Machine Learning for wall modeling in LES of separating flows

HiFiLeD, 3rd High-Fidelity Industrial LES/DNS Symposium,
December 14th-16th 2022, Brussels, Belgium

M. Boxho, M. Rasquin, T. Toulorge, G. Dergham, G. Winckelmans and K. Hillewaert
ULiege, UCLouvain, Cenaero, Safran Tech
Contact: margaux.boxho@cenaero.be

Industrial context

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions** but remains costly at large Reynolds numbers.

Wall-models reduces the computational cost by modeling the near-wall energetic scales.

PROD-F-015-02

Cenaero

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions** but remains costly at large Reynolds numbers.

Wall-models reduces the computational cost by modeling the near-wall energetic scales.

Simulations of turbomachines to compute the **operating points** of the engine.

Calibration of the wall models using **neural networks** with **wrLES** on basic and academic test cases and blades.

In turbomachines, transitional flows are frequently encountered, such as laminar **separation bubbles**.

Industrial context

RANS frequently fails at **off-design conditions** due to its inherent modeling assumptions.

LES reduces the **modeling assumptions** but remains costly at large Reynolds numbers.

Wall-models reduces the computational cost by modeling the near-wall energetic scales.

Feed lower-fidelity models on larger components (wmLES)

Simulations of turbomachines to compute the **operating points** of the engine.

Calibration of the wall models using **neural networks** with **wrLES** on basic and academic test cases and blades.

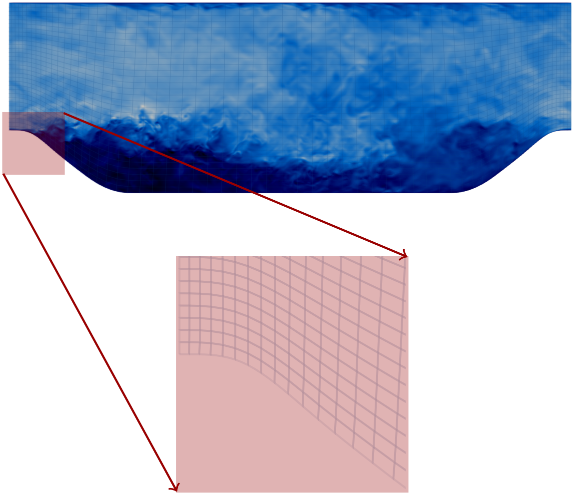
In turbomachines, transitional flows are frequently encountered, such as laminar **separation bubbles**.

Increasing fidelity on sub-components (wrLES,DNS)

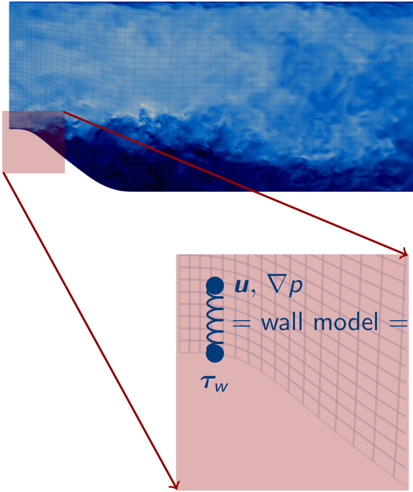
PROD-F-015-02

Cenaero

Wall modeled LES with neural networks

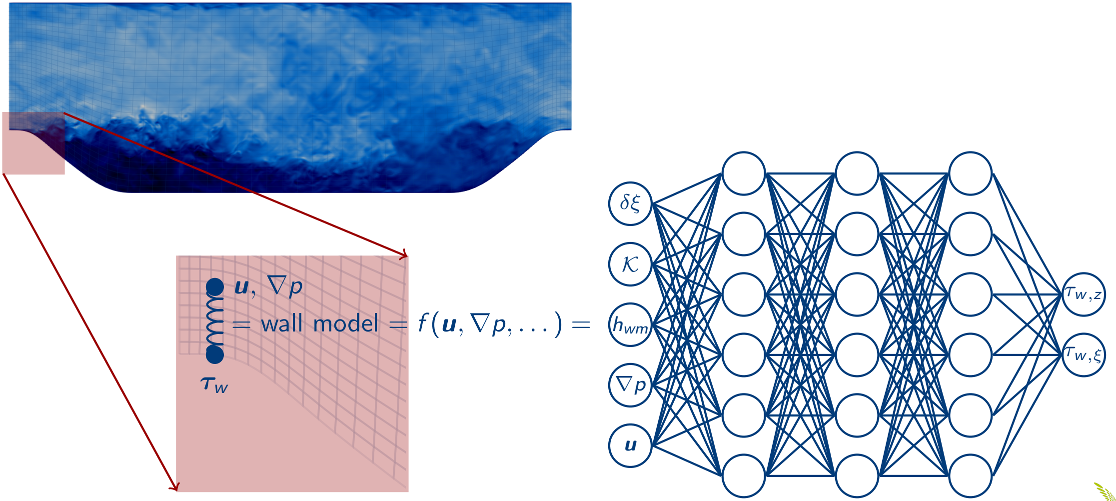


Cenaero

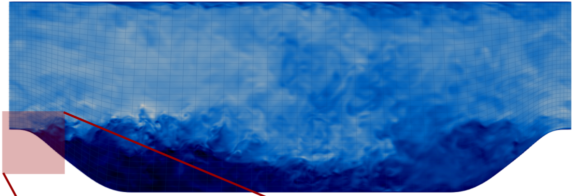


$$= \text{wall model} = f(\boldsymbol{u}, \nabla p, \ldots) =$$

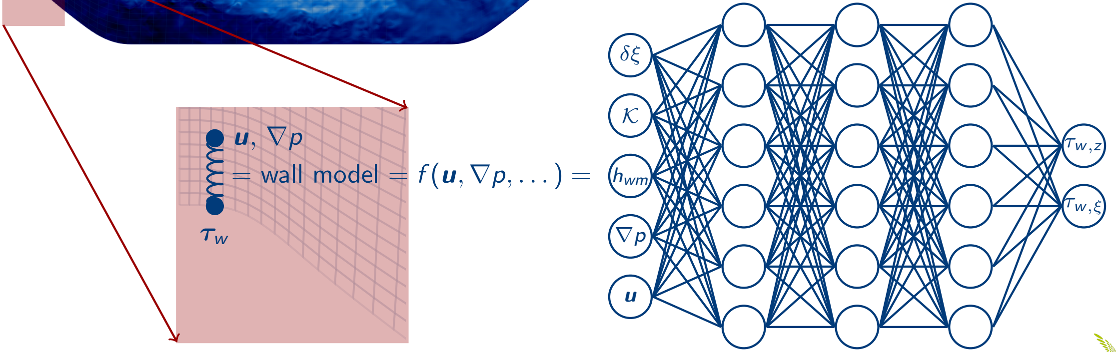
Wall modeled LES with neural networks



Wall modeled LES with neural networks

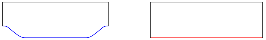


Problem definition Finding a complex and dynamic relation between instantaneous fields, geometrical parameters, and the wall-parallel components of the wall shear stress

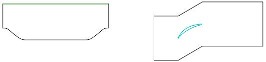


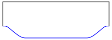
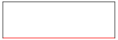
PROD-F-015-02

Cenaero



[1] Generate **databases** from **basic**, **academic**, and **blades** test cases

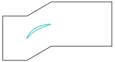


[1] Generate **databases** from **basic**, **academic**, and **blades** test cases

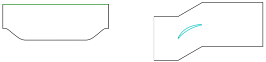
→

[2] Normalize the input/output pairs to get a uniform database for the training

Cenaero

[1] Generate **databases** from **basic**, **academic**, and **blades** test cases

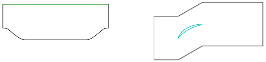


→ **[2] Normalize** the input/output pairs to get a uniform database for the training

→ **[3] Train** neural networks (CNN and MDN) for the prediction of the wall shear stress

PROD-F-015-02

[1] Generate **databases** from **basic**, **academic**, and **blades** test cases

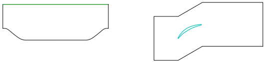


→ **[2] Normalize** the input/output pairs to get a uniform database for the training

→ **[3] Train** neural networks (CNN and MDN) for the prediction of the wall shear stress

↓ **[4] A priori** validation (i.e., mean, variance, power density spectrum, instantaneous field, ...)

