FIGURE 1. Flow configuration.



Example of an incompressible flow configuration studied by

- Experimental measurements
- Direct Numerical Simulation (DNS) : unsteady simulation of all the scales present in the flow

Spalart & Watmuff, 1993

AIRBUS

Turbulent boundary layer profiles

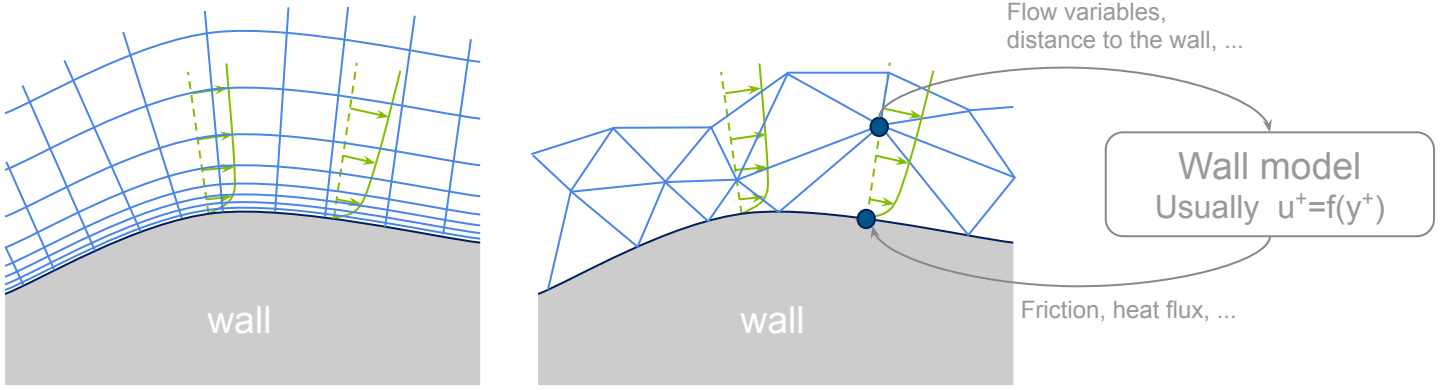
Example of an incompressible flow configuration studied by

- Experimental measurements
- Direct Numerical Simulation (DNS) : unsteady simulation of all the scales present in the flow

Wall-unit scaling:

- Wall shear stress : $\tau_w \rightarrow$ shear velocity $u_\tau = (\tau_w/\rho)^{1/2}$
- Profiles of $u^+=u/u_\tau$ as a function of $y^+=yu_\tau/\nu$ are *more or less* universal *close to the wall*

⇒ when/how can this be used to make computations more efficient?



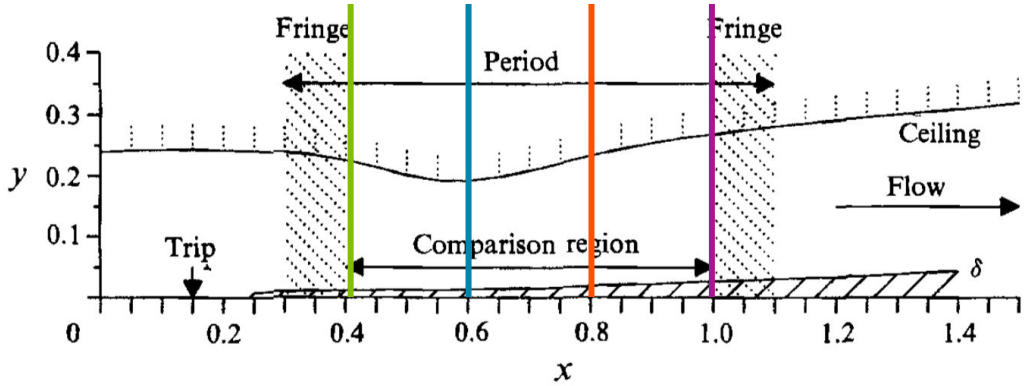
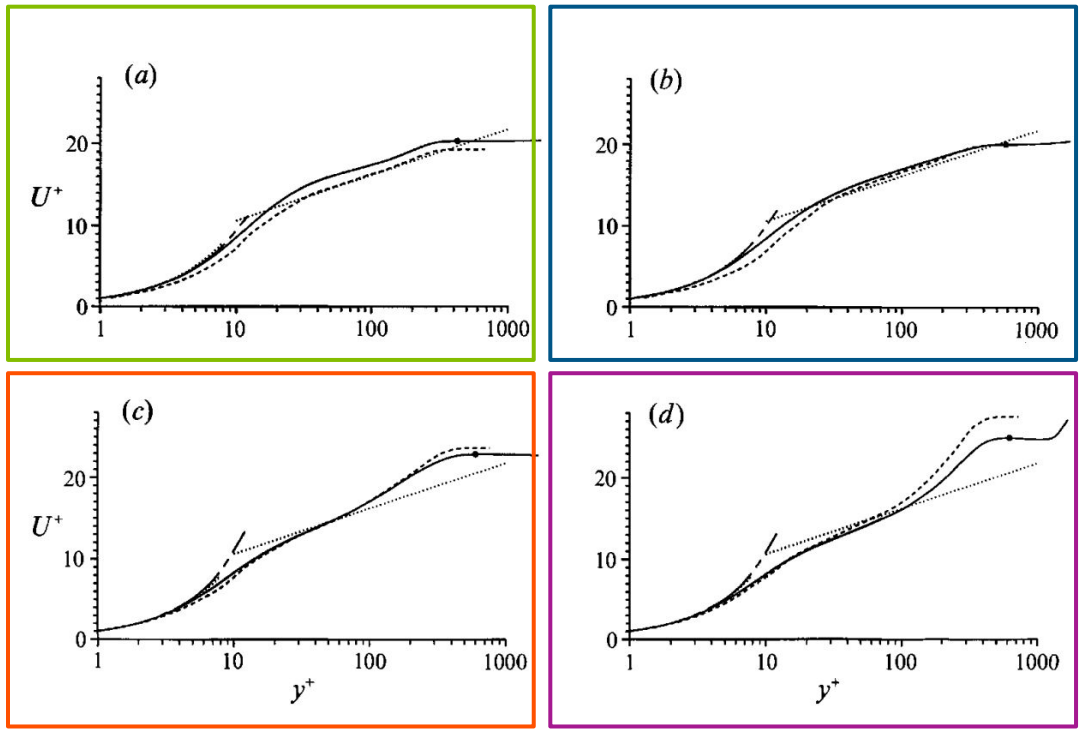


