Available online at www.eccomasproceedia.org
Eccomas Proceedia EUROGEN (2023) 22-39

ECCOMAS
Proceedia

EUROGEN 2023
15th ECCOMAS Thematic Conference on
Evolutionary and Deterministic Methods for Design, Optimization and Control
N. Gauger, K. Giannakoglou, M. Papadrakakis, J. Periaux (eds.)
Chania, Crete Greece, 1–3 June 2023

PHYSICS-BASED REGULARIZATION OF NEURAL NETWORKS FOR AERODYNAMIC FLOW PREDICTIONS

Derrick Hines Chaves*, Mateus Dias Ribeiro, and Philipp Bekemeyer

German Aerospace Center (DLR)
Institute of Aerodynamics and Flow Technology, C²A²S²E
Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: {Derrick.HinesChaves, Mateus.DiasRibeiro, Philipp.Bekemeyer}@dlr.de

Abstract. *Aerodynamic data plays a central role in the process of aircraft design, optimization and certification. For these processes a vast amount of data is required for various flight conditions throughout the flight envelope. Currently this data is commonly produced using Computational Fluid Dynamics (CFD). However, such simulations based on the Reynolds-averaged Navier-Stokes equations are computationally expensive and become prohibitive for tasks such as load analysis and shape optimization. During the last decades, this has motivated research focusing on the use of data-driven models with lower evaluation times than the full-order model to replace high-fidelity CFD simulations. More recently, deep learning approaches have gathered significant interest in the aerodynamic community. For the task of predicting surface pressure coefficient distributions, one of the proposed models consists of a multilayer perceptron that for each node in the mesh outputs a prediction of the local coefficient based on the node coordinates and the global operational conditions. If required, known integration formulas are used to compute integral quantities, such as the lift and pitching moment coefficients, based on the previously obtained distribution. In this paper we train a multilayer perceptron that predicts pressure coefficient distributions and uses known integration formulas to compute predictions of global coefficients, both during training and inference. We examine the effect that is achieved in the prediction of global coefficients with the use of a physics-based regularization term that during training penalizes the multilayer perceptron if the predicted global coefficients deviate from the reference values. The method is tested for the NASA Common Research Model transport aircraft with an underlying mesh consisting of around* $500,000$ *surface points. Results show that, when using the mentioned approach for the fine-tuning of a trained multilayer perceptron, physical knowledge can be explicitly revealed to the deep learning model but only limited improvements are achieved in the predictions of the lift and pitching moment coefficients.*

Keywords: Deep Learning, Neural Network, Regularization, Aerodynamics

Peer-review under responsibility of the organizing committee of EUROGEN 2023.
doi: 10.7712/140123.10188.18859

1 INTRODUCTION

During aircraft design and optimization accurate flow field predictions are required for various operational conditions over the flight envelope in order to analyze aerodynamic performance, structural loads and handling qualities. Computational fluid dynamics (CFD) simulations are typically used to produce high-fidelity data. However, each simulation has a very high computational cost making it infeasible to use high-fidelity Reynolds-averaged Navier-Stokes (RANS) solvers to produce all the required data during industrial development cycles. For this reason, fast surrogate models for the accurate prediction of flow fields and integral quantities are of great interest. Data-driven models have gained an increasing attention in recent years. Within this class, proper orthogonal decomposition (POD) [1] as a dimensionality reduction technique combined with an interpolation method such as radial basis functions or Gaussian Processes is arguably the most common method. Application examples for aerodynamics are widespread and can be found in [2, 3, 4, 5, 6]. These models are reported to be easy to construct and highly accurate as long as only linear behaviour is present. However, in transonic flows, in the presence of shocks, the accuracy of such models reduces significantly. Therefore, alternative methods are sought after.

Deep learning (DL) models have attracted attention in recent research due to their success at the extraction and representation of hierarchical data features [7]. Publications are available for the prediction of aerodynamic coefficients for airfoils, including integral quantities (lift and drag) [8], fields (surface pressure distribution) [9] as well as unsteady forces [10, 11]. In [12] convolutional neural networks are introduced to predict the velocity field in non-uniform steady laminar flows, while several extensions are available in [13, 14, 15, 16]. An extension towards industrial relevant 3D cases featuring transonic flows including shocks and boundary layer separation relying on a multilayer perceptron used for the pointwise prediction of pressure coefficients has been studied in [17, 18]. In recent work the model was extended to include the surface normals as additional input achieving further improved results [19]. For the prediction of integral coefficients, such as the lift and pitching moment coefficients, well-established physical equations were used to compute them based on the obtained surface pressure coefficient distribution. However, in this approach prior knowledge about the physical system, such as the relationship between the pressure distribution and the lift coefficient, was available but unused during the training of the network. An often pursued alternative is to construct separate models for distributed and global quantities of interest which can easily lead to differing trends when using both models together.