[1] Generate **databases** from **basic**, **academic**, and **blades** test cases



[2] Normalize the input/output pairs to get a uniform database for the training

PyTorch

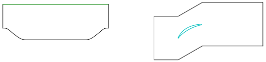
[3] Train neural networks (CNN and MDN) for the prediction of the wall shear stress

[4] A priori validation (i.e., mean, variance, power density spectrum, instantaneous field, ...)

[5] Implementation of the data-driven wall model in a high-order flow solver (i.e., Argo-DG)

```
#include <torch/script.h>
```

[1] Generate **databases** from **basic**, **academic**, and **blades** test cases



PyTorch

[2] Normalize the input/output pairs to get a uniform database for the training

[3] Train neural networks (CNN and MDN) for the prediction of the wall shear stress

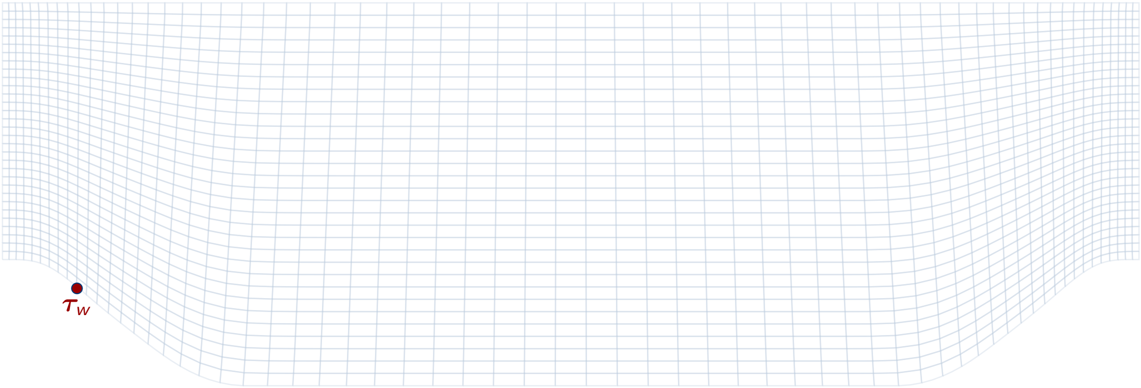
[4] A priori validation (i.e., mean, variance, power density spectrum, instantaneous field, ...)

[5] Implementation of the data-driven wall model in a high-order flow solver (i.e., Argo-DG)

```
#include <torch/script.h>
```

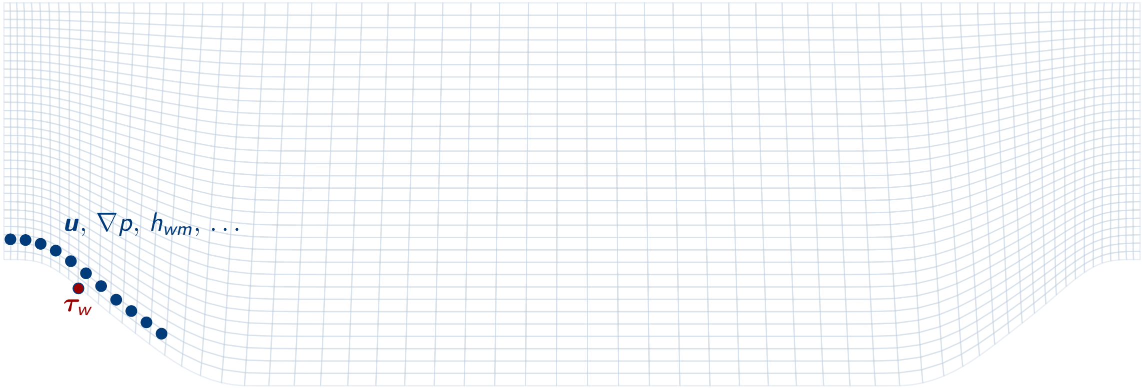
[6] A posteriori validation on the two-dimensional periodic hill at $Re_b = 10{,}595$

Where is the data extracted?



Cenaero

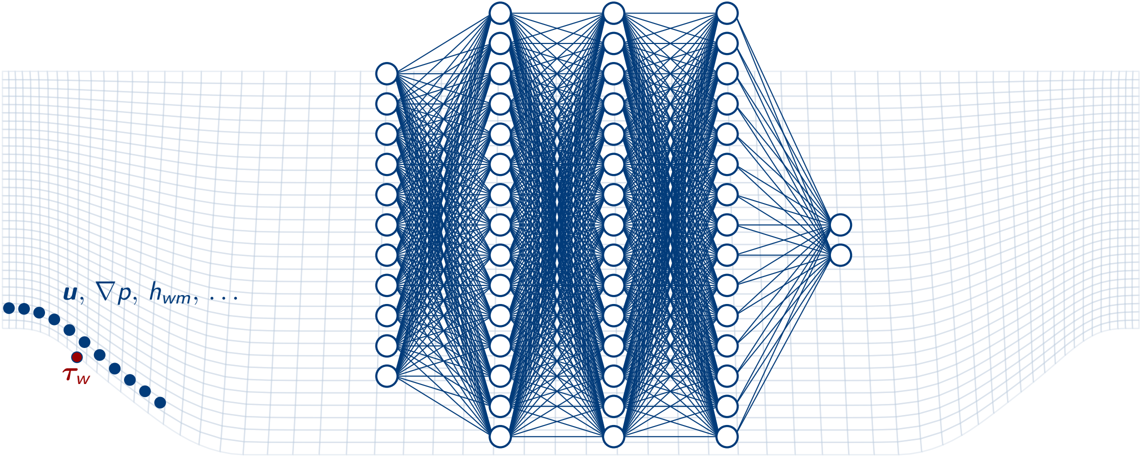
Where is the data extracted?



PROD-F-015-02

Cenaero

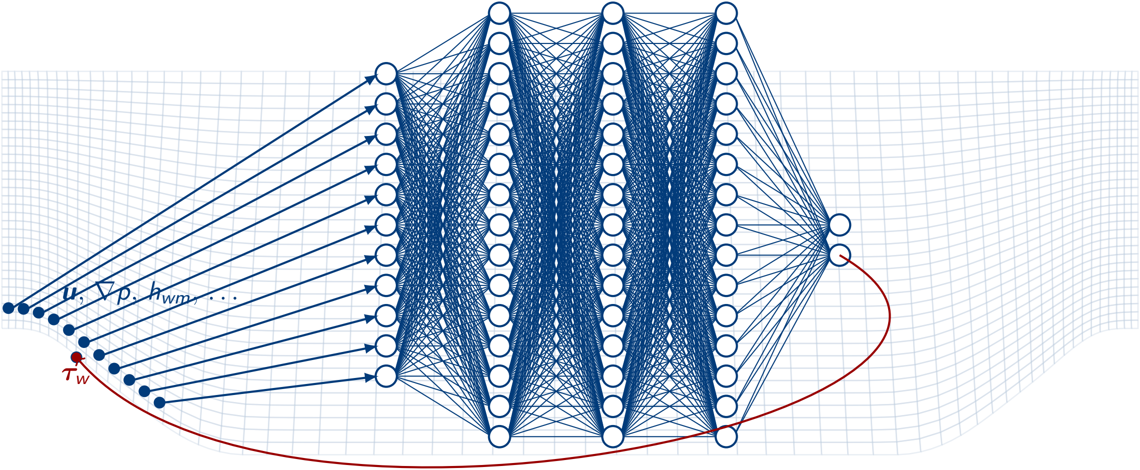
Where is the data extracted?