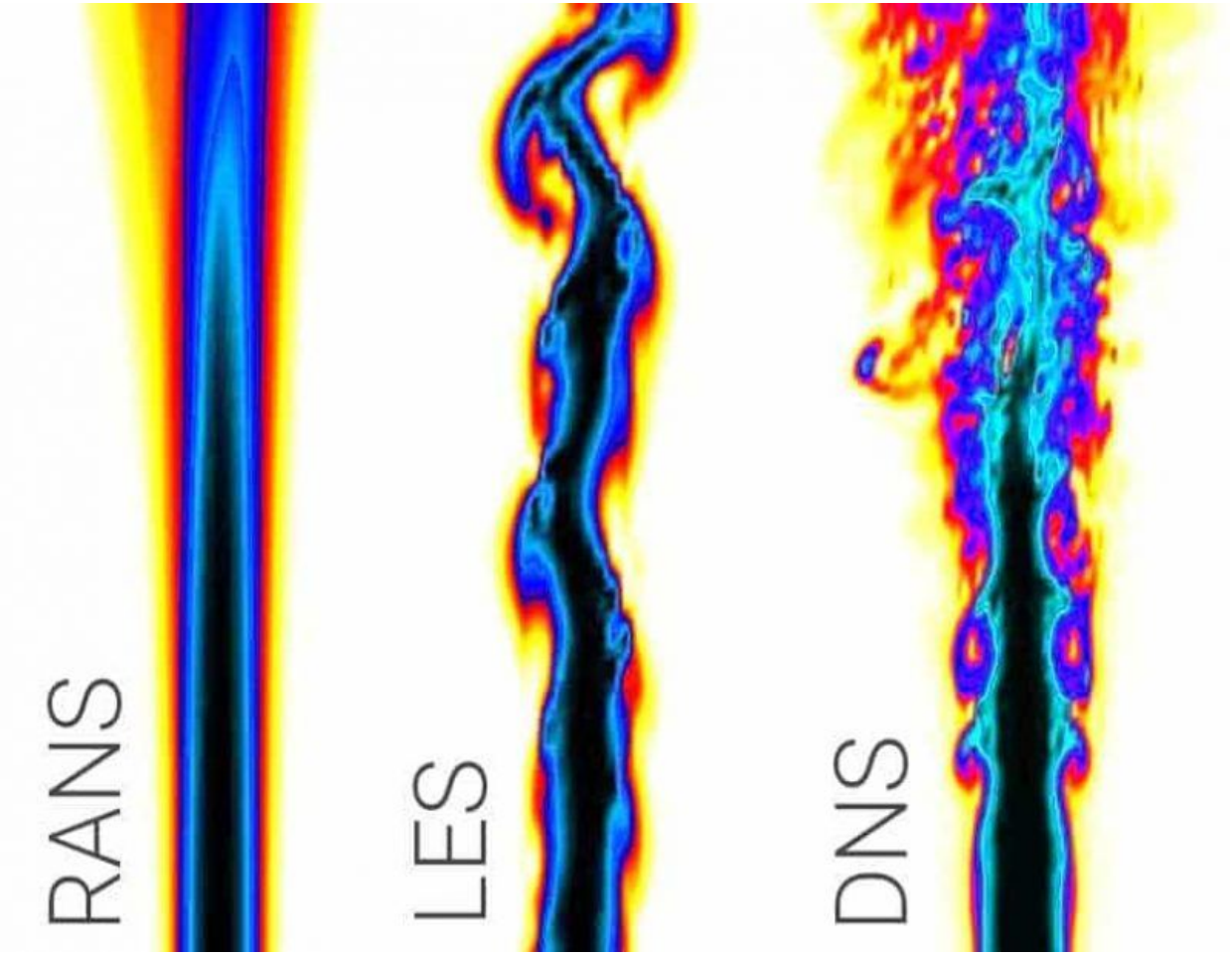
FIGURE 1. Flow configuration.



Spalart & Watmuff, 1993

AIRBUS

Turbulence modeling



With increasing complexity

- Steady RANS
- Unsteady RANS :
  - Only large scale unsteady motion
  - Industrial applications: for propeller simulations
- Hydrid RANS/LES or WMLES :
  - Steady boundary layers, unsteady shear layers & mixing
  - Industrial applications : aeroacoustics, combustion
- LES
  - No industrial application
- DNS
  - No industrial application

Wall models are applicable*

Wall models are applicable for RANS, URANS, hybrid RANS/LES and WMLES but not used when viscous drag is of interest as they are not accurate enough

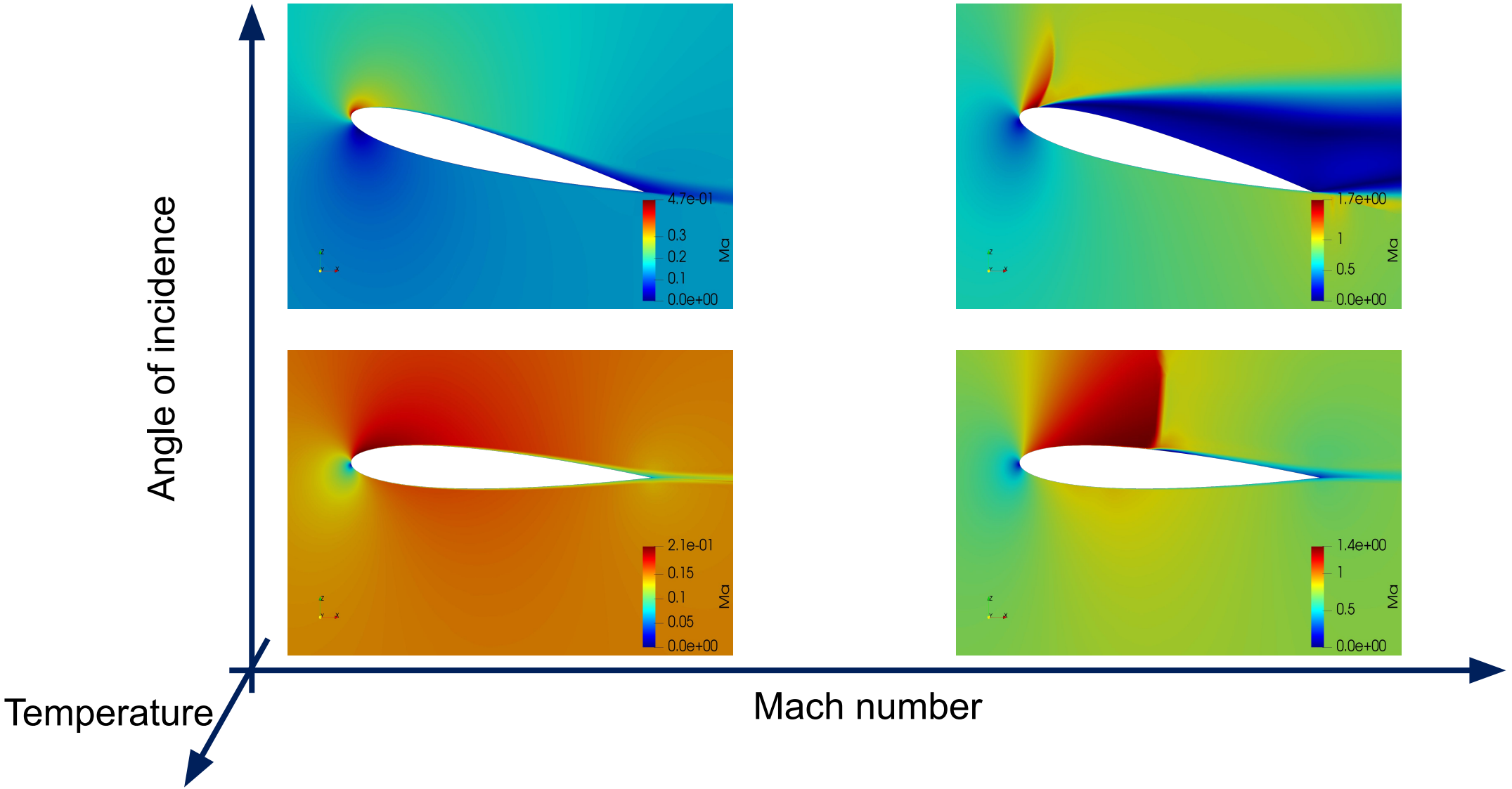
⇒ Can we learn more complex wall models to have better accuracy on complex configurations?