In this paper we train a multilayer perceptron that predicts pressure coefficient distributions and uses known integration formulas to compute predictions of global coefficients, both during training and inference. We examine the effect that is achieved in the prediction of global coefficients with the use of a physics-based regularization term [20] that during training penalizes the multilayer perceptron if the predicted global coefficients deviate from the ground truth. The studied methodology is applied for an industrial relevant 3D case known as the NASA Common Research Model (NASA CRM) transport aircraft. It is compared to the same model without the physics-regularization term and to the aforementioned proper orthogonal decomposition coupled with interpolation. We refer to these three methods as indirect methods because they predict the surface pressure coefficient distribution and then as a second step calculate the desired global coefficient. As mentioned before, a direct prediction of the global coefficients is also possible and several methods have been proposed [21, 22]. Two of these approaches are radial basis function interpolation (RBF) and a multilayer perceptron for global predictions. To

provide a more comprehensive analysis, the performance of the indirect methods is compared to the one of these direct methods. We highlight that the indirect methods have the added benefit that they yield a consistent model in which the predicted integral quantities agree with the distributed quantities. Such models can be used well beyond the prediction of a global quantity during the multidisciplinary process of aircraft design.

This paper is organized as follows. Section 2 presents the methodology used to obtain the ground global truth global coefficients and then describes the surrogates models. Next, Section 3 outlines further details of the test case and subsequently Section 4 discusses the results. The conclusion is found in Section 5 and additional information are given in the Appendix 7.

2 METHODOLOGY

The three-dimensional Navier-Stokes equations in conservative form are given on a control volume Ω and its closed surface $\partial\Omega$ by

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{W}d\Omega + \oint_{\partial\Omega}(\vec{F}_{\mathrm{c}} - \vec{F}_{\mathrm{v}})\,dS = 0 \tag{1}$$

The conservative variables are expressed by $\vec{W} = (\rho, \rho u, \rho v, \rho w, \rho E)^{\top}$, where ρ is density, u, v, w the velocity components in the spatial directions and E the total energy. $\vec{F}_c$ and $\vec{F}_v$ are the vectors of convective and viscous fluxes, respectively. The steady state is achieved if for all control volumes the time derivative vanishes, or equivalently if the fluxes become zero. Employing a finite volume scheme a steady-state solution is found, yielding a pressure distribution at the surface of the aircraft Γ formed by n faces Γ_i. The pressure p_i is hence obtained at each face Γ_i. The pressure coefficient $c_{\mathrm{p},i}$ is a non-dimensional quantity computed as

$$c_{\mathrm{p},i} = \frac{p_i - p_\infty}{q_\infty} \tag{2}$$

where p_∞ and q_∞ are the static pressure and dynamic pressure of the incoming flow, respectively. The obtained $\vec{c}_{\mathrm{p}}$ distribution at the surface of the aircraft is then used to obtain global aerodynamic force and moment coefficients.

In this work we investigate the prediction of lift coefficient C_{L} and pitching moment coefficient C_{my} and omit the viscous force components during their calculation since the considered models do not predict the skin friction coefficients. We consider the aircraft flying in a symmetric flight at a given Mach number under an angle of attack α and use a body-fixed coordinate system, shown in Figure 1a, with axial, transversal and normal directions denoted by the unit vectors $\hat{x}, \hat{y}, \hat{z}$ respectively. The pressure force $\vec{F}_{\mathrm{p},i}$ at the face Γ_i is obtained as

$$\vec{F}_{\mathrm{p},i} = -c_{\mathrm{p},i}\,\vec{n}_i \tag{3}$$

where $\vec{n}_i = (n_{i,x}, n_{i,y}, n_{i,z})^{\top} = \Delta s_i \hat{n}_i$ is the face normal vector, which is the outward unit normal vector multiplied by the area of the face. The pressure forces are summed up to get a total pressure force $\vec{F}_{\mathrm{p}}$ over the aircraft surface as