PROD-F-015-02

Where is the data extracted?

The input size is based on the analysis of space-time correlations[1]



[1] Boxho, M. *et al.*: Analysis of space-time correlations to support the development of wall-modeled LES. Flow, Turbulence and Combustion, 2022.

PROD-F-015-02

The **inputs** are normalized as,

$$\boldsymbol{u}^\star = \frac{\boldsymbol{u}}{u_{\nu,p}}$$

$$(\nabla p)^\star = \frac{h_{wm}}{(\rho u_{\nu,p}^2)} \nabla p$$

$$h_{wm}^\star = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right)$$

$$(\delta\xi)^* = \frac{\delta\xi}{y_{\nu,p}}$$

Normalization of the input/output pairs

The **inputs** are normalized as,

$$\boldsymbol{u}^\star = \frac{\boldsymbol{u}}{u_{\nu,p}}$$

$$(\nabla p)^\star = \frac{h_{wm}}{(\rho u_{\nu,p}^2)} \nabla p$$

$$h_{wm}^\star = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right)$$

$$(\delta\xi)^* = \frac{\delta\xi}{y_{\nu,p}}$$

$$u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}$$

PROD-F-015-02

Cenaero

Normalization of the input/output pairs

The **inputs** are normalized as,

$$\boldsymbol{u}^{\star} = \frac{\boldsymbol{u}}{u_{\nu,p}}$$

$$(\nabla p)^{\star} = \frac{h_{wm}}{(\rho u_{\nu,p}^2)} \nabla p$$

$$h_{wm}^{\star} = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right)$$

$$(\delta\xi)^{*} = \frac{\delta\xi}{y_{\nu,p}}$$

$$u_{\nu,p} = \sqrt{u_{\nu}^2 + u_p^2}$$

$$u_{\nu} = \sqrt{\frac{\nu \| u_{\|} \|}{h_{wm}}}$$

$$u_p = \left| \frac{\nu}{\rho} \| \nabla p_{\|} \| \right|^{1/3}$$

PROD-F-015-02

Cenaero

The **inputs** are normalized as,

$$\boldsymbol{u}^\star = \frac{\boldsymbol{u}}{u_{\nu,p}}$$

$$(\nabla p)^\star = \frac{h_{wm}}{(\rho u_{\nu,p}^2)} \nabla p$$

$$h_{wm}^\star = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right)$$

$$(\delta\xi)^* = \frac{\delta\xi}{y_{\nu,p}}$$

$$u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}$$

$$y_{\nu,p} = \frac{\nu}{u_{\nu,p}}$$

$$u_\nu = \sqrt{\frac{\nu \| u_{||} \|}{h_{wm}}}$$

$$u_p = \left| \frac{\nu}{\rho} \| \nabla p_{||} \| \right|^{1/3}$$

PROD-F-015-02

Normalization of the input/output pairs

The **inputs** are normalized as,

$$\boldsymbol{u}^\star = \frac{\boldsymbol{u}}{u_{\nu,p}}$$

$$(\nabla p)^\star = \frac{h_{wm}}{\left(\rho u_{\nu,p}^2\right)} \nabla p$$

$$h_{wm}^\star = \ln\left(\frac{h_{wm}}{y_{\nu,p}}\right)$$

$$(\delta\xi)^* = \frac{\delta\xi}{y_{\nu,p}}$$

$$u_{\nu,p} = \sqrt{u_\nu^2 + u_p^2}$$

$$y_{\nu,p} = \frac{\nu}{u_{\nu,p}}$$

$$u_\nu = \sqrt{\frac{\nu \| u_{\|} \|}{h_{wm}}}$$

$$u_p = \left| \frac{\nu}{\rho} \| \nabla p_{\|} \| \right|^{1/3}$$

The **outputs** are normalized as,

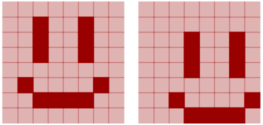
$$\boldsymbol{\tau}_w^\star = \frac{\boldsymbol{\tau}_w}{\frac{1}{2}\rho \langle u_{\nu,p}^2 \rangle_{\xi,z}}$$

Neural networks: Properties and architecture

Convolutional neural network (CNN)

Prop.

Invariant to **translation**



Mixture Density Network (MDN)

Produces a **distribution** as a linear combination of Gaussian distribution

$$p(y|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y; \mu_k, \sigma_k^2)$$

PROD-F-015-02

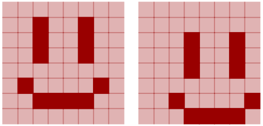
Cenaero

Neural networks: Properties and architecture

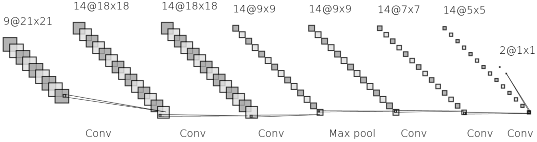
Convolutional neural network (CNN)

Prop.

Invariant to **translation**



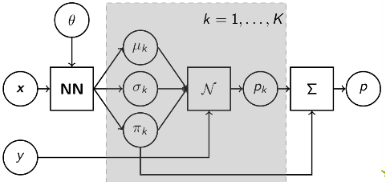
Archit.

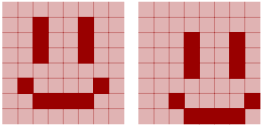
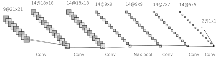
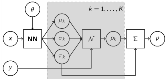


Mixture Density Network (MDN)

Produces a **distribution** as a linear combination of Gaussian distribution

$$p(y|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y; \mu_k, \sigma_k^2)$$

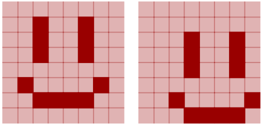


	Convolutional neural network (CNN)	Mixture Density Network (MDN)
Prop.	Invariant to **translation** 	Produces a **distribution** as a linear combination of Gaussian distribution $$p(y\|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(y; \mu_k, \sigma_k^2)$$
Archit.		
Obj. fun.	$$\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} (y_i - \hat{y}_i)^2$$	$$\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \frac{(y_i - \mu(\boldsymbol{x}_i))^2}{2\sigma(\boldsymbol{x}_i)} + \log(\sigma(\boldsymbol{x}_i)) + C$$

Convolutional neural network (CNN)

Prop.

Invariant to **translation**



Archit.

Obj. fun.

$$\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} (y_i - \hat{y}_i)^2$$

Mixture Density Network (MDN)

Produces a **distribution** as a linear combination of Gaussian distribution

p … $_k, \sigma_k^2)$

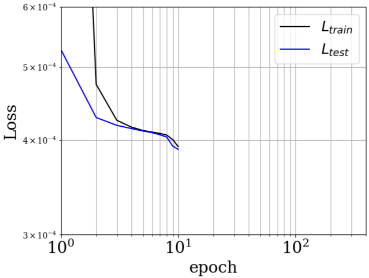
The prediction is obtained from a **sampling** of the generated distribution.