AIRBUS

Learning data & objectives

2D NACA0012 configuration

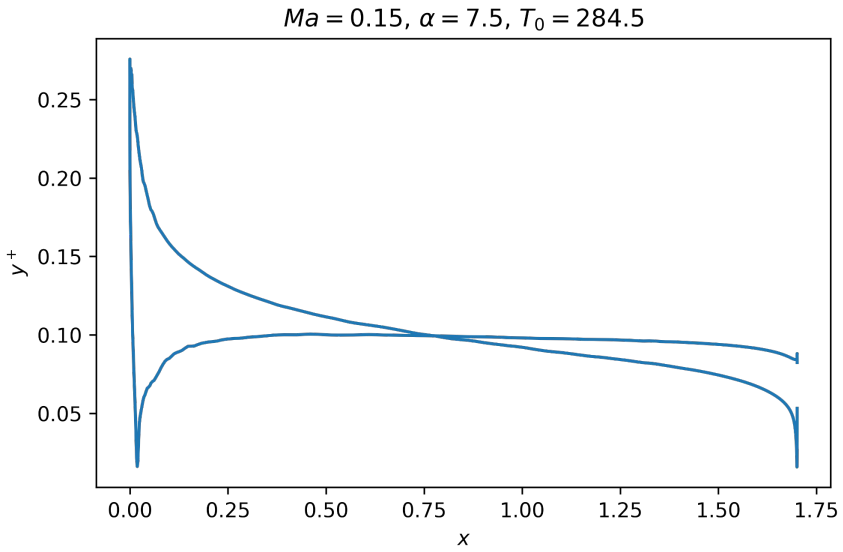
Acceleration & deceleration, separation, shocks



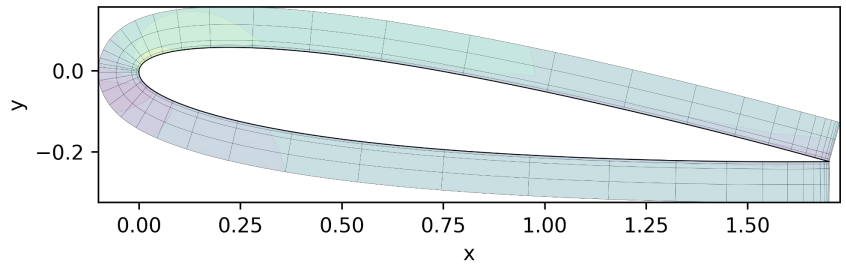
AIRBUS

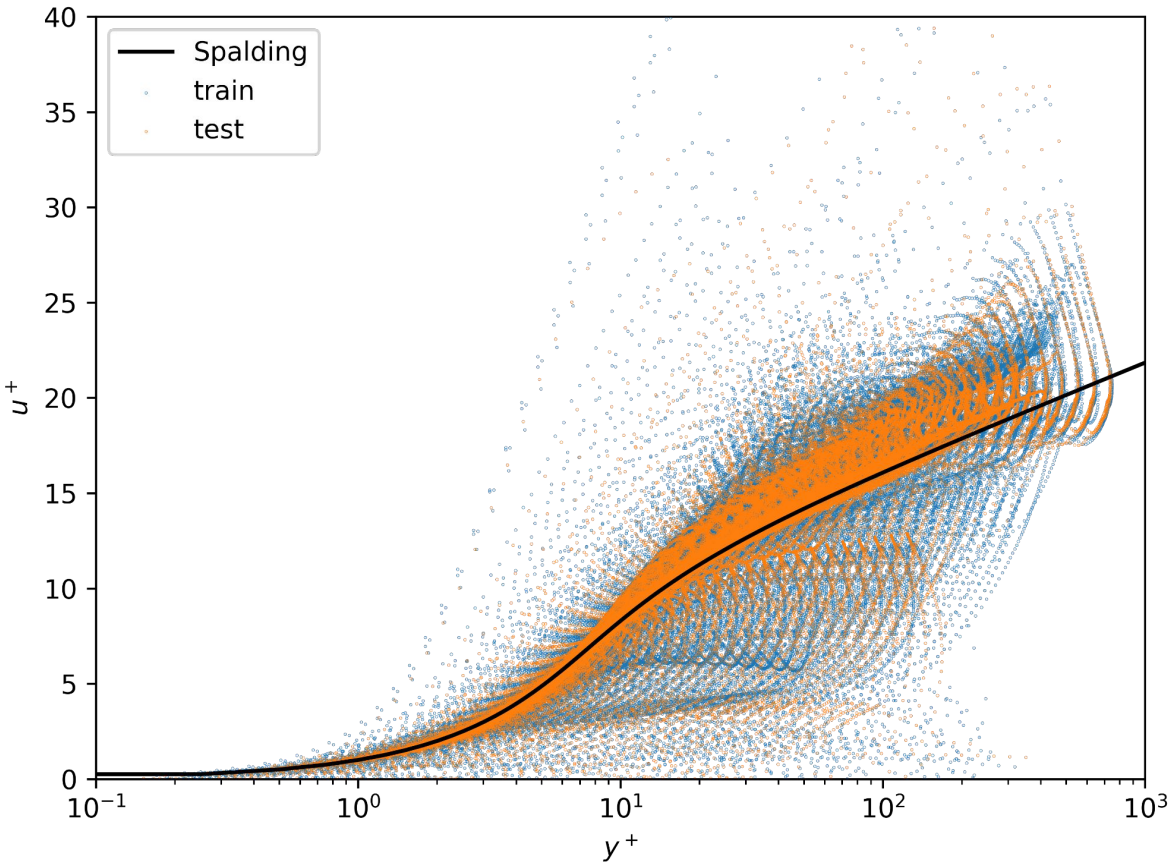
Interpolation of the flow field on a structured grid around the profile

