Field Inversion
and
Machine Learning

Presentation by Alix Bernard

Based on the paper:

Field-Inversion and Machine Learning with Embedded Neural Networks

https://www.researchgate.net/publication/333808531_Field_Inversion_and_Machine_Learning_With_Embedded_Neural_Networks_Physics-Consistent_Neural_Network_Training

Authors:

J. Holland J. Baeder K. Duraisamy

Introduction

Computational Fluid Dynamics for turbulent flows is widely used in engineering but accurate predictions can be really expensive computationally.

It is especially the case when dealing with high Reynolds numbers, therefore the need for faster accurate predictions is present.

Introduction

Three common types of CFD simulations are:

- DNS: highly accurate at the cost of being computationally expensive
- LES: middle ground solution computing only the main eddies of the flow and modeling the rest
- RANS simulation: inexpensive computationally with low accuracy due to modelisation of the entire flow

Introduction

The idea is to improve the low accuracy RANS simulation using ML. Different methods exist, however the paper considered focus on the use of Field-Inversion Machine Learning (FIML) to achieve improvement.

Field-Inversion

The FI method assumes that a correction parameter β should modify the production term of the model equation to provide accurate predictions.

Spalart-Allmaras model equation:

$$\frac{\partial \hat{\nu}}{\partial t} + u_j \frac{\partial \hat{\nu}}{\partial x_j} = \gamma c_{b1}(1 - f_{t2})\hat{S}\hat{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\hat{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \hat{\nu})\frac{\partial \hat{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}\right]$$

The parameter β is integrated into the production term, for each node k of the field, as follows:

$$P_k = \beta_k \gamma c_{b1}(1 - f_{t2})\hat{S}\hat{\nu}$$

Field-Inversion

The parameter β being a multiplicative coefficient to the production term, to inhibit its presence we can simply set it as 1.

The goal from here is to find a better value for β that improve the accuracy knowing that we possess the truth-data. To do so, a cost function J(β) accounting for the discrepancy between the truth-data and modeled-data is introduced.

Now two different approaches are distinguished, the classic and direct approaches.

FIML-classic

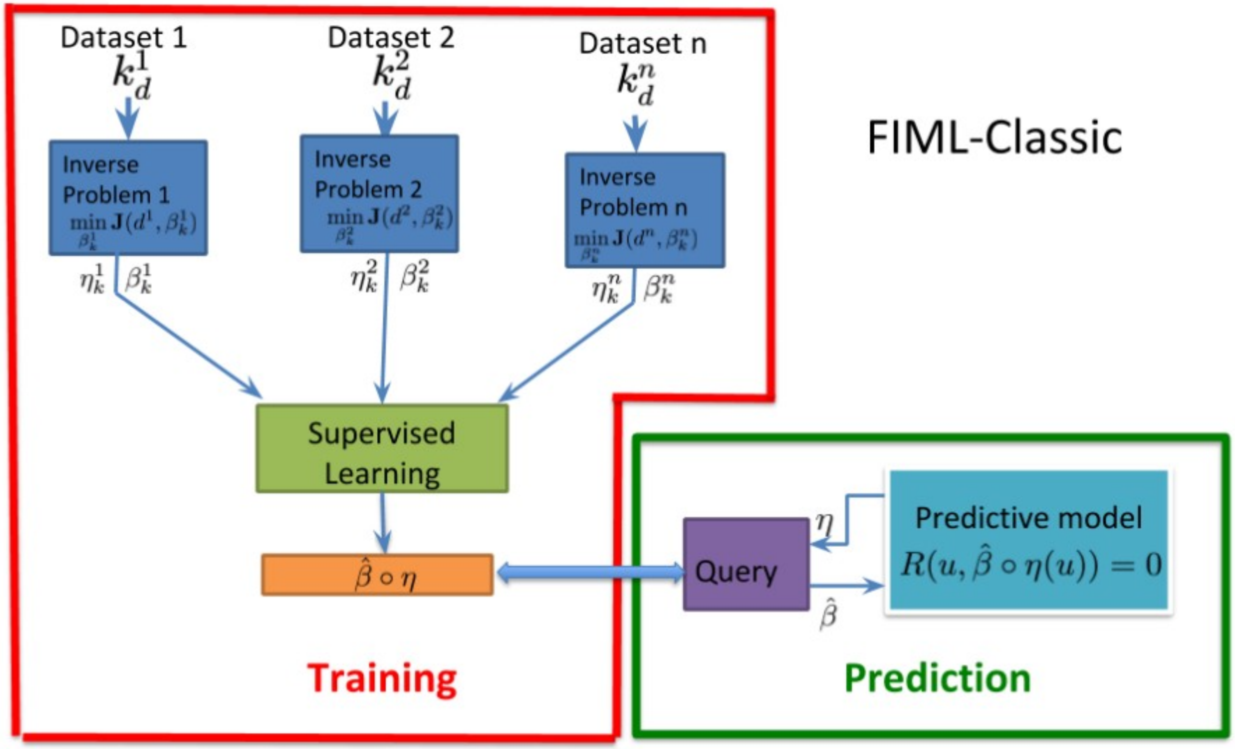
We define a cost function $J_c(\beta)$:

$$J_c(\beta) = ||k_d - k_m(\beta)||_2^2 + \lambda||\beta - 1.0||_2^2$$

- k_d: truth data
- k_m: modeled data
- λ: confidence value into the modeled data
  (if high confidence then λ close to 0)

FIML-classic

For each dataset we minimize $J_c(\beta)$ to obtain what is considered the true β. Then all the features are assembled and fed to a NN that is trained to predict β from the features η.



FIML-classic

Such approach will contain residual training errors as there is no guarantee that there exists an algorithm that can produce $\beta(\eta)$.

Therefore a second approach is introduced to minimize such training errors, the direct approach.

FIML-direct

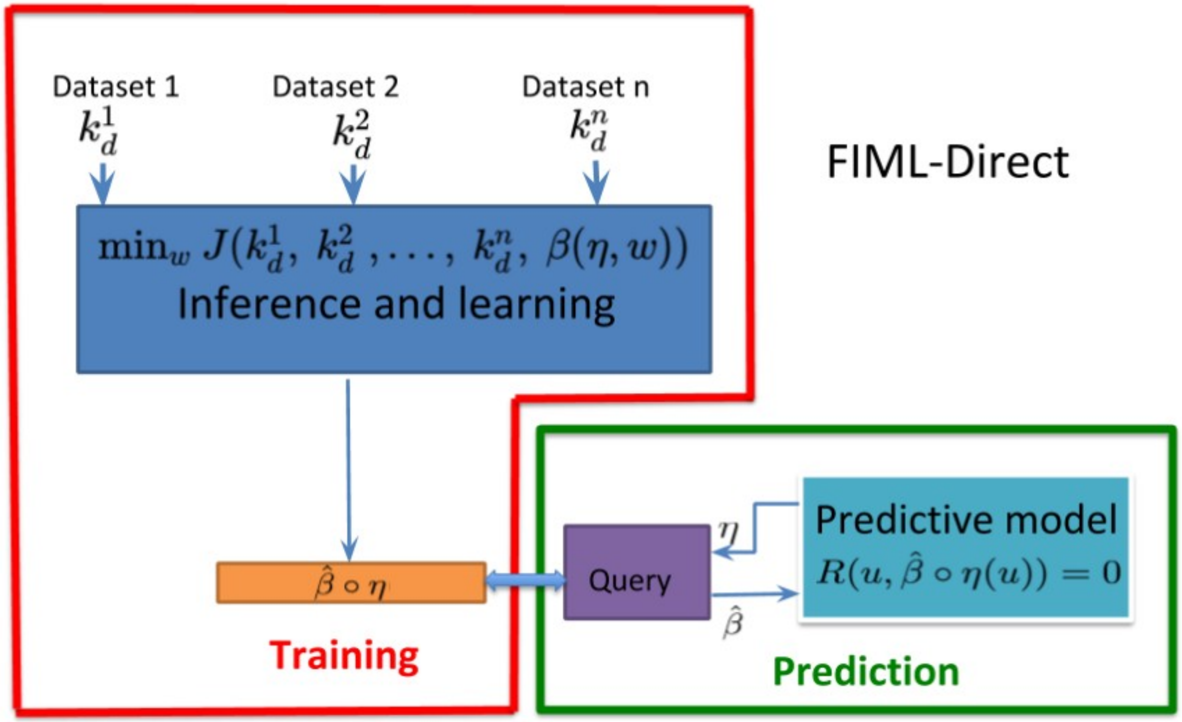
In this approach the inversion process to obtain β is integrated in the NN by defining β as a function of the features η and the weights w of the NN.