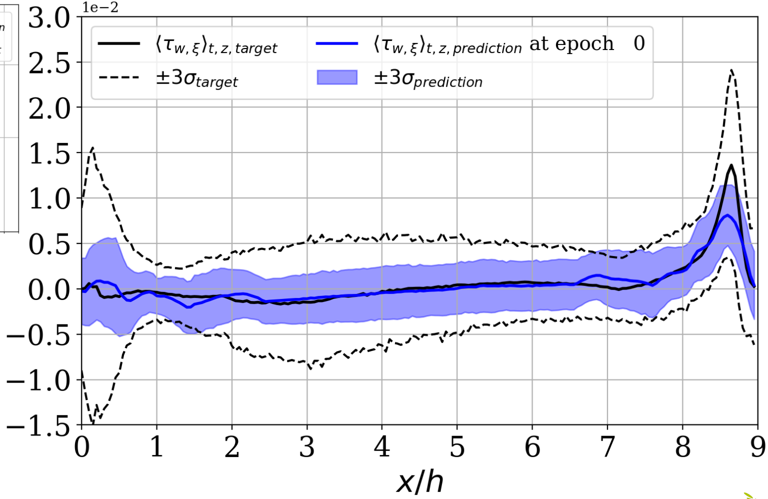
$$\arg\min_{\theta} \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \frac{(y_i - \mu(\boldsymbol{x}_i))^2}{2\sigma(\boldsymbol{x}_i)} + \log(\sigma(\boldsymbol{x}_i)) + C$$

PROD-F-015-02

Cenaero

Training a CNN-1d-SAL on the periodic hill $Re_b = 10{,}595$



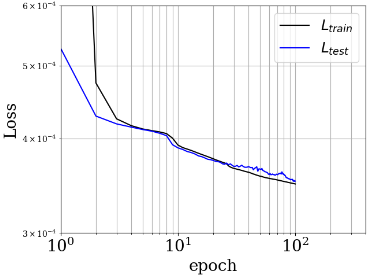


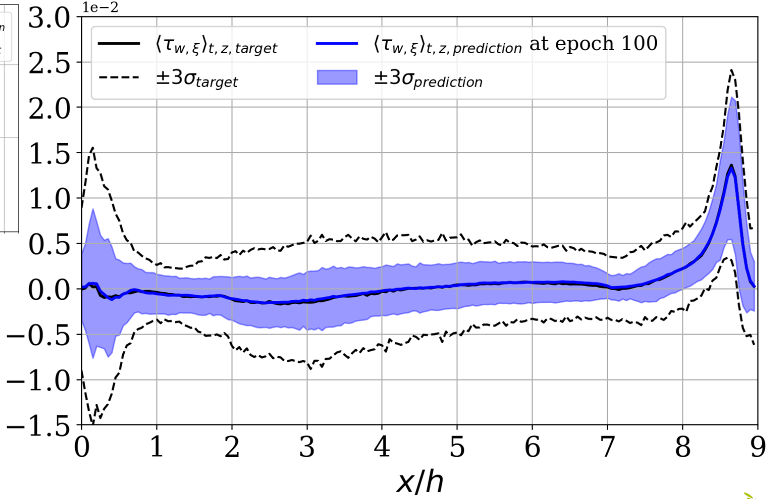
#Parameters: 6,440
Learning rate: 0.001
Batch size: 1024
Database size: 2,736,000
Training time: 5h13min
#Epoch: 400

PROD-F-015-02

Cenaero

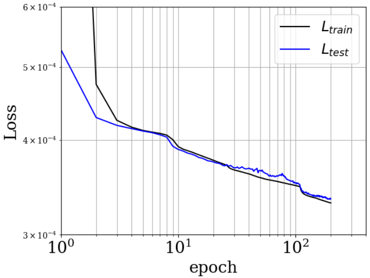
Training a CNN-1d-SAL on the periodic hill $Re_b = 10{,}595$

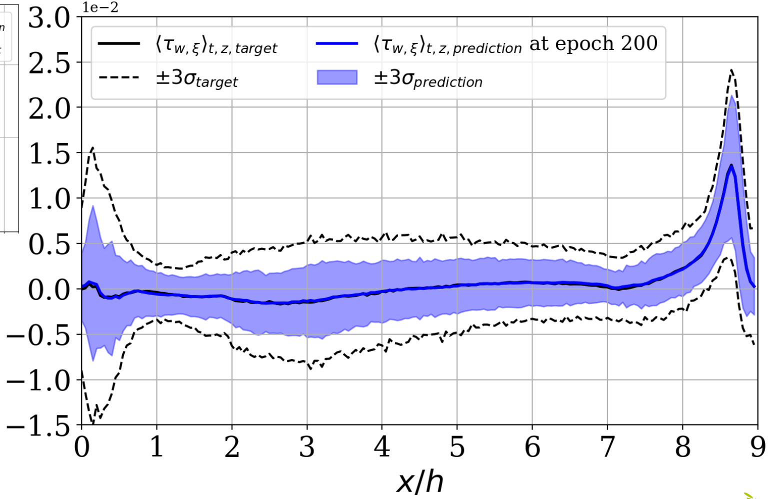




#Parameters: 6,440
Learning rate: 0.001
Batch size: 1024
Database size: 2,736,000
Training time: 5h13min
#Epoch: 400

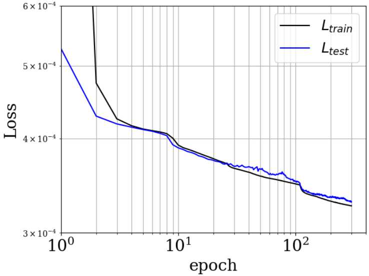
Training a CNN-1d-SAL on the periodic hill $Re_b = 10{,}595$

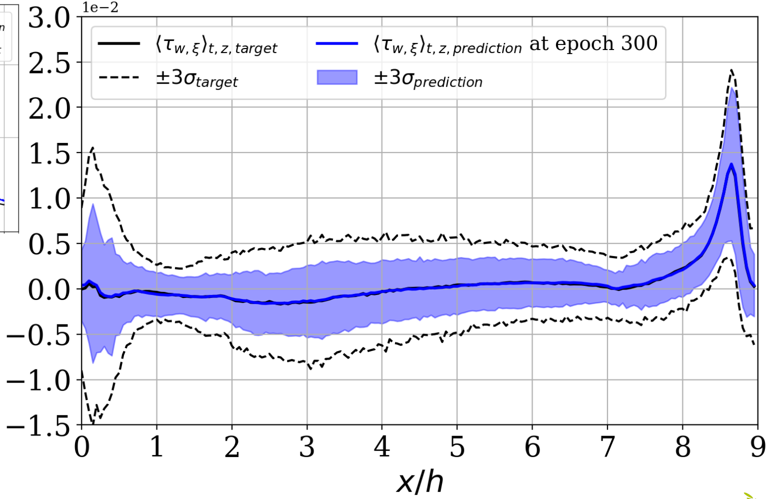




#Parameters: 6,440
Learning rate: 0.001
Batch size: 1024
Database size: 2,736,000
Training time: 5h13min
#Epoch: 400