- Wall-resolved simulations performed on 2D configurations



- Extraction of flow variables (U, P, T, k, ω, …) and their gradients in the wall-normal and tangential directions





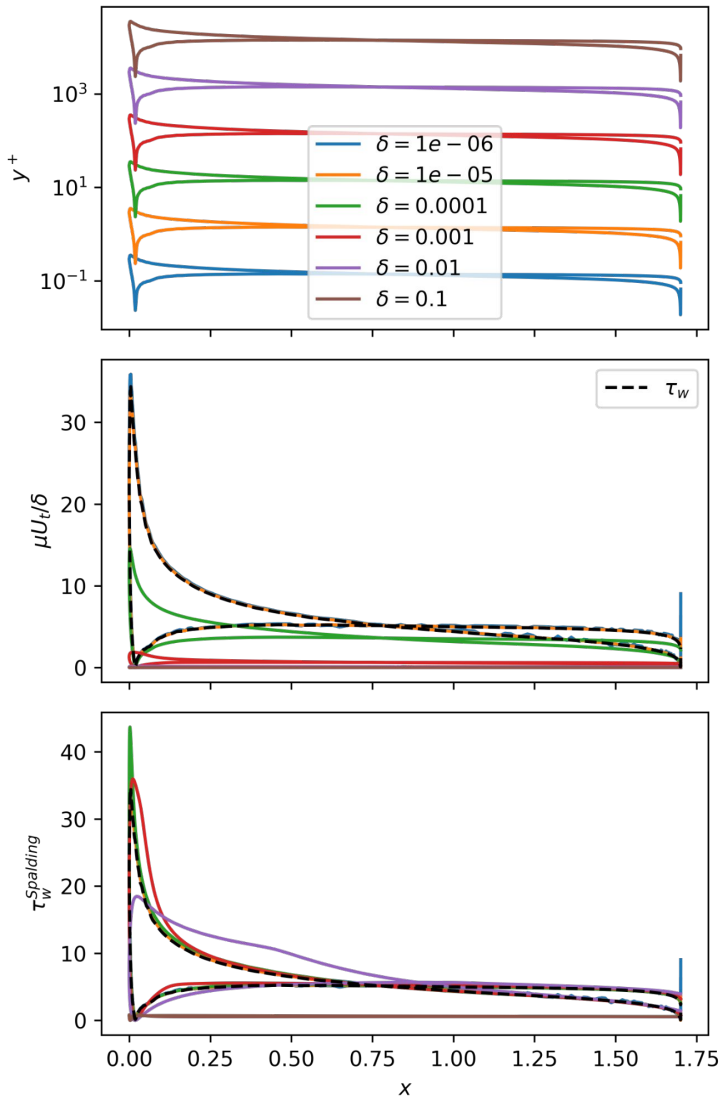
Velocity profile in wall units for the CFD data & comparison with a classical relation $u^{\pm}=f(y^{\pm})$ (Spalding)

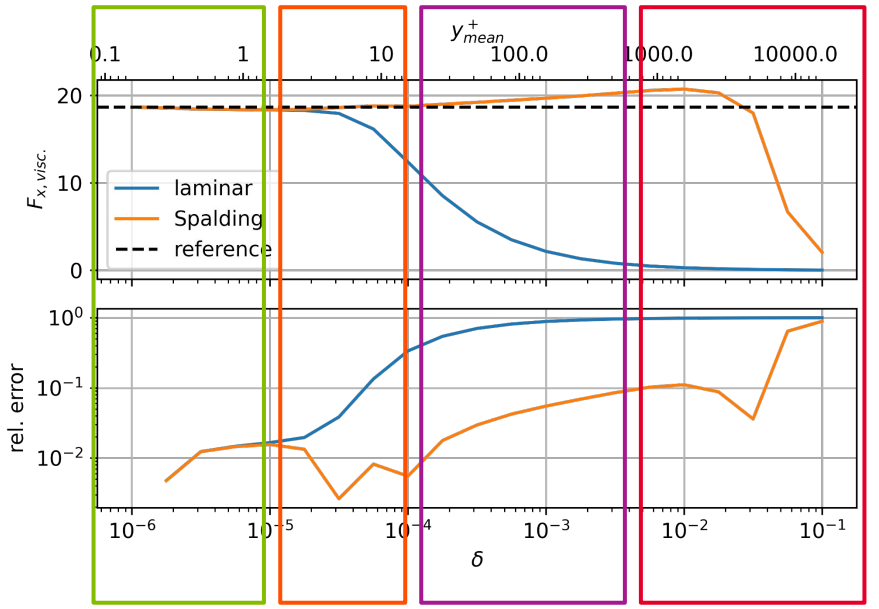
AIRBUS

Example

Ma=0.15, α=7.5°, T_0=284.5K

- Data interpolated at distances δ=10^{-6} → 10^{-1} from the wall
- The shear stress is assessed:
  - from the simulation (using the simulation mesh)
  - assuming velocity varies linearly up to δ
  - with a classical wall model
- Discrepancies occur mostly close to the leading edge, where curvature / acceleration / deceleration are important.
- The viscous drag (i.e. integration of the shear along the x-direction along the profile) is then computed to assess the overall model performance



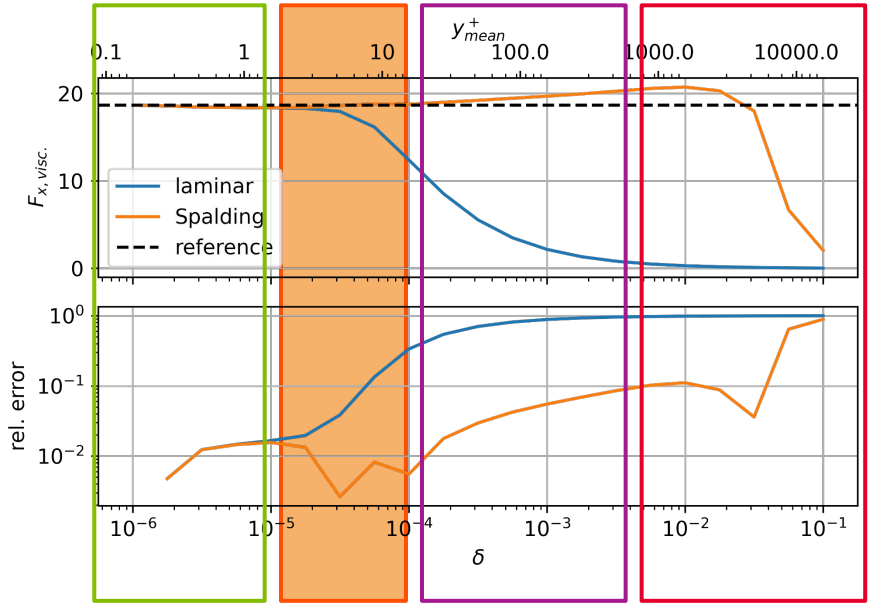


4 regions in terms of y^+:

- $y^+ < 1$: wall models are not needed
- $1 < y^+ < 20$: wall models perform well
- $20 < y^+ < 200\text{-}500$: wall models are not accurate enough
- $200\text{-}500 < y^+$: wall models cannot apply

AIRBUS

Goal



Can we use a learning approach to **build a wall model that extends the orange zone** where the wall model performs well?