$$\vec{F}_{\mathrm{p}} = \sum_{i=1}^{n}\vec{F}_{\mathrm{p},i} \tag{4}$$

Considering a reference area S_Γ, the coefficients C_A and C_N are calculated as

$$C_\mathrm{A} = \frac{\vec{F}_\mathrm{p} \cdot \hat{x}}{S_\Gamma}, \quad C_\mathrm{N} = \frac{\vec{F}_\mathrm{p} \cdot \hat{z}}{S_\Gamma} \tag{5}$$

representing normalized versions of the axial and normal components of the total pressure force. The lift coefficient C_L in the aerodynamic coordinate system, which is a clockwise rotation of α about the transversal axis as shown in Figure 1b, is calculated as

$$C_\mathrm{L} = -C_\mathrm{A} \sin\alpha + C_\mathrm{N} \cos\alpha \tag{6}$$

yielding a normalization of lift. Overall, in this discrete physical model the lift coefficient is obtained as a linear transformation of the pressure coefficient distribution $\vec{c}_\mathrm{p}$, dependent on the angle of attack α and the aircraft surface geometry $\mathcal{G}$.

$$C_\mathrm{L} = T_{C_\mathrm{L}}(\vec{c}_\mathrm{p}; \alpha, \mathcal{G}) = \sum_{i=1}^{n} \left(\frac{n_{i,x} \sin\alpha - n_{i,z} \cos\alpha}{S_\Gamma} \right) c_{\mathrm{p},i} =: \vec{a}_{C_\mathrm{L}} \cdot \vec{c}_\mathrm{p} \tag{7}$$

Regarding the calculation of C_my, the contribution of the face Γ_i to the moment coefficient with respect to a reference point is obtained as

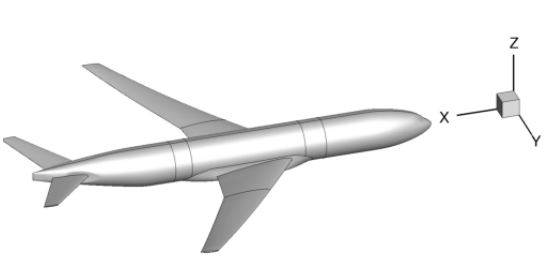
$$\vec{M}_i = \vec{R}_i \times \vec{F}_{\mathrm{p},i} \tag{8}$$

where $\vec{R}_i = (\Delta x_i, \Delta y_i, \Delta z_i)^\top$ is the displacement vector of the integration point in the face Γ_i from the reference point. The pitching moment coefficient C_my is given by the sum of the y-components of $\vec{M}_i$, divided by the product of a reference chord length $\mathcal{L}_\mathrm{chord}$ and a reference area S_Γ:

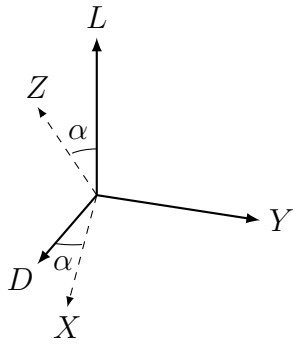
$$C_\mathrm{my} = \frac{\sum_{i=1}^{n} \vec{M}_{i,y}}{\mathcal{L}_\mathrm{chord}\, S_\Gamma} \tag{9}$$

This model then gives the pitching moment coefficient as a linear transformation of the pressure coefficient distribution as

$$C_\mathrm{my} = T_{C_\mathrm{my}}(\vec{c}_\mathrm{p}; \alpha, \mathcal{G}) = \sum_{i=1}^{n} \left(\frac{\Delta x_i\, n_{i,z} - \Delta z_i\, n_{i,x}}{\mathcal{L}_\mathrm{pitch}\, S_\Gamma} \right) c_{\mathrm{p},i} =: \vec{a}_{C_\mathrm{my}} \cdot \vec{c}_\mathrm{p} \tag{10}$$