Training a CNN-1d-SAL on the periodic hill $Re_b = 10{,}595$



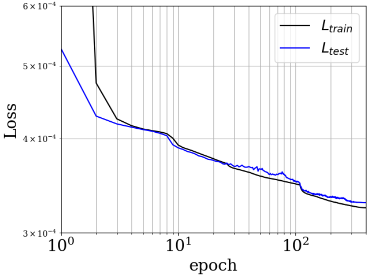


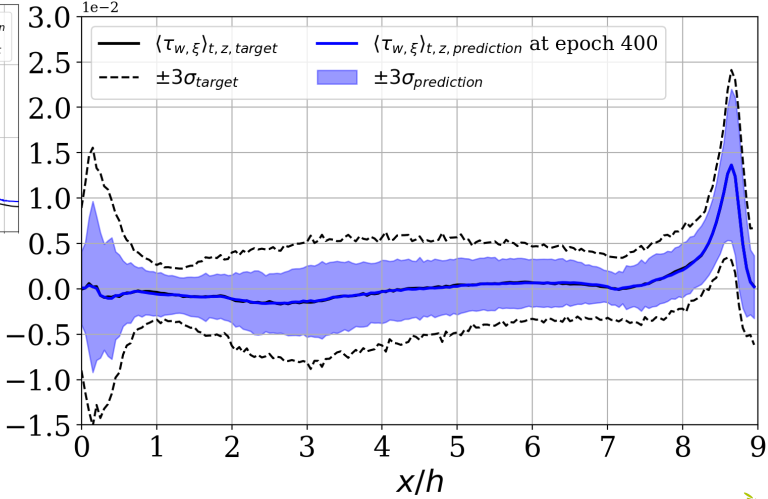
#Parameters: 6,440
Learning rate: 0.001
Batch size: 1024
Database size: 2,736,000
Training time: 5h13min
#Epoch: 400

PROD-F-015-02

Cenaero

Training a CNN-1d-SAL on the periodic hill $Re_b = 10{,}595$





#Parameters: 6,440
Learning rate: 0.001
Batch size: 1024
Database size: 2,736,000
Training time: 5h13min
#Epoch: 400

PROD-F-015-02

Cenaero

A priori validation on the periodic hill $Re_b = 10{,}595$

Distribution of the wall shear stress

Instantaneous wall-shear stress field

Autocorrelations of the predicted structures

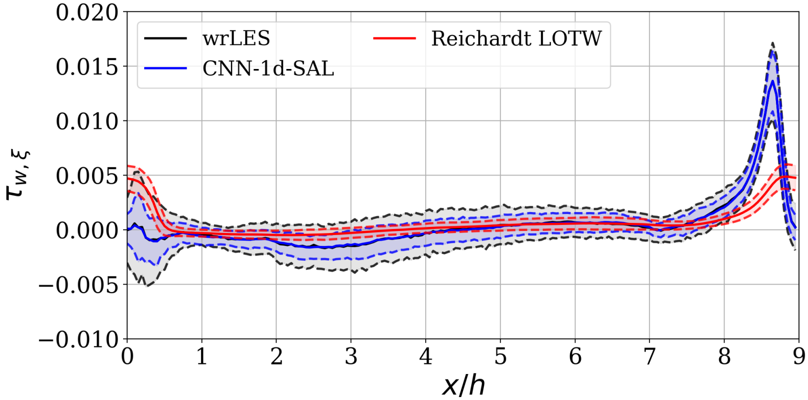
Power **Spectrum** Density (PSD)

PROD-F-015-02

Cenaero

A priori validation on the periodic hill $Re_b = 10{,}595$

Distribution of the wall shear stress

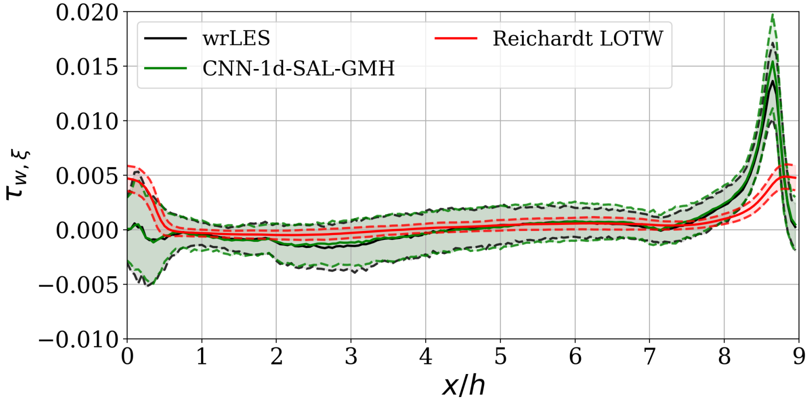


PROD-F-015-02

Cenaero

A priori validation on the periodic hill $Re_b = 10{,}595$

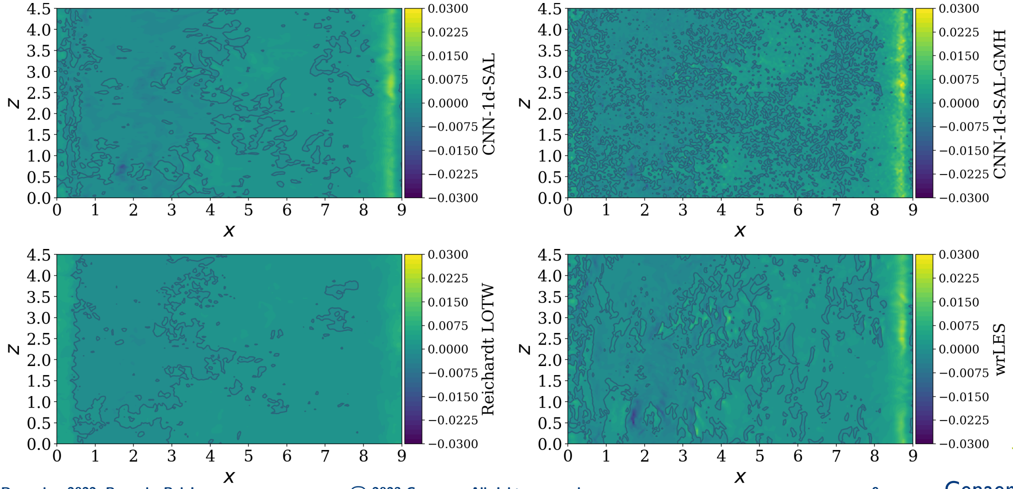
Distribution of the wall shear stress



PROD-F-015-02

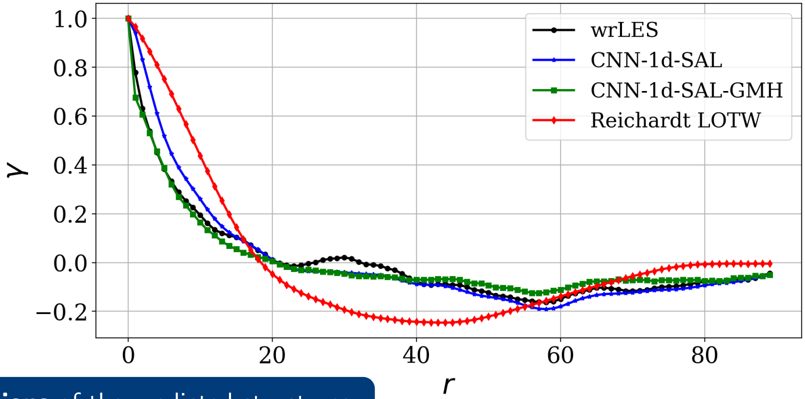
A priori validation on the periodic hill $Re_b = 10{,}595$

Instantaneous wall-shear stress field



PROD-F-015-02

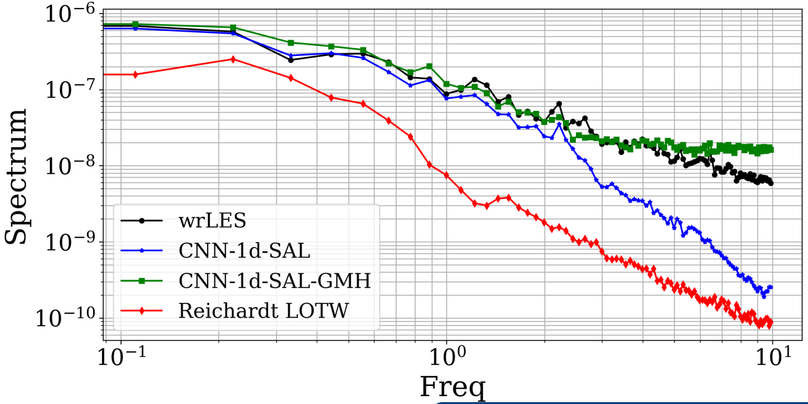
Cenaero



Autocorrelations of the predicted structures

Of $\tau_{w,\xi}$ at a given time t computed in the ξ-direction and averaged in the z-direction.

