Learning Aerodynamics Through Data to Improve Optimization Algorithms

PIR defense by

Mario Leupolt*

Tutor: Prof. Joseph Morlier°

24/05/2021

*mario.leupolt@student.isae-supaero.fr
°joseph.morlier@isae-supaero.fr

Content

- Introduction
- Background
- Methodology
- Results
- Conclusion

Introduction

- airfoil geometry crucial part of design process
- obtaining of aerodynamic coefficients:
  - windtunnel
  - simulations

→ expensive and time consuming

- neural networks
- existing database:
  - two surrogate models already built that use this database

GOAL

Develop models to predict aerodynamic coefficients

↓

WHY?

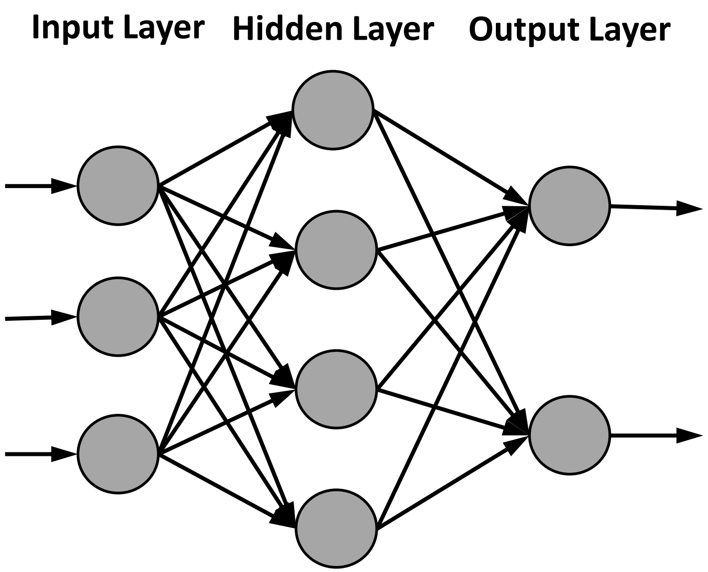
improve optimization algorithms
Scalar and graph predictions

[1] A. I. J. Forrester, A. Sobester, and A. J. Keane. „Engineering design via surrogate modelling“. Wiley, 2008.
[2] D. P. Raymer. “Aircraft design: A conceptual approach”. AIAA, 1992
[3] Bouhlel, M. A., He, S., and Martins, J. R. R. A., “mSANN Model Benchmarks,” Mendeley Data, 2019. https://doi.org/10. 17632/ngpd634smf.1.

Background – Neural Networks

- computational model:
  - trying to imitate working process in human brain
  - training, validation, testing phase
- layers with several neurons
- neurons process inputs:
  - activation function, weight, bias
- every training step tries to improve

  weights and biases to get better predictions:
  - loss function, optimizer ⟶ backpropagation



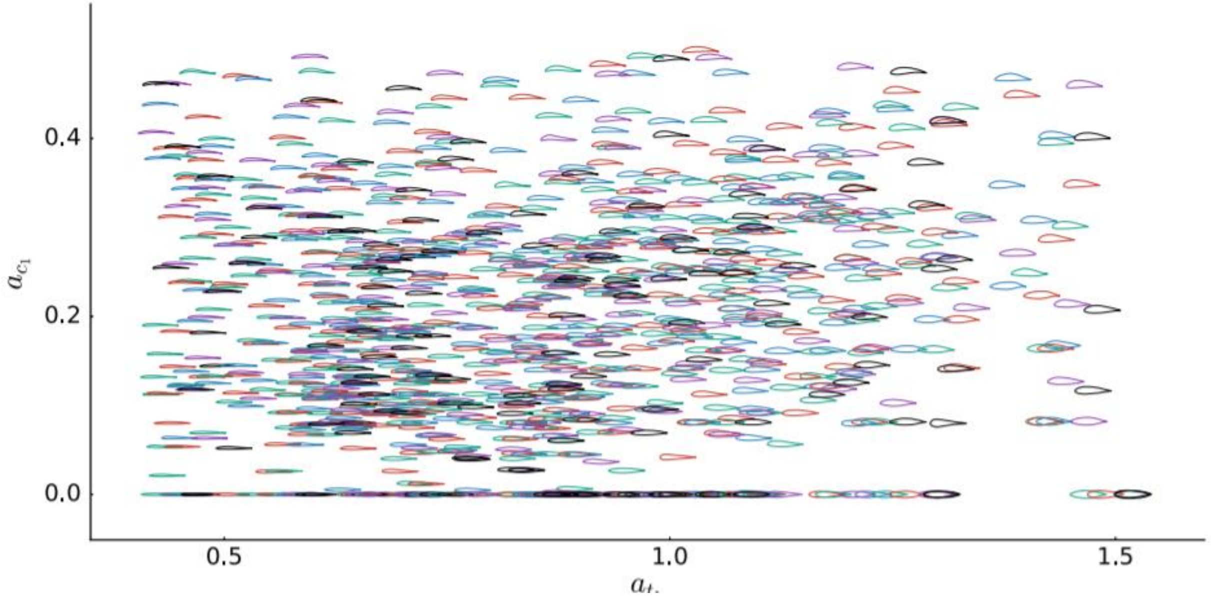
$$y_j = \varphi\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right)$$

[1] Michelucci, U., “Applied Deep Learning,” Springer Science, 2018.

Background - Parameterisation

- input in neural networks: camber and thickness mode shapes
- from thickness and camber lines of 1172 airfoils from UIUC database:
- Singular Value Decomposition (SVD)

$$A_t = U_t \Sigma_t V_t^T$$



[distribution of UIUC airfoils on the base of first camber andthickness mode shape] [1]

[1] J. Li, M. Amine Bouhlel, and J. R. R. A. Martins. Data-based approach for fast airfoil analysis and optimization. AIAA Journal, February 2019

Background - Parameterisation

- How to obtain mode shapes out of a random airfoil geometry?

Interpolation with cubic B-Spline

Uniform distribution scheme of coordinate points

$$x_i = \frac{1}{2}\left(\cos\left(\frac{2\pi(i-1)}{250}\right)+1\right) \qquad i = 1 \ldots 251$$

Calculation of camber and thickness line

matrix multiplication from camber and thickness modes with camber and thickness lines

$$\boldsymbol{a}_c^T = \boldsymbol{y}_c^T \boldsymbol{\phi}_c \qquad \boldsymbol{a}_t^T = 2\boldsymbol{y}_t^T \boldsymbol{\phi}_t$$

- How to get back from the mode shapes to the airfoil geometry?

matrix multiplication from camber and thickness mode shapes with mode matrix

$$\boldsymbol{y} = \begin{pmatrix} \boldsymbol{\phi}_c & \boldsymbol{\phi}_t \\ \boldsymbol{\phi}_c & -\boldsymbol{\phi}_t \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_c \\ \frac{1}{2}\boldsymbol{a}_t \end{pmatrix}$$

Background - Parameterisation

- How to o

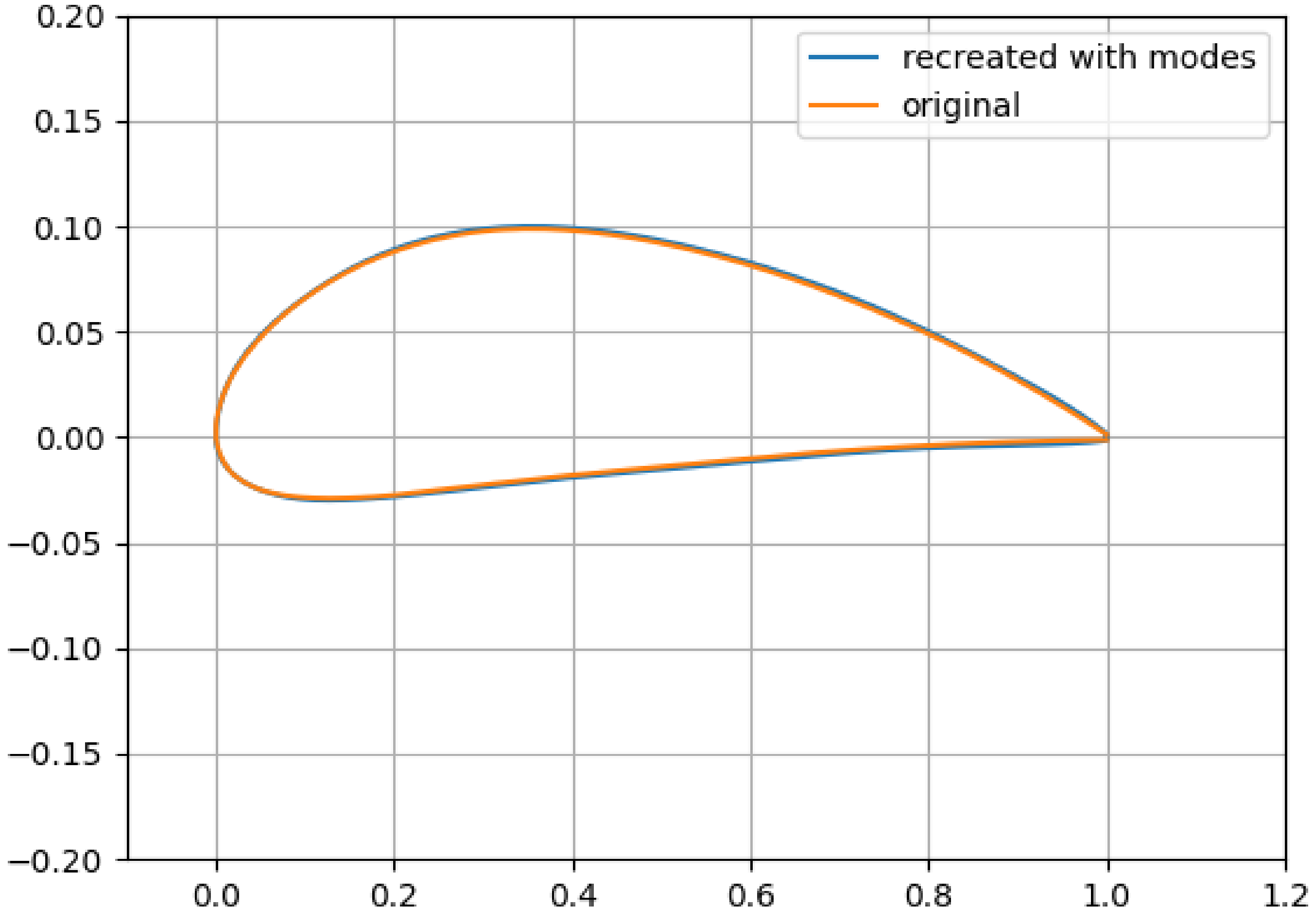
Uniform

Calc

matrix n
mo

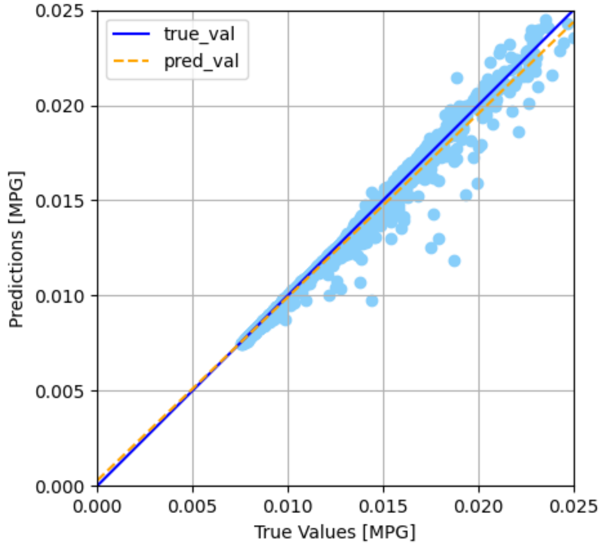
- How to g

matrix n

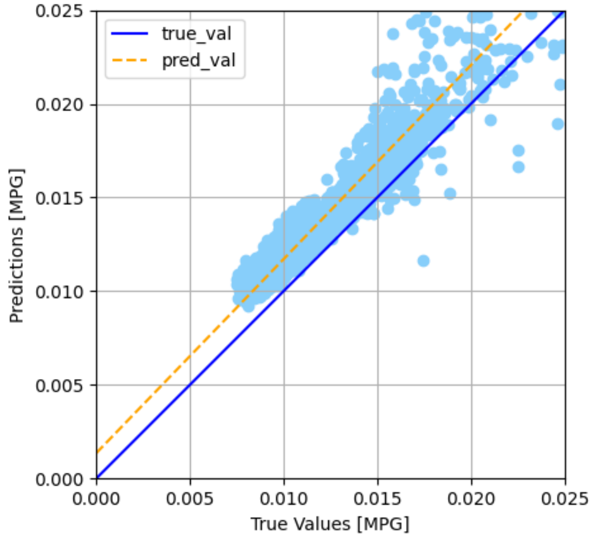


Methodology - Keras [2]

- deep learning API that uses the platform tensorflow
- Determination if separate models or one model for all coefficients:
    - Networks with simple architecture



[Cd single]



[Cd all]

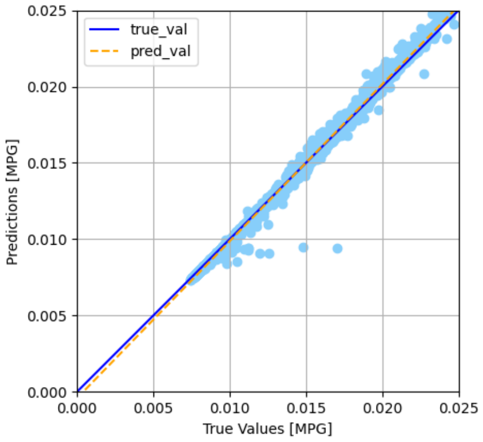
CONCLUSION:

- **separate models**
- **improvement of architecture**

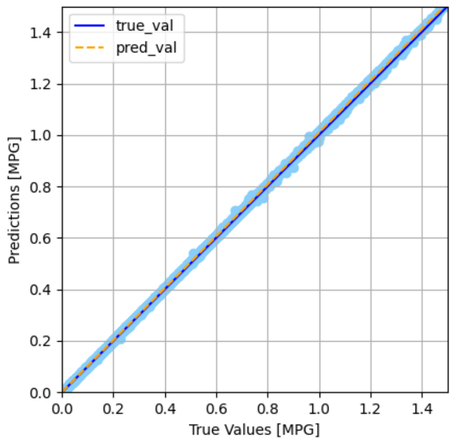
[1] Keras documentary. Online accessed on 05/05/2021. https://keras.io/about/.
[2] Keras logo. Online accessed on 07/05/202. https://keras.io/

Methodology - Keras

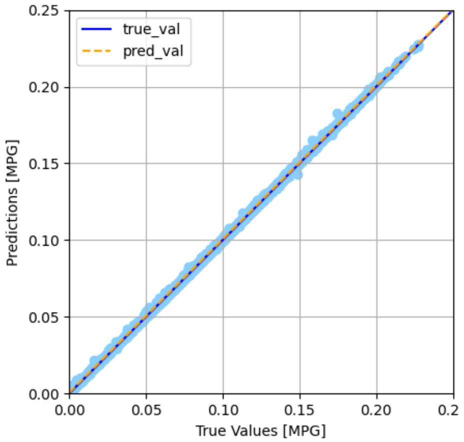
- two approaches with k-cross validation:
  - architecture from paper as base for hyperparameter study
  - LSTM layers
- hyperparameter study brought best results → base for final architecture



[Cd]



[Cl]

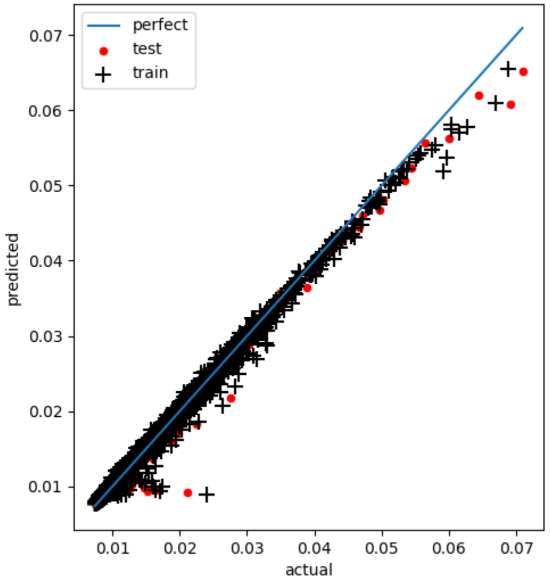


[Cm]

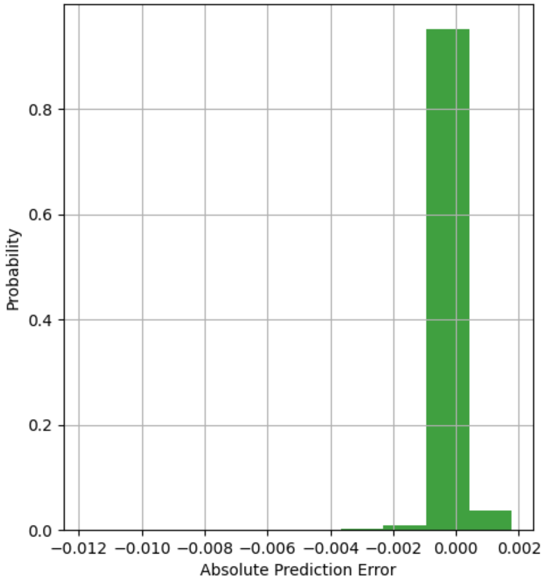
[1] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," Neural Computation, 1997.
[2] Refaeilzadeh, P., Tang, L., and Liu, H., "Cross-Validation," Springer US, 2009.

Methodology - SMT

- SMT – collection of surrogate modelling, sampling and benchmark functions
- emphasises the use of gradient information
- Construction of separate GENN:
  - multilayer perceptron
  - incorporate gradient information during training phase



[prediction quality]

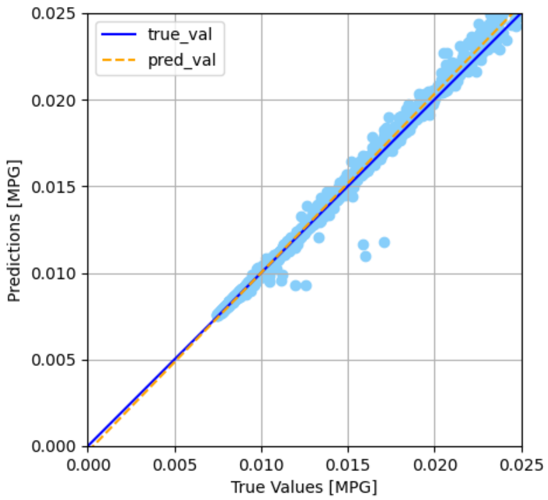


[distribution of absolute error]

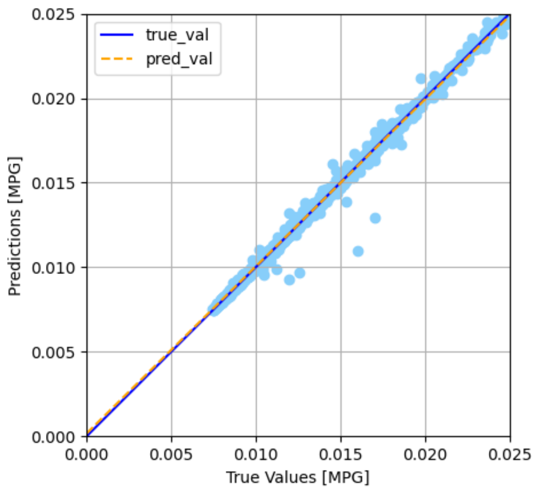
[1] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins. A python surrogate modeling framework with derivatives. page 102662, 2019

Methodology – MONOLITH [2]

- online platform: predictions and solving optimization problems
- network for all coefficients and separate networks for each coefficient
- first optimization of an airfoil using the mode shapes



[Cd all]



[Cd single]

ASSUMPTION:

- **separate models will deliver better predictions**

[1] Monolith. Online accessed on 05/05/2021. https://www.monolithai.com/industry/reduce-testing.
[2] Monolith logo. online accessed on 07/05/2021. https://www.monolithai.com/

Results – Optimization Monolith AI

- optimization with and without mode shape bounds:
  - target: best lift to drag ratio at a certain Mach number and angle of attack

CONCLUSION:

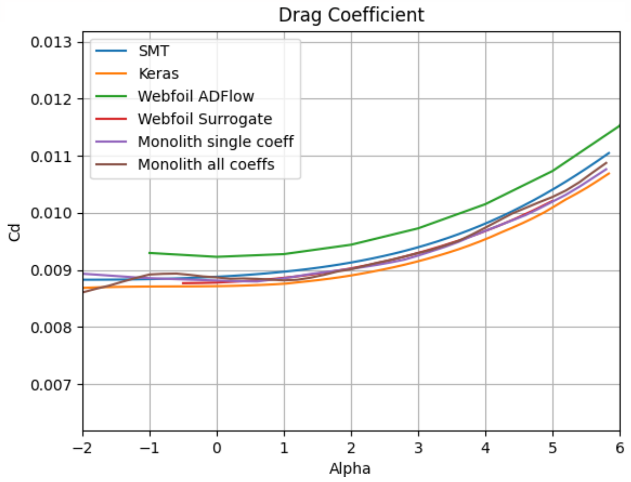
no bounds: unrealistic airfoil shape
bounds: more realistic; still sharp trailing edge

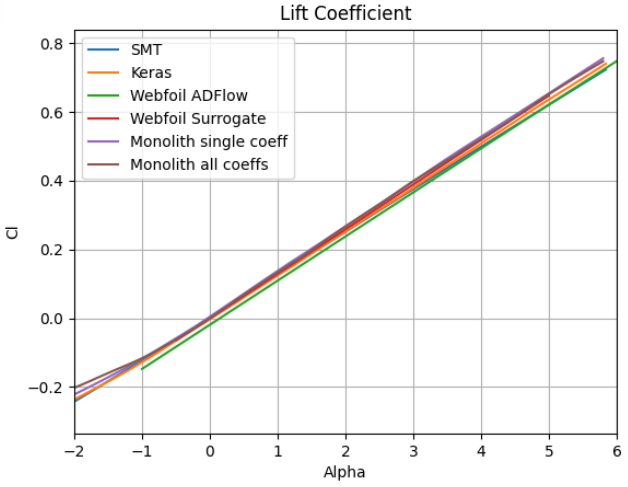
FUTURE WORK:

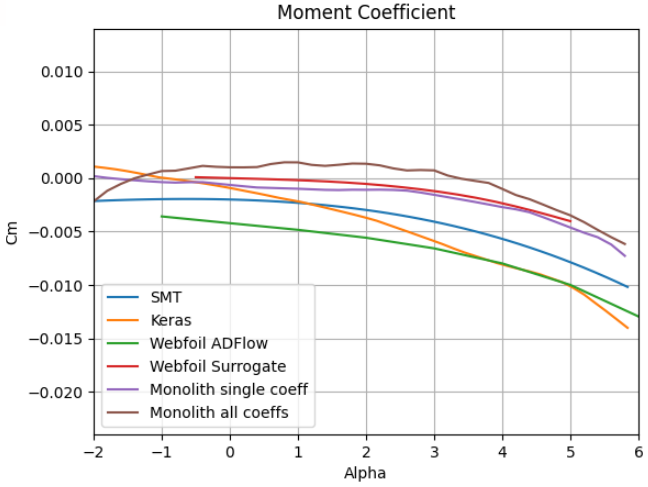
Define optimal mode shapes for optimization

Results - Models

- prediction of the aerodynamic coefficients at Ma = 0.5 over alpha
- comparison to the results from Webfoils surrogate model and ADFlow (calculated coefficients for database)

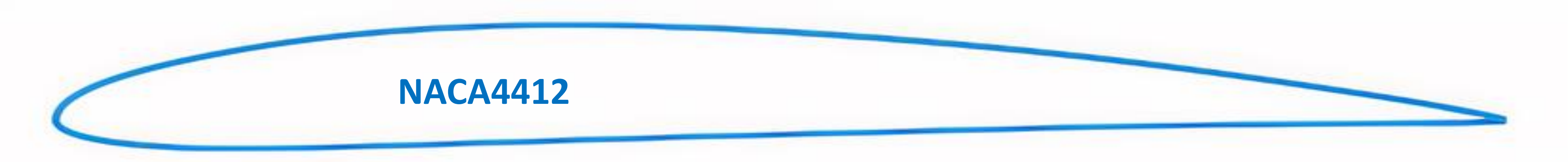


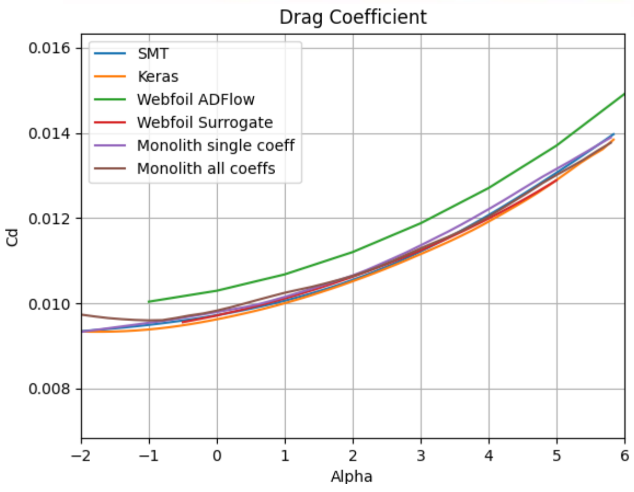


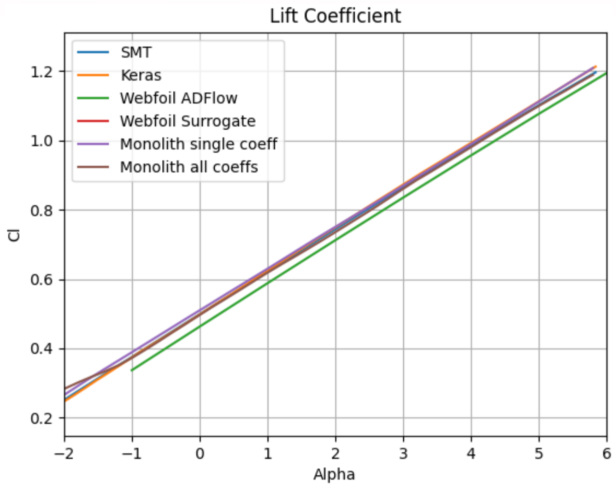


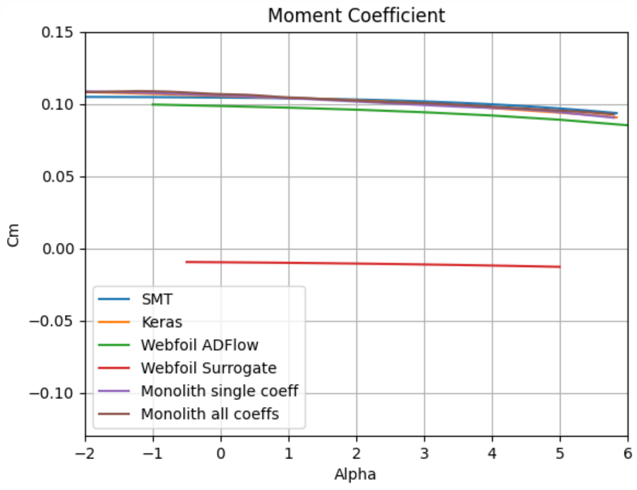
Results - Models

- prediction of the aerodynamic coefficients at Ma = 0.5 over alpha
- comparison to the results from Webfoils surrogate model and ADFlow (calculated coefficients for database)









Conclusion

- mode shape bounds must be applied
- SMT delivers best predictions → incorporate in optimization algorithms

FUTURE WORK:

- **define optimal mode shape bounds for optimization**
- **improve the SMT models**
- **incorporate the predictions from the SMT models in optimization algorithms**

Appendix

- final architecture for the models in Keras

Model	Drag Coefficient		Lift Coefficient		Moment Coefficient	
Learning Rate	0.001		0.0005		0.0005	
R2-Value	0.9998		0.9995		0.9995	
RMSE-Value	0.0007		0.0074		0.0012	
Layers	tanh	180	tanh	100	tanh	180
	sigmoid	160	sigmoid	120	sigmoid	160
	selu	140	selu	140	selu	140
	selu	140	selu	120	selu	120
	selu	140	Leaky ReLU	180	Leaky ReLU	80
	Leaky ReLU	120	Leaky ReLU	180	Leaky ReLU	40
	Leaky ReLU	120				
	Leaky ReLU	120				