… that may be dependent on the turbulence model

... and is consistent with $y^+=u^+$ when $y^+ < 1$

… and is usable in a 3D CFD code

AIRBUS

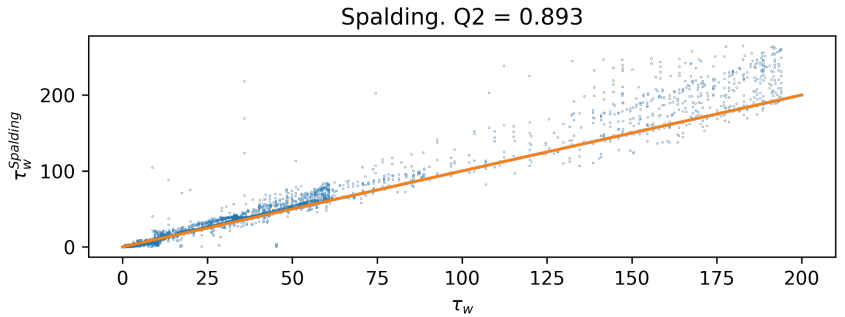
Metamodels for subsonic flows

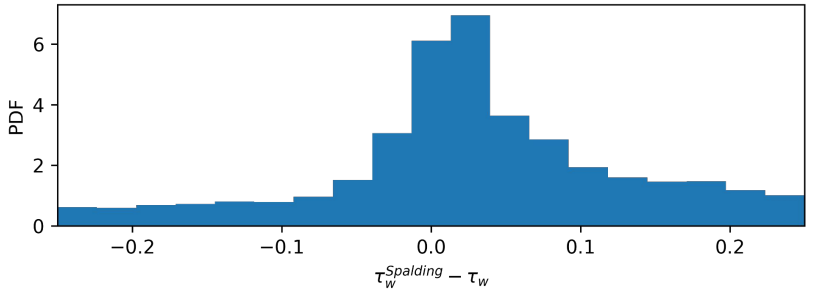
Reference : classical wall model (Spalding)

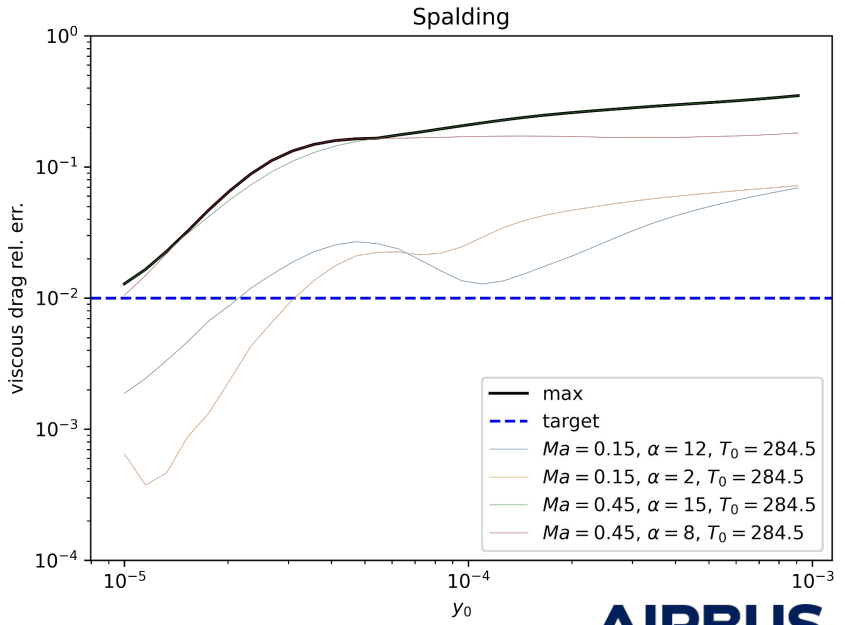
- Analytical model:

$$y^+ = u^+ + \exp(-\kappa.B)[\exp(\kappa.u^+) - 1 - (\kappa.u^+)^2/2 - (\kappa.u^+)^3/6]$$

- Default choice of parameters from flat-plate boundary layer measurements (κ=0.41, B=5.0)
- Inputs :
  - Distance to the wall δ
  - Tangential velocity U_t
  - Viscosity v (nearly constant here)
- Evaluation in 3.10^{-7} s / point







AIRBUS

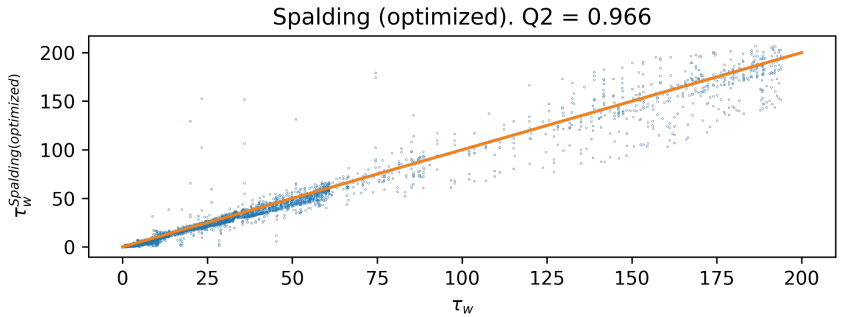
Reference : classical wall model (Spalding)

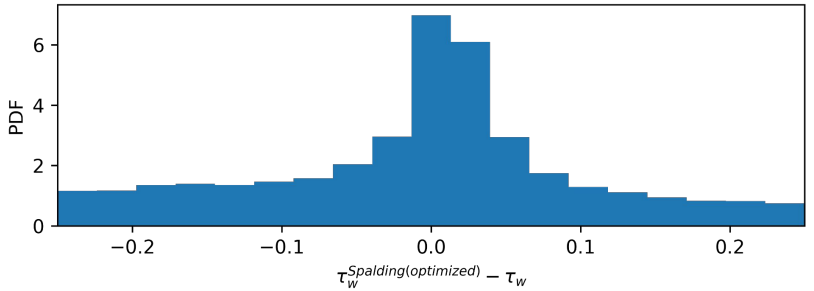
Optimization of the model parameters

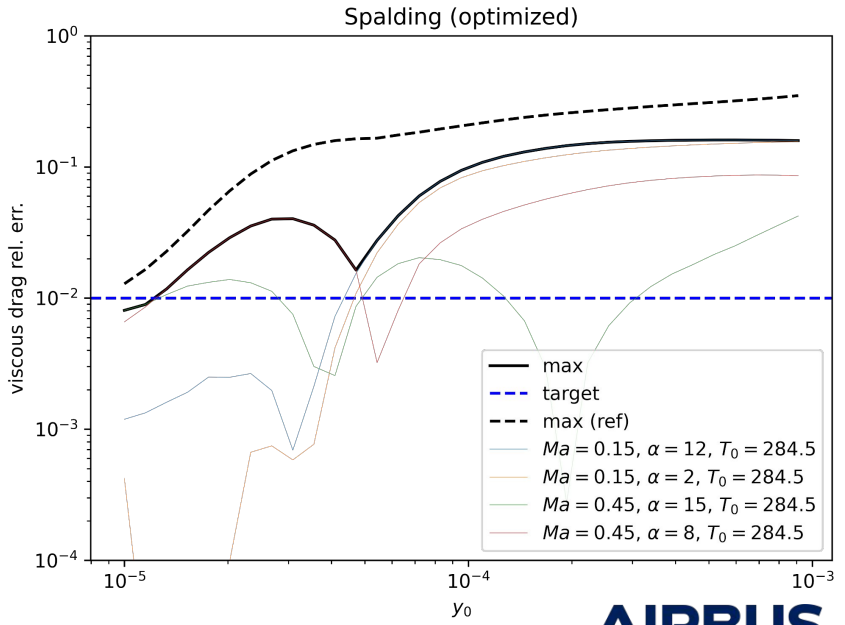
- Analytical model:

$$y^+ = u^+ + \exp(-\kappa.B)[\exp(\kappa.u^+) - 1 - (\kappa.u^+)^2/2 - (\kappa.u^+)^3/6]$$

- Optimisation of κ and B on the training data
- (dashed lines : reference model)







AIRBUS

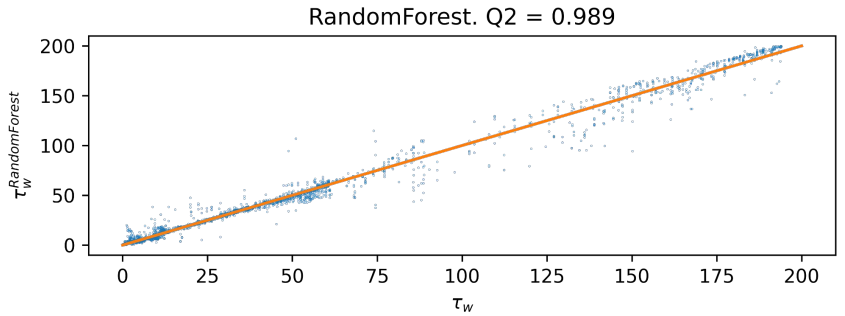
Metamodel #1: random forest

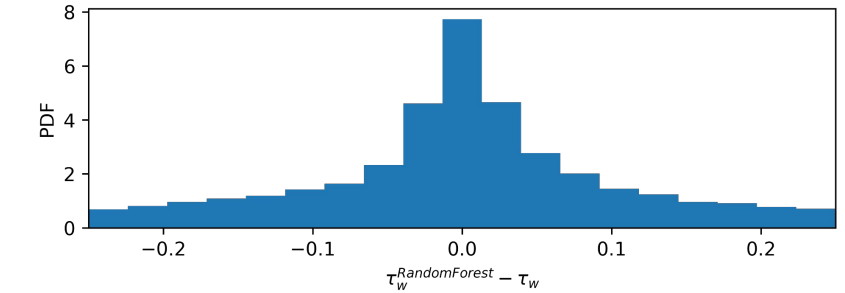
Taken from scikit-learn

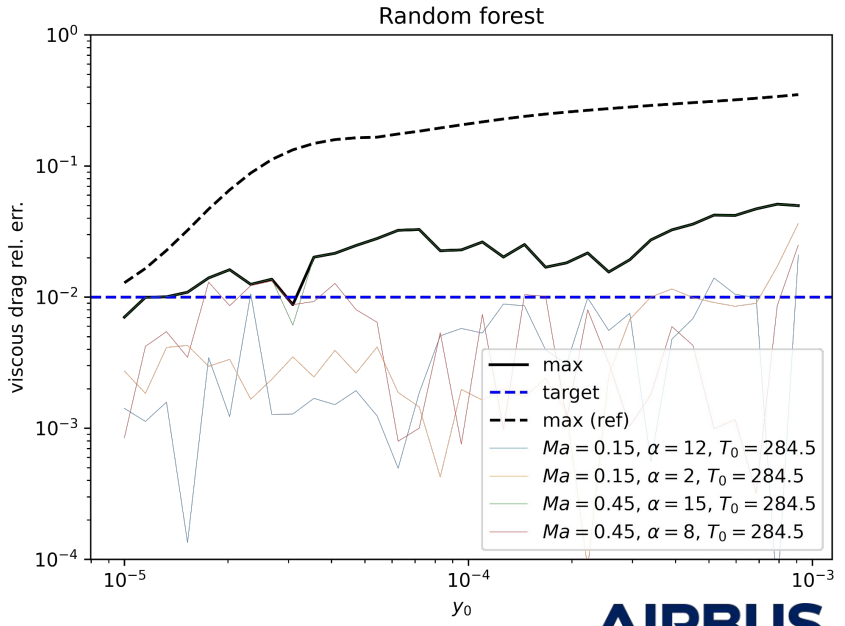
- Inputs:
  - Distance to the wall δ
  - Tangential velocity U_t
  - Turbulent kinetic energy k
  - Tangential pressure gradient ∇P.t
- We got accurate models while fixing max_depth & min_samples_split and increasing number of trees!

```
rf = RandomForestRegressor(max_depth=None, random_state=3,
                           bootstrap=True, min_samples_split=5,
                           max_samples=len(X_train), n_estimators=500,
                           n_jobs=-1)
algo = Pipeline([('scaler', MinMaxScaler()), ('rf',  rf)])
algo.fit(X_train, y_train)
```

- Evaluation in ~3.10^{-6} s / point







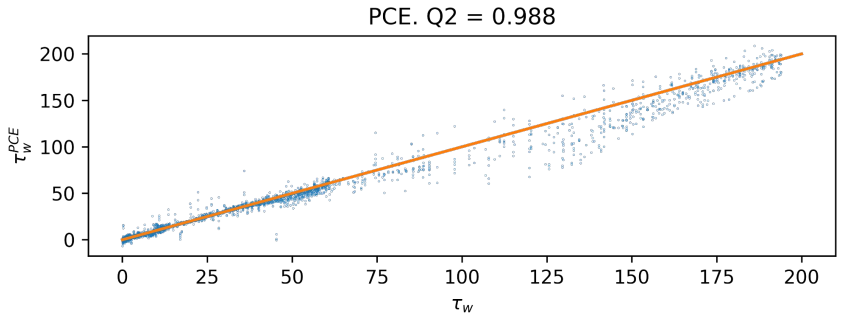
AIRBUS

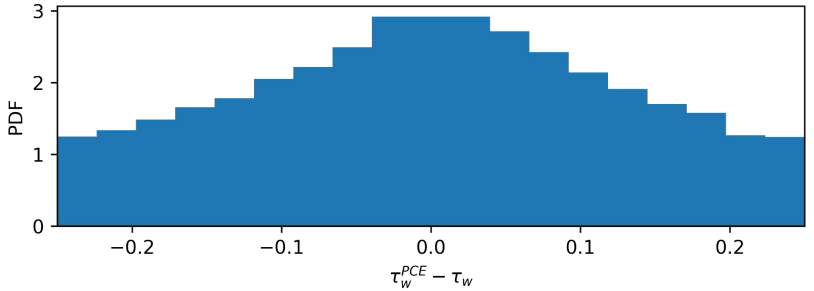
Metamodel #1: PCE

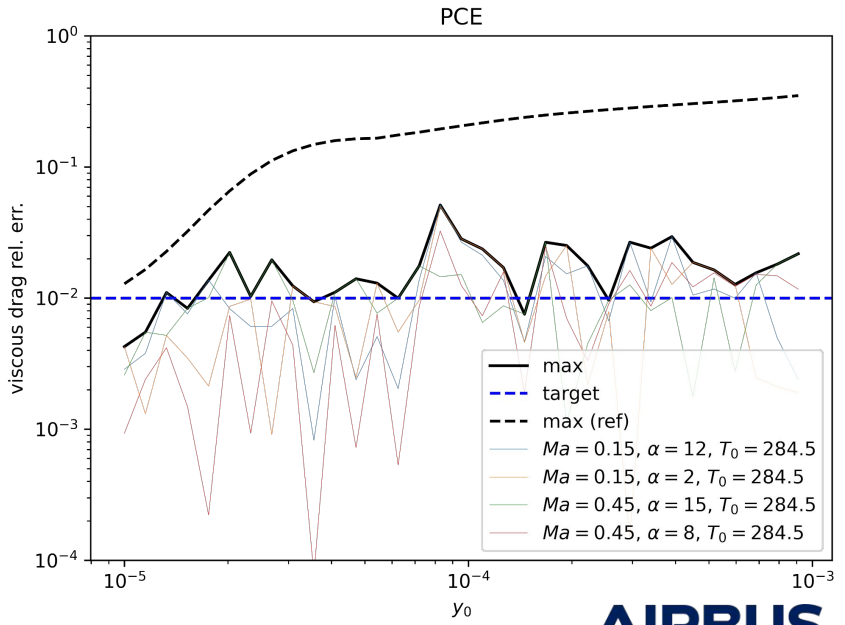
- Same inputs as for RF
- Settings:
  - Non parametric estimation of the input distribution D using histograms for the margins and the independent copula
  - D is mapped into the uniform distribution over $[-1, 1]^4$ using the isoprobabilistic transformation T
  - Expansion in terms of multivariate tensorized Legendre polynomials $(P_i)_{i \in I}$ in the uniform space using a sparsity inducing weighted hyperbolic norm and a weighted quadratic loss
  - The meta-model reads :

$$\tau_w = \sum_{i \in I} P_i(T(\delta, U_t, \partial P_t, K))$$

  with $I \subset N$ and $\#I = 500$.
- Evaluation in ~1.10^{-4} s / point







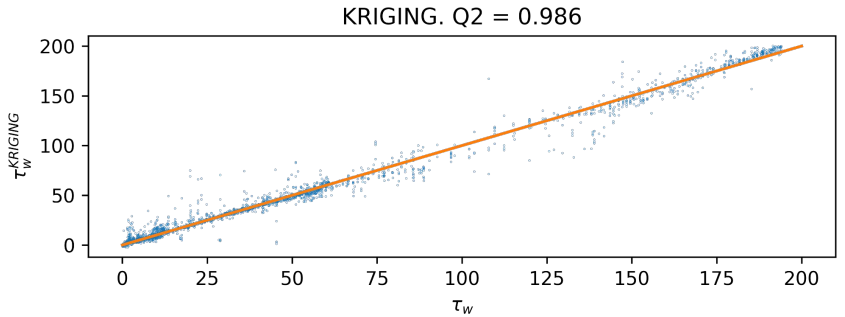
AIRBUS

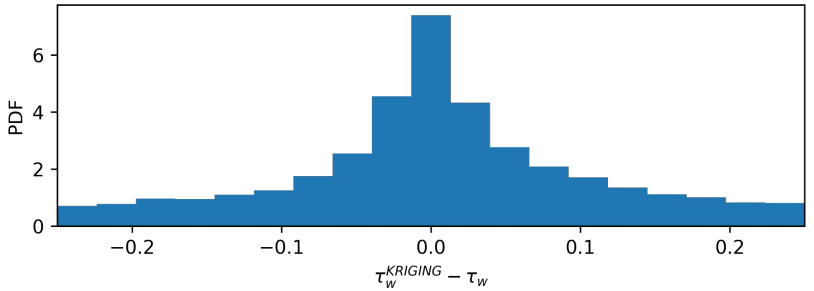
Kriging

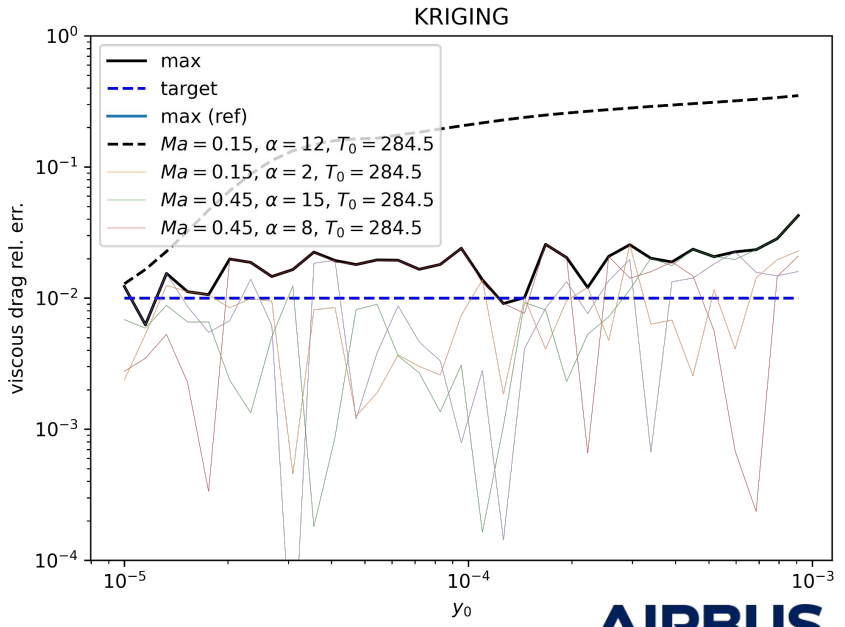
- Same inputs as for RF
- Settings:
  - We use a Matèrn(1/2) kernel K_θ and a linear basis x → a + b.x for the trend
  - The hyperparameter θ is estimated using its maximum likelihood estimator θ with multi-start cobyla optimizer (20 QMC starting points in $[0.1, 10.0]^4$), based on 5% of the learning database X.
  - The kriging model is then built using the full learning database
  - The meta-model reads

$$\tau_w = a + b.x + \sum_{x_i \in \mathcal{X}} w_i K_{\hat{\theta}}(x, x_i)$$

  with X=(δ, Ut, ∂P_t, K)
- Evaluation in ~1.10^{-3} s / point



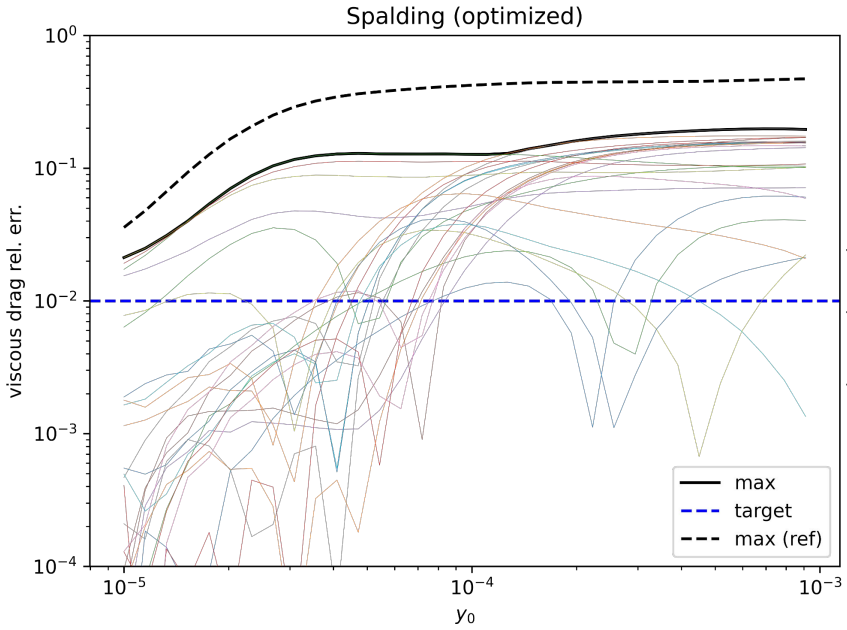


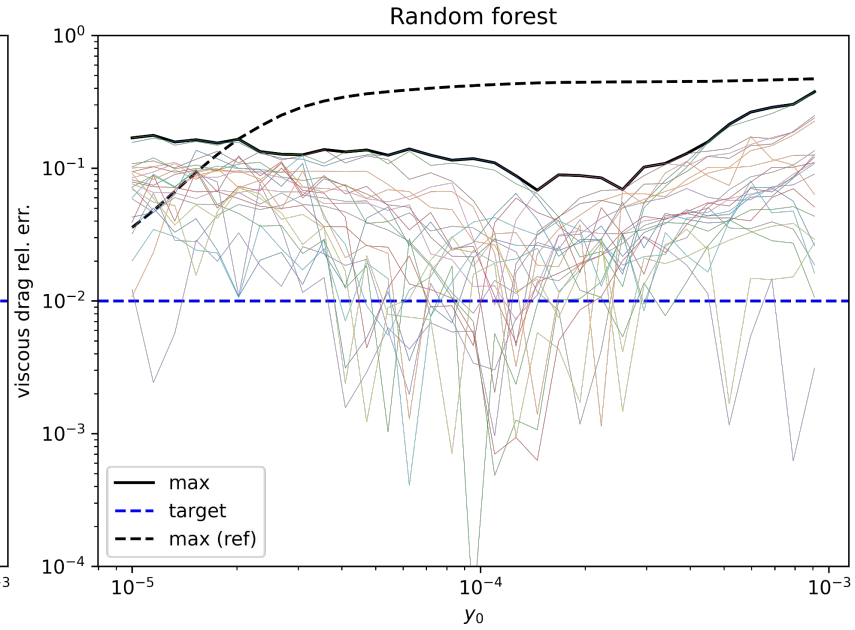


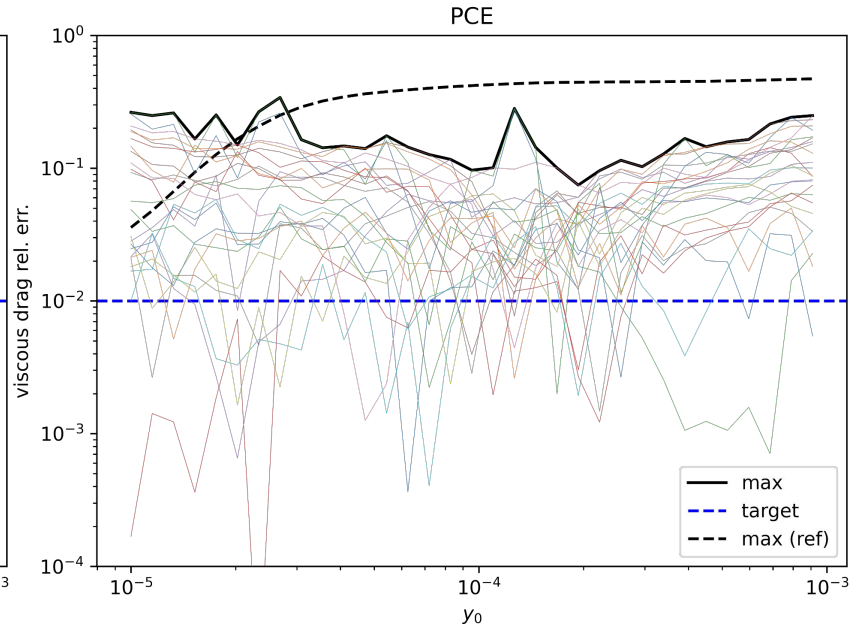
AIRBUS

What about transonic flows?

The Mach number is added to the input variables







AIRBUS

Conclusions & outlook

AIRBUS

Conclusions and outlook

- Conclusions
  - Using learning models it is possible to derive wall models that significantly improve viscous drag predictions
  - Work is still needed to have metamodel usable for transonic flows

- Outlook
  - Find suitable metamodels for transonic configurations
  - Predict not only the drag but also the heat transfer
  - Implement (some of) these models in 3D CFD codes

Thank you

AIRBUS