A priori validation on the periodic hill $Re_b = 10{,}595$



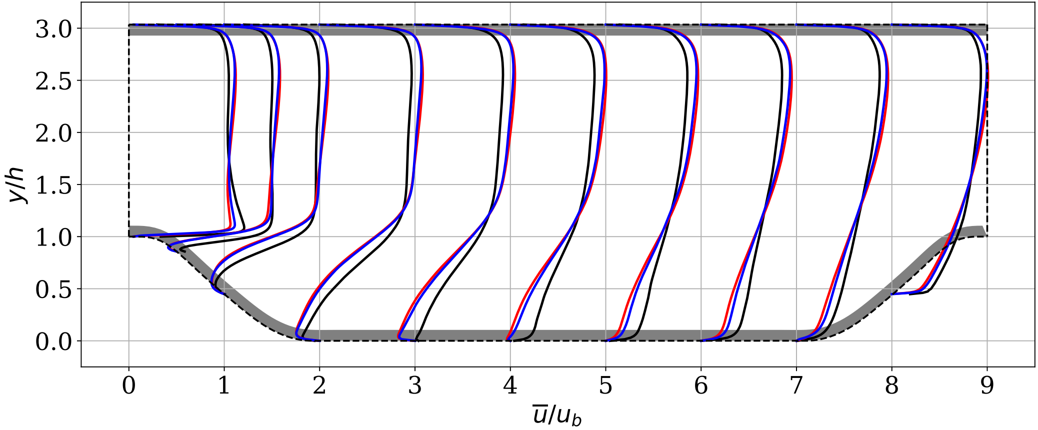
Power Density **Spectrum** (PDS)

Of $\tau_{w,\xi}$ at a given time t computed in the ξ-direction and averaged in the z-direction.

PROD-F-015-02

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

Velocity profiles averaged over 13.5 flow-through time:



PROD-F-015-02

Remark: The size of the recirculation bubble is underestimated and it impacts the whole domain.

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

How to correct the recirculation bubble location?

- Directional inconsistency [J. Bae *et al.*, 2022] between
  - the direction of the wall shear stress predicted by the network
  - the direction of the velocity measured at h_{wm}

PROD-F-015-02

Cenaero

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

How to correct the recirculation bubble location?

- Directional inconsistency [J. Bae *et al.*, 2022] between
  - the direction of the wall shear stress predicted by the network
  - the direction of the velocity measured at h_{wm}
- A quick fix in the code:

$$\boldsymbol{\tau}_w \cdot \boldsymbol{u} < 0 \Rightarrow \boldsymbol{\tau}_{\boldsymbol{w}} = \boldsymbol{0}$$

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

How to correct the recirculation bubble location?

- Directional inconsistency [J. Bae *et al.*, 2022] between
  - the direction of the wall shear stress predicted by the network
  - the direction of the velocity measured at h_{wm}
- A quick fix in the code:

$$\boldsymbol{\tau}_w \cdot \boldsymbol{u} < 0 \Rightarrow \boldsymbol{\tau_w} = \boldsymbol{0}$$

- Can also be added as an inequality constraint in the objective function while training the neural network as,

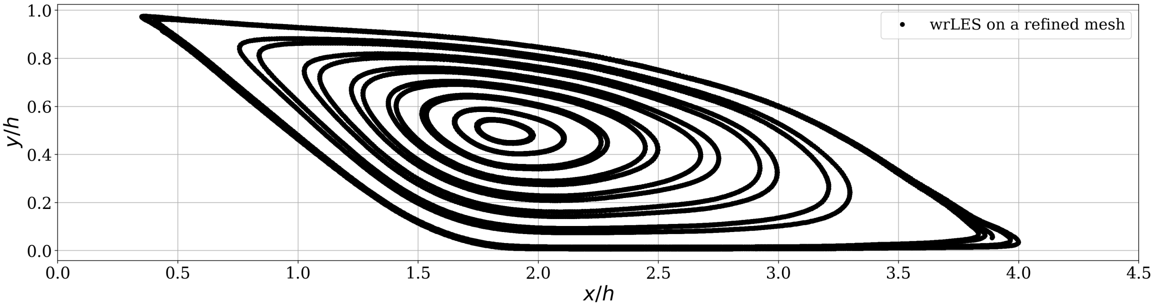
$$\mathcal{L} = \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \left[(\tau_{\xi,i} - \hat{\tau}_{\xi,i})^2 + (\tau_{z,i} - \hat{\tau}_{z,i})^2 \right] + \lambda \sum_{\boldsymbol{x}_i, y_i \in \boldsymbol{d}} \mathrm{ReLU}\left(-\boldsymbol{\tau}_i \cdot \boldsymbol{u}_i \right) \,,$$

  where λ is a certain hyper-parameter that penalizes more or less the directional inconsistency in the training dataset.

PROD-F-015-02

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

Observation of the **streamlines** in the recirculation bubble using the quick fix:

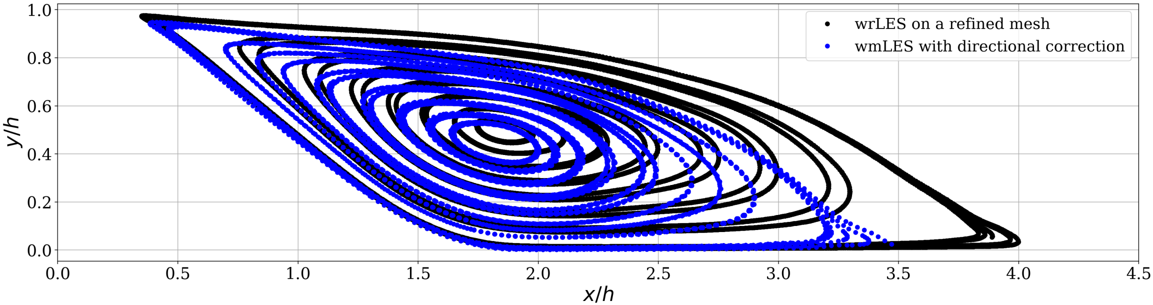


PROD-F-015-02

Cenaero

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

Observation of the **streamlines** in the recirculation bubble using the quick fix:

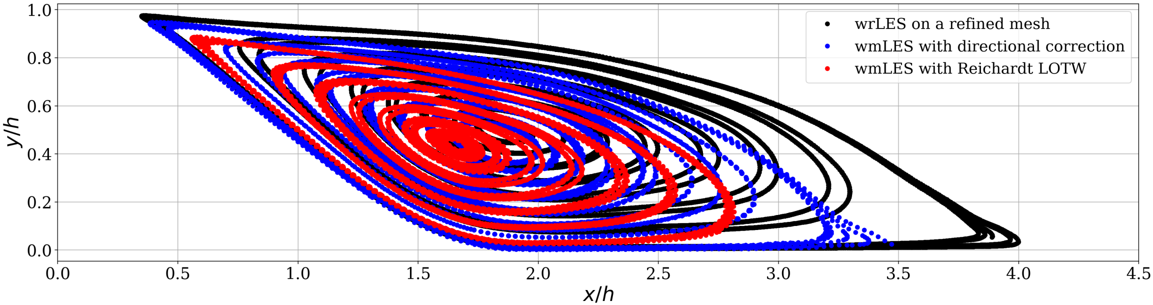


PROD-F-015-02

Cenaero

A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

Observation of the **streamlines** in the recirculation bubble using the quick fix:

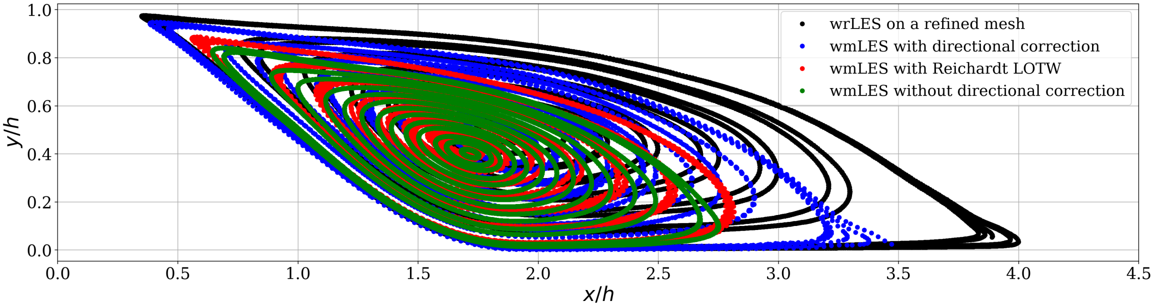


PROD-F-015-02

Cenaero

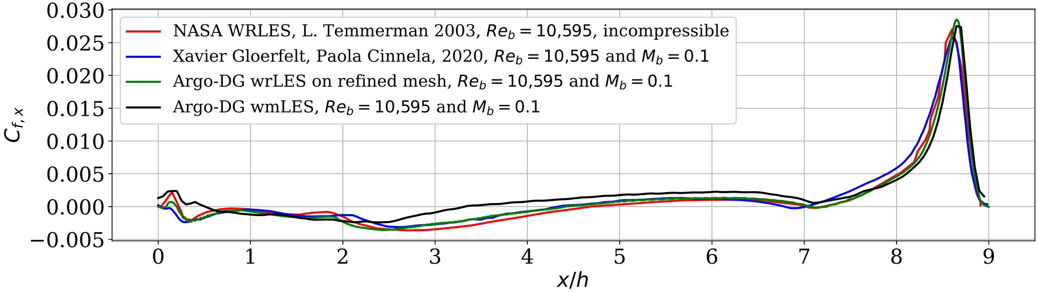
A posteriori validation on the CNN-1d-SAL for the periodic hill $Re_b = 10{,}595$

Observation of the **streamlines** in the recirculation bubble using the quick fix:



PROD-F-015-02

A posteriori validation on the lower wall of the periodic hill $Re_b = 10{,}595$

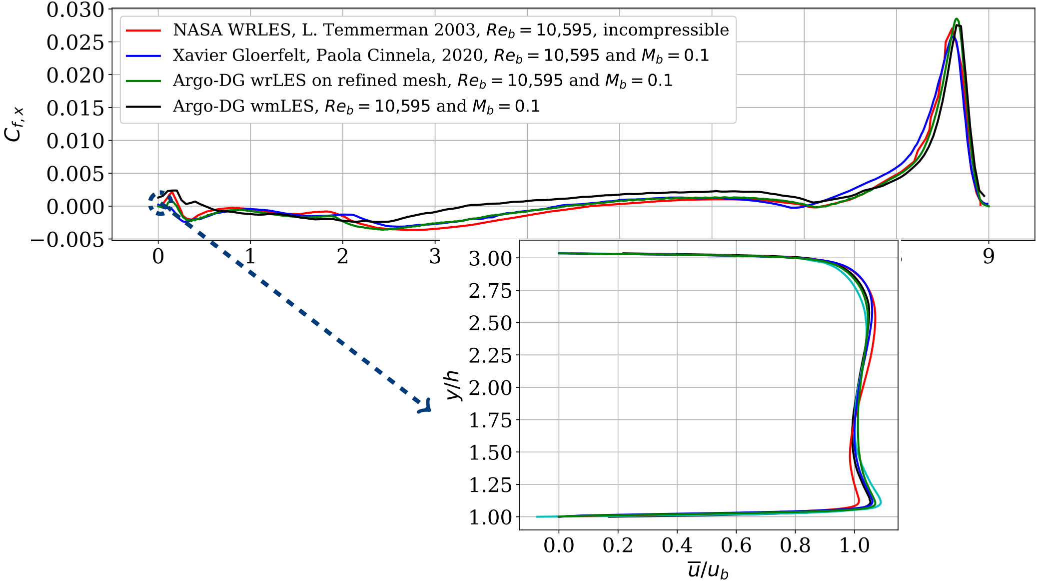


Mean velocity profiles at three x/h locations on the lower wall compared to the literature and a Reichardt LOTW (in cyan) computed with Argo-DG

PROD-F-015-02

Cenaero

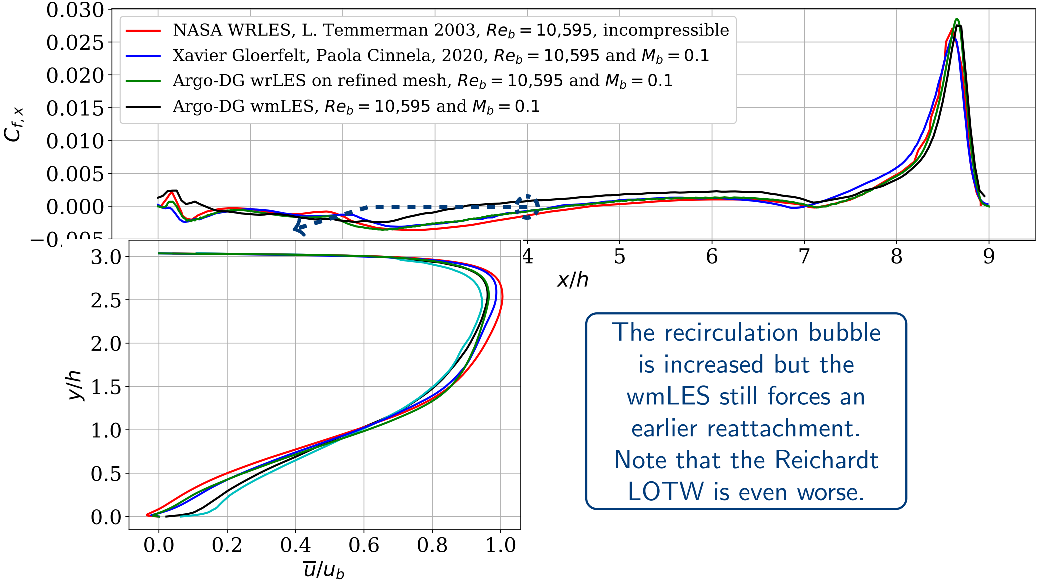
A posteriori validation on the lower wall of the periodic hill $Re_b = 10{,}595$