The modified cost function becomes:

$$J_d(w) = ||k_d - k_m(w)||_2^2 + \lambda||\beta(w) - 1.0||_2^2,$$

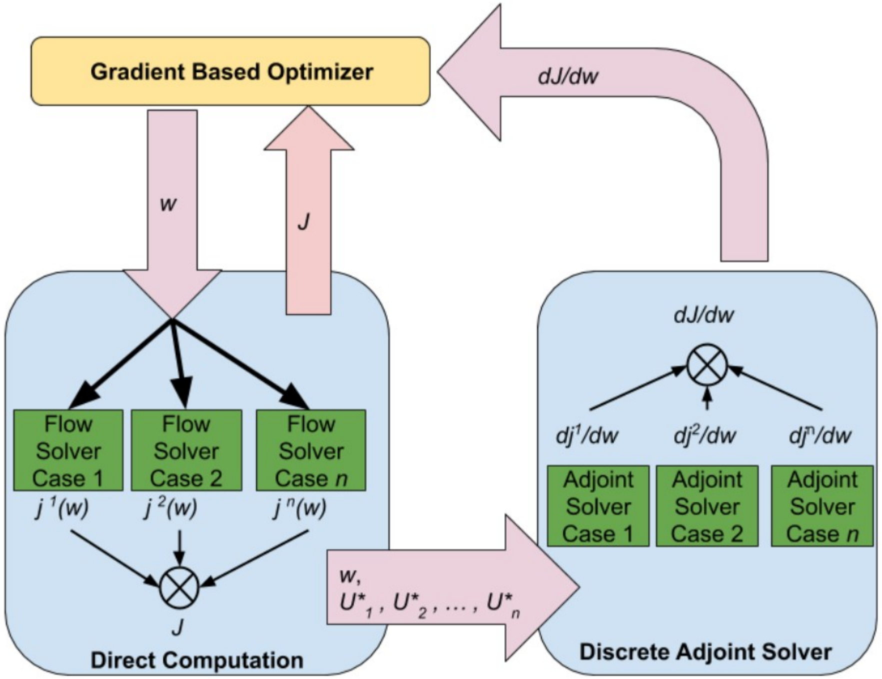
FIML-direct

All the features are assembled and the inversion is done by minimizing the cost function $J_d(\beta(w))$ by directly modifying the weights of the NN since they infer β.



FIML-direct

In this direct approach, the prediction of the model with correction is computed so we have $J_d(\beta)$, from there the discrete adjoint solver can calculate the gradient of J_d according the the weights since β is a function of the weights. Finally a better β can be chosen and the process repeated until β is suitable enough.

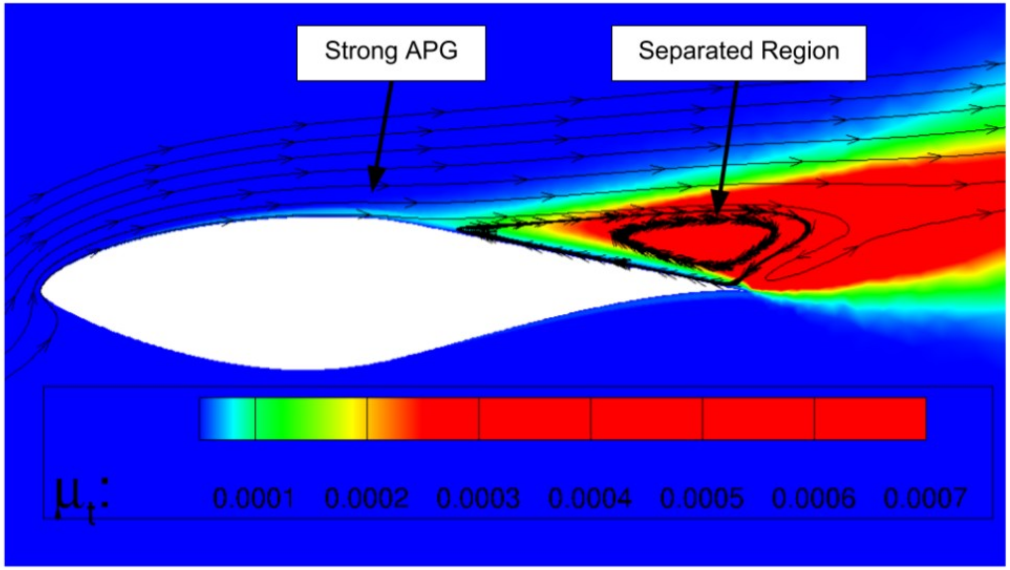


FIML-direct

By setting the weights of the NN during the inversion process we get rid of the uncertainty of the NN (considering on the chosen NN structure) to predict an optimal β which would lead to a lower accuracy as seen for the FIML-classic approach.

FIML-classic over Airfoil

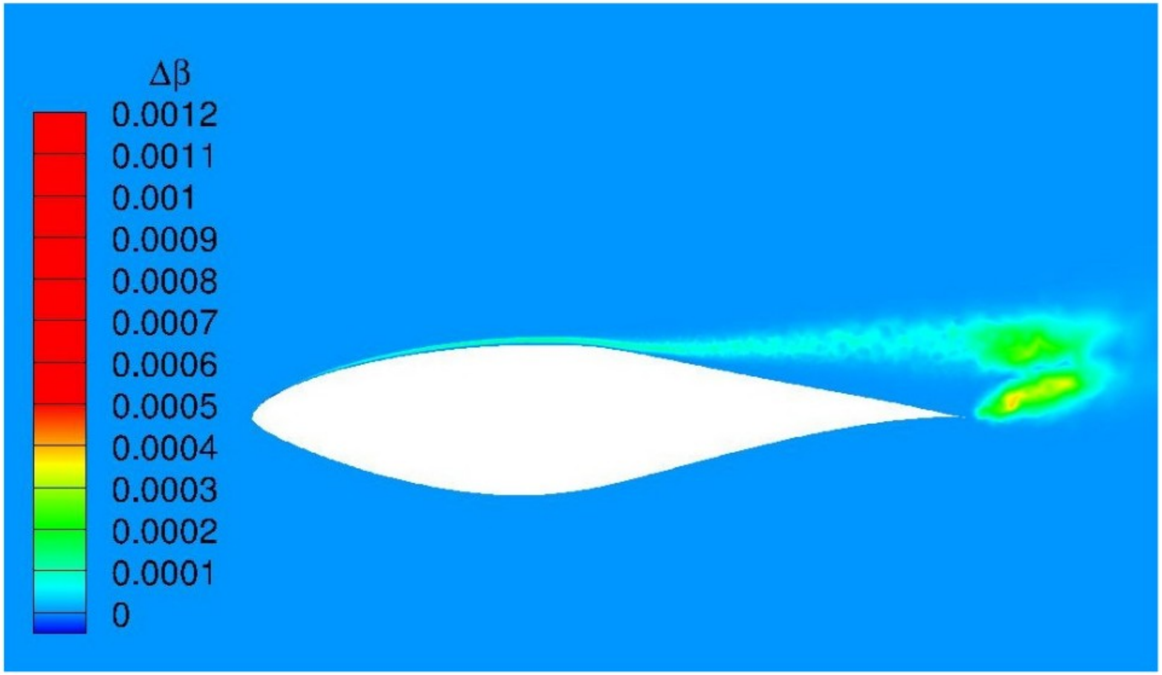
Here is the prediction of eddy viscosity via a baseline RANS model for a flow over an airfoil with angle of attack 1.02 degrees:



RANS models are known to over-predict the eddy viscosity in the adverse pressure gradient (APG) area and inaccurate in the recirculating region.

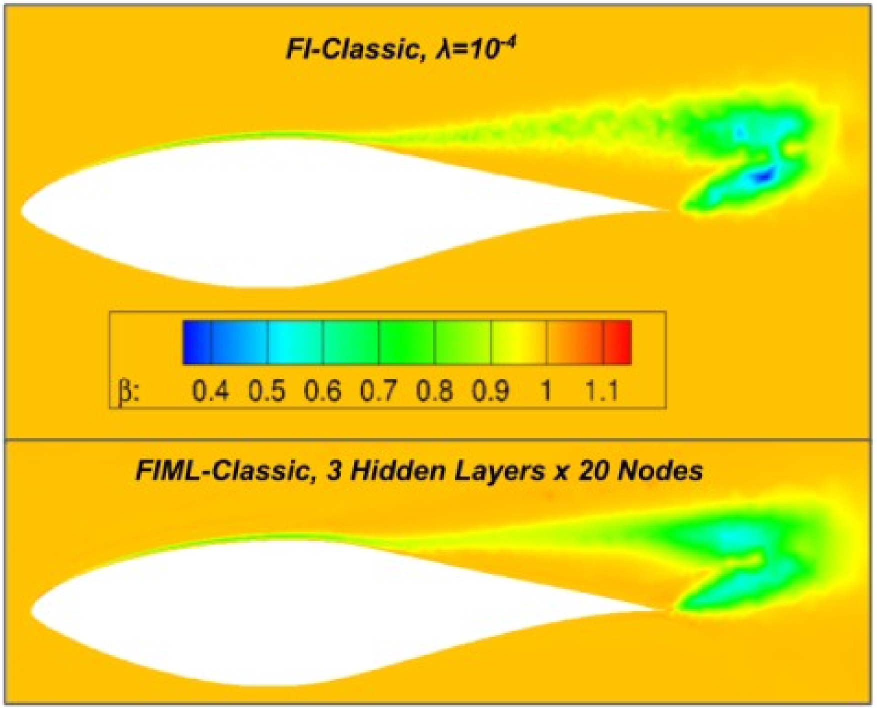
FIML-classic over Airfoil

If we look at the FI-classic gradient for β we can notice that indeed those two areas are the one that have the higher gradient for β as it is as expected poorly modeled with RANS model.



FIML-classic over Airfoil

Below is the comparison between FI-classic and FIML-classic.



	C_L	Error
Target	1.0546	-
Baseline	1.15826	0.10366
Inverse	1.06726	0.01266
Augmentation	1.06914	0.01454

We notice that the FI-classic provide a great improvement to the model and the FIML-classic as well even if slightly less accurate than the FI-classic.

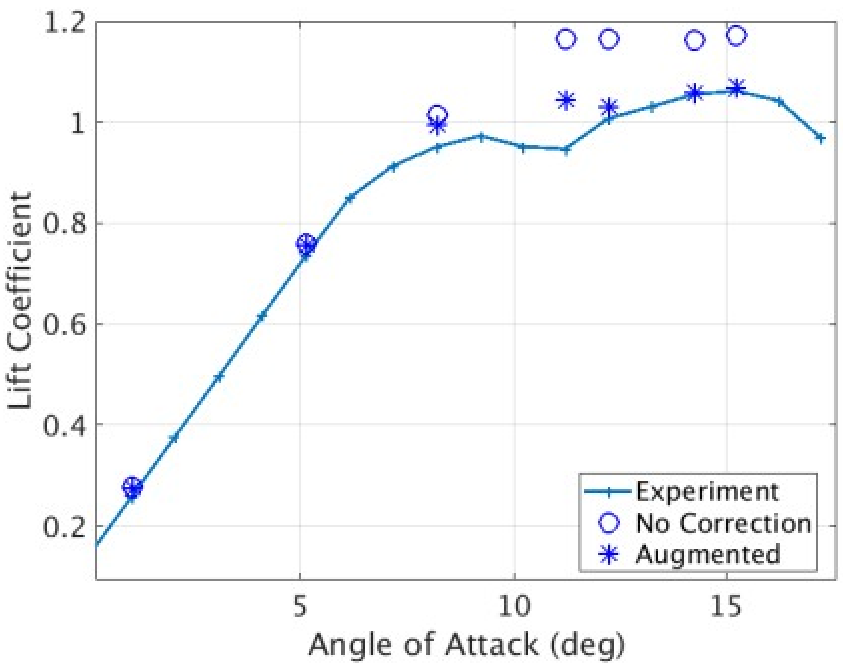
FIML-direct over Airfoils

Here is considered the direct approach for field inversion considering multiple dataset simultaneously.

Three angles (1.02, 8.2, 14.24 degrees) of attack are considered to train the NN however for the testing four different angles (5.13, 11.21, 12.22, 15.24 degrees) are also used.

FIML-direct over Airfoils

Looking at the prediction of lift coefficient via FIML-direct on the seven angles of attack we can see a good improvement especially for high angles of attack.



Conclusion

We have seen that the FIML process allows to improve the prediction of RANS models for flows around airfoils, especially for the strong APG area and in the recirculating region.

Two different FIML methods have been presented, the classic and direct one.

The direct method have the advantage of providing an optimal (considering the limitation of the chosen NN structure) inference of the corrective parameter by the NN since it is embedded in the inverse problem.