(a) Body-fixed coordinate system and aircraft surface Γ



(b) Aerodynamic coordinate system

Figure 1: Body-fixed and aerodynamic coordinate systems

The goal of the methods in this paper is the prediction of a global coefficient C, either C_{L} or C_{my} from operational conditions given by the Mach number, angle of attack, inboard aileron deflection angle, outboard aileron deflection angle, elevator deflection angle and horizontal tailplane deflection angle that we denote as $\vec{x} = (M, \alpha, \phi_{\mathrm{inAil}}, \phi_{\mathrm{outAil}}, \phi_{\mathrm{el}}, \phi_{\mathrm{htp}})^{\top}$. The global coefficient is considered as a function of the operational conditions:

$$C = f(\vec{x}) \tag{11}$$

We distinguish between two classes of surrogate methods for the estimation of global coefficients: direct and indirect methods. Direct methods directly estimate global coefficients from given operational conditions without considering the surface pressure distribution and the integration formulas. One of these methods is radial basis function interpolation (RBF) that obtains the prediction as

$$C(\vec{x}) = \sum_k w_k \varphi(||\vec{x} - \vec{x}_k||) \tag{12}$$

where w_k are learnable weights, φ is a radial function and $\vec{x}_k$ the operational conditions for the k-th sample in the training data. In this paper we allow the augmentation of the previous formulation with polynomials and also consider under this umbrella term Gaussian Process (GP) interpolation with fixed and variable exponents [23]. Another direct method is a multilayer perceptron for global predictions (MLPG) that calculates the global coefficient as

$$C(\vec{x}) = \mathrm{MLPG}(\vec{x}) = W_K \left(\cdots \left(W_2 \phi(W_1 \vec{x} + \vec{b}_1) + \vec{b}_2 \right) \cdots \right) + \vec{b}_K \tag{13}$$

where W_k and $\vec{b}_k$ are learnable weights and biases.

In contrast, indirect methods predict the surface pressure distribution and then use the physical equations to calculate the global coefficient. These methods have the benefit that they yield a consistent model that guarantees that the predicted surface pressure coefficient distributions are in agreement with the predicted global coefficient. They are aware of the integration formulas 7 and 10 that establish how the global coefficients are obtained from the surface pressure distribution, the operational conditions and the aircraft geometry. The prediction is hence done in two steps:

$$\vec{y} = f_\theta(\vec{x}) \tag{14}$$

$$C = T_{\mathrm{C}}(\vec{y}) \tag{15}$$

where $\vec{y}$ is the predicted distribution and θ represents learnable parameters. Proper orthogonal decomposition coupled with interpolation (PODI) is arguably the most common data-driven method used. Modes $\Phi \in \mathbb{R}^{n,d}$ and lower dimensional representations $Z \in \mathbb{R}^{d,m}$ are computed so that $\Phi \vec{z}_i \approx \vec{y}_i$, where d is the lower dimension, and m the number of training samples. The prediction for new parameters $\vec{x}$ is calculated as

$$\vec{y}(\vec{x}) = \Phi \vec{z}(\vec{x}) \tag{16}$$

$$C = T_{\mathrm{C}}(\vec{y}(\vec{x})) \tag{17}$$

where $\vec{z}(\vec{x})$ are POD coefficients obtained using an interpolation technique. We refer to the resulting model as PODI. The interest reader is referred to [24, 17, 18] for a more detailed description of this model. This work focuses on two other indirect methods, based on a multilayer perceptron, that are described next.

2.1 MULTILAYER PERCEPTRON FOR LOCAL PREDICTIONS

Following the work in [17, 19], in this approach the operational parameters, and the mesh coordinates and face normals are used to perform pointwise predictions of the pressure coefficients using a multilayer perceptron for local predictions (MLPL). The prediction of the surface pressure distribution for the parameters $\vec{x}$ is calculated as

$$\vec{y}(\vec{x}) = \left[\mathrm{MLPL}(\vec{x} \oplus \vec{c}_1 \oplus \vec{\nu}_1), \ldots, \mathrm{MLPL}(\vec{x} \oplus \vec{c}_n \oplus \vec{\nu}_n)\right]^\top \tag{18}$$

where $\vec{c}_i$ and $\vec{\nu}_i$ represent in the undeformed mesh the body-fixed coordinates of node i and the unit normal of the corresponding face, respectively. The loss function is given by

$$\mathcal{L}_{\mathrm{local}} = \mathrm{MSE}(\vec{y}(\vec{x}), \vec{y}_{\mathrm{true}}(\vec{x})) = \frac{\sum_{i=1}^{n} \left(\vec{y}(\vec{x})_i - \vec{y}_{\mathrm{true}}(\vec{x})_i\right)^2}{n} \tag{19}$$

After training the model, the predictions of the surface pressure distributions are used to predict the global coefficient as

$$C = T_{\mathrm{C}}(\vec{y}(\vec{x})) \tag{20}$$

2.2 MULTILAYER PERCEPTRON WITH GLOBAL COEFFICIENT REGULARIZATION

The previous model is an indirect method in which the integration formulas 7 and 10 are only used as a post-processing step of the predicted distribution. However, during training it disregards them, so that the model remains uniformed about the underlying physical relation between the local pressure coefficients and the global coefficient. That results in a model that is not directly optimized for the prediction of the desired global coefficient, so that certain deviations could arise. For example, if it occurs that the prediction error vector of the distribution $\Delta\vec{y}(\vec{x}) := \vec{y}(\vec{x}) - \vec{y}_{\mathrm{true}}(\vec{x})$ has a high correlation with the integration coefficients $\vec{a}_{\mathrm{C}}$ even small individual errors in the predicted distribution can sum up to a larger error $|\Delta C(\vec{x})|$ in the global coefficient. By way of illustration, at an angle of attack equal to zero and without surface deflections, if all surface pressure predictions at the top of the wing are higher while at the bottom they are lower than the ground truth values, then the predicted lift coefficient can become much lower than the actual one. This situation seems unlikely, nonetheless, the network can still develop some error biases and the incorporation of the integration formulas during training can help the model to adapt and become more accurate at the estimation of the global coefficient.

In this work we train MLPL as usual and then during the final epochs fine-tune it for the task of global coefficient prediction, by modifying the loss function to include an additional penalty for the deviation of the predicted global coefficients from the ground truth. As proposed in recent work [20], a global loss function and a penalty parameter are introduced. The global loss is multiplied by the penalty parameter and then added to the local loss to form the total loss function as expressed by

$$\mathcal{L}_{\mathrm{global}} = \mathrm{MSE}(C(\vec{x}), C_{\mathrm{true}}(\vec{x})) = \left(C(\vec{x}) - C_{\mathrm{true}}(\vec{x})\right)^2 \tag{21}$$

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{local}} + \lambda_{\mathrm{global}}\mathcal{L}_{\mathrm{global}} \tag{22}$$

We refer to the model obtained in this way as MLPPR, standing for multilayer perceptron with physics regularization. A summary of the direct and indirect surrogate methods is provided in Table 1.

Class	RBF	MLPG	PODI	MLPL	MLPPR
Direct	✓	✓			
Indirect			✓	✓	✓
Physics-aware training					✓

Table 1: Surrogate methods

3 TEST CASE

The test case is the NASA Common Research Model (NASA CRM), an industrial-relevant configuration resembling a modern commercial transport aircraft. High-fidelity RANS-CFD simulations were carried out with the DLR flow solver TAU [25], using the Spalart-Allmaras turbulence model. The DLR Surrogate Modeling for AeRo data Toolbox in python (SMARTy) [26] was used for the computation of the global coefficients and the construction of the surrogate models. The computational grid modeling the configuration without a vertical tailplane comprises approximately 43 million points and the corresponding surface grid, shown in Figure 2, consists of $n = 454,404$ surface points. The grid was derived based on DLR's experience during the AIAA Drag Prediction Workshop [27, 28] and is a slightly improved version of the fine grid used in [28]. Hence, results are comparable in accuracy when solving the RANS equations coupled with the Spalart–Allmaras turbulence model. The underlying coupled fluid–structure

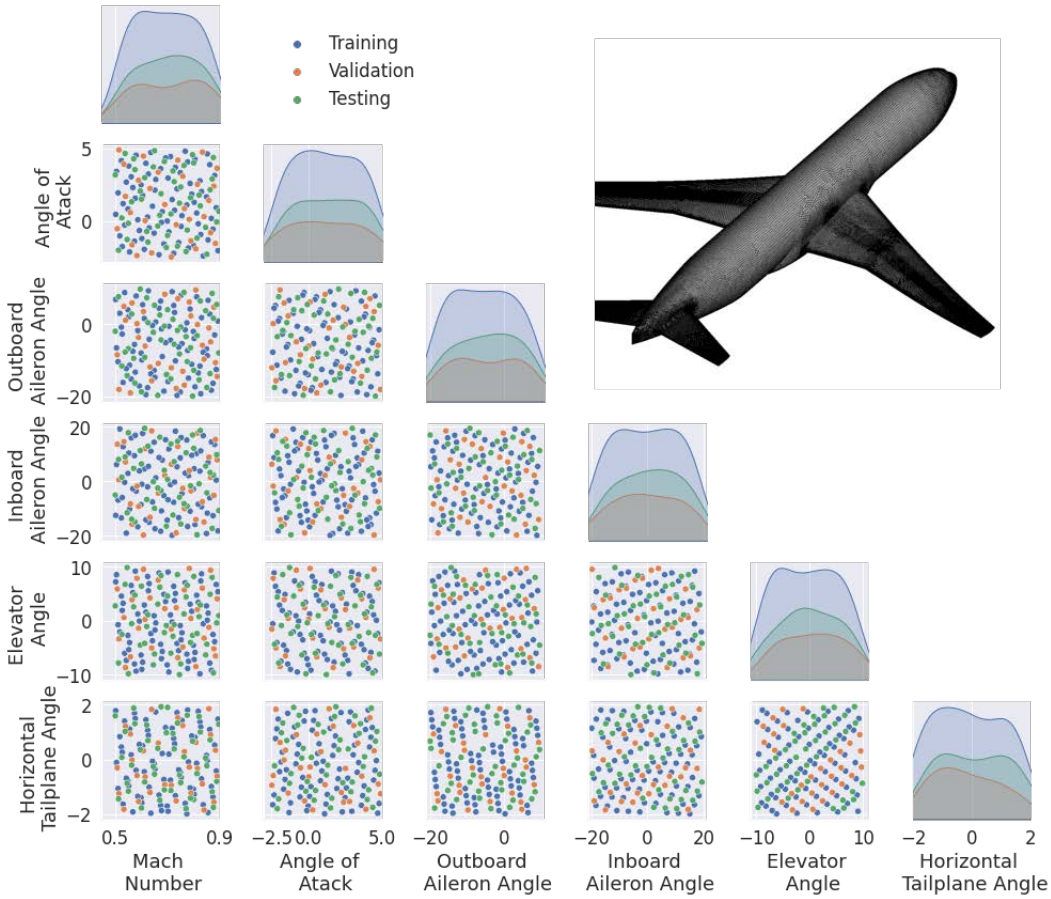


Figure 2: Pairplot of sampling strategy and computational mesh for the NASA CRM

simulation used to obtain the deformed 1g flight shape was compared to results from a European transonic wind-tunnel test campaign in [29]. A static deformation test was not part of the measurement campaign done for the AIAA Drag Prediction Workshop, and no direct experimental results are available. Note that, the flight shape is kept fixed in this paper. For further details, such as mesh convergence studies, turbulence model influence and comparison to experimental results as well as other codes, the interested reader is referred to the corresponding literature.

All CFD simulations were ran until a residual below $\mathrm{e}{-8}$ was reached. In addition to the density residual criterion, a Cauchy convergence criteria was employed for the lift and pitching coefficient with values of $\mathrm{e}{-3}$ and $5\mathrm{e}{-5}$, respectively, to accelerate convergence. The altitude was set to $37,000$ ft and the samples were parameterized by six independent parameters: Mach number, angle of attack, inboard aileron deflection angle, outboard aileron deflection angle, elevator deflection angle and horizontal tailplane deflection angle that we denote as $M, \alpha, \phi_{\mathrm{inAil}}, \phi_{\mathrm{outAil}}, \phi_{\mathrm{el}}, \phi_{\mathrm{htp}}$, respectively. A Halton sequence was used to produce 157 six-dimensional operational conditions, of which $m = 149$ were retained after running the CFD simulations. Out of the $m = 149$ total samples, the first 75 are selected for training, the next 30 are selected for validation, while the remaining 44 samples are used for testing. This partitioning of the data is shown in Figure 2.

4 RESULTS

This section presents the results for the task of predicting global coefficients for various operational conditions. Different models are built for the lift coefficient C_{L} than for the pitching moment coefficient C_{my}, and the results are presented separately. In order to simulate a production environment and evaluate the generalization capabilities of the models, the results are based on the testing samples, which were not seen by the models during training and validation. The metrics used to evaluate the performance of the models with respect to the test samples are the mean absolute error (MAE), the median of the absolute error (MedianAE) and the maximum absolute eror (MaxAE). Additionally