PROD-F-015-02

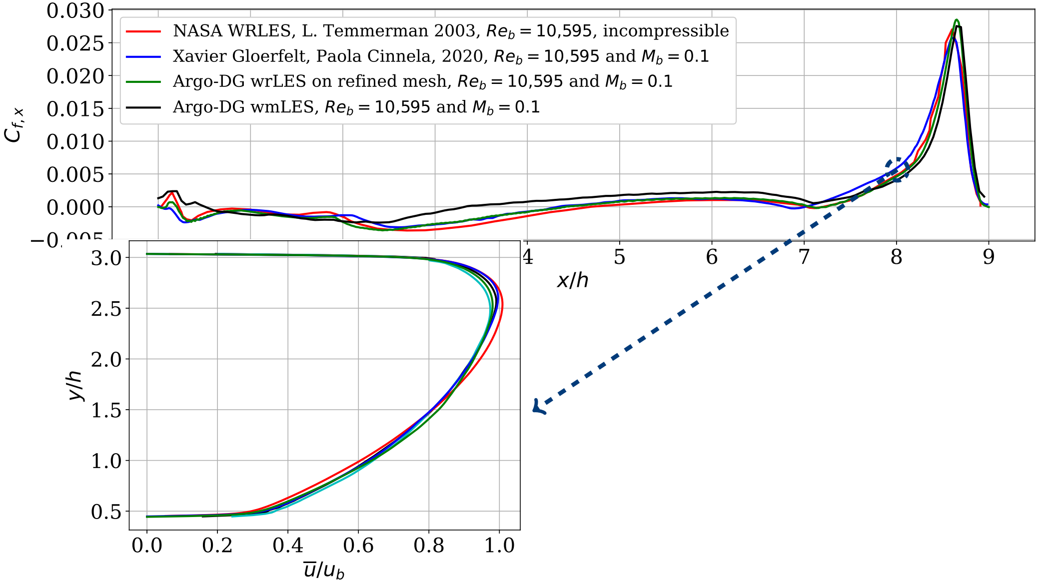
Cenaero

A posteriori validation on the lower wall of the periodic hill $Re_b = 10{,}595$

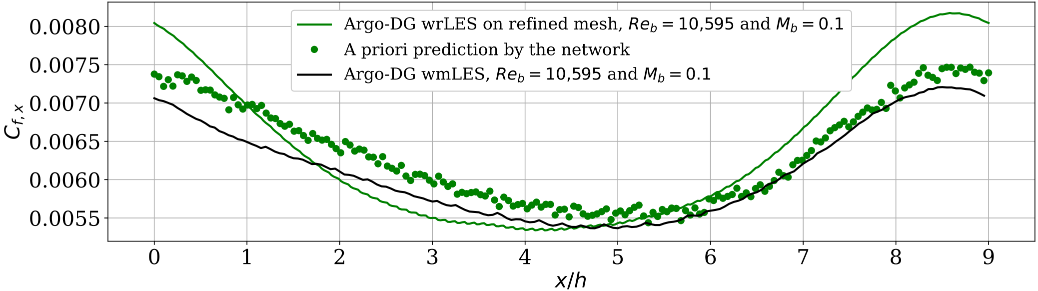


The recirculation bubble is increased but the wmLES still forces an earlier reattachment. Note that the Reichardt LOTW is even worse.

A posteriori validation on the lower wall of the periodic hill $Re_b = 10{,}595$



A posteriori validation on the upper wall of the periodic hill $Re_b = 10{,}595$



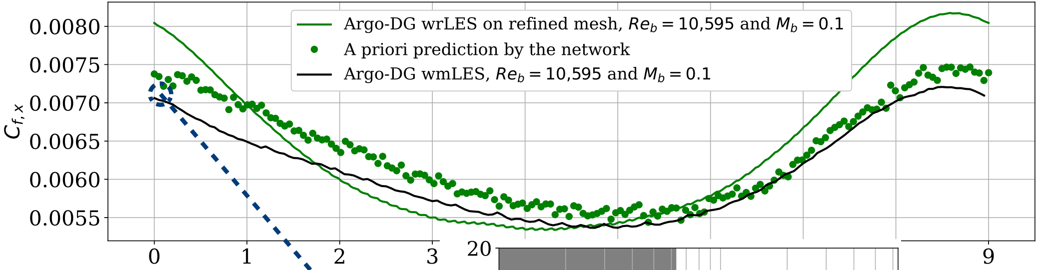
Mean non-dimensional velocity profiles at two x/h locations on the **upper wall** compared to the wrLES on a refined mesh.

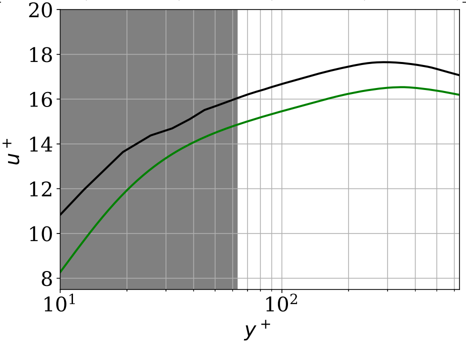
Remarks: The wmLES friction coefficient is similar to the one *a priori* predicted by the neural network **but** the a priori one is far from the true value.

PROD-F-015-02

Cenaero

A posteriori validation on the upper wall of the periodic hill $Re_b = 10{,}595$



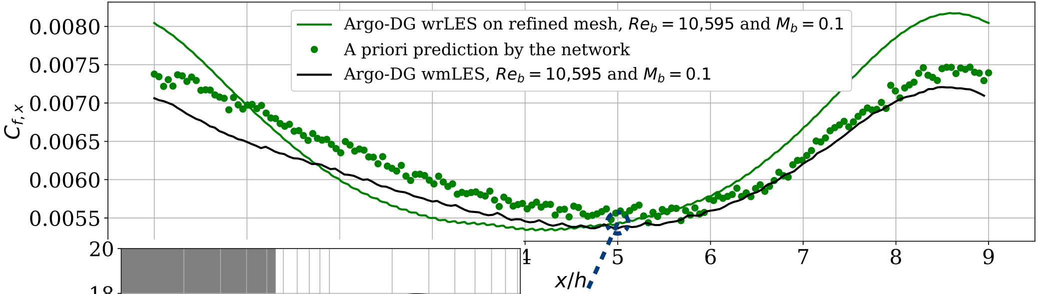


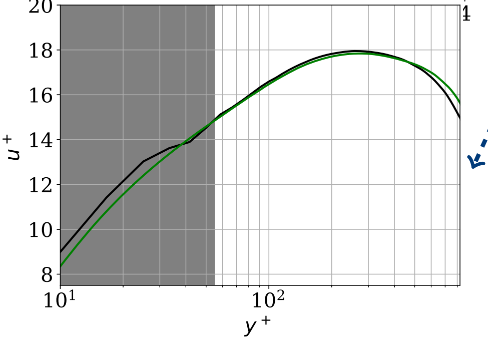
The relative error at $x/h = 0$ is 11% and hence we observe a discrepancy w.r.t. the wrLES non-dimensional profiles.

PROD-F-015-02

Cenaero

A posteriori validation on the upper wall of the periodic hill $Re_b = 10{,}595$





The two curves show a better agreement in the center of the channel.

Cenaero

- The *a posteriori* validation on the periodic hill, initially, gives:
  - a **wrong** recirculation bubble size;
  - a too **early** reacceleration of the flow;

  which impacts the whole physics.
- Using the **directional correction** proposed earlier, the recirculation bubble size increases, but it is still smaller than the wrLES one.

Conclusion

- The *a posteriori* validation on the periodic hill, initially, gives:
  - a **wrong** recirculation bubble size;
  - a too **early** reacceleration of the flow;

  which impacts the whole physics.
- Using the **directional correction** proposed earlier, the recirculation bubble size increases, but it is still smaller than the wrLES one.
- The non-dimensional velocity profiles on the upper wall **need to be corrected** (i.e., the relative error of 11% approximately) with a better neural network.
  - hard to predict with the neural network;
  - need to have more information in the input stencil (e.g., velocity gradient);
  - need to change the normalization (i.e., use the friction velocity).

PROD-F-015-02

The present research benefited from computational resources made available on the **Tier-1 supercomputer** of the Fédération Wallonie-Bruxelles, infrastructure funded by the Walloon Region under the grant agreement n°1117545. We would also like to gratefully acknowledge **SafranTech**'s funding of Mrs. Boxho's thesis.

PROD-F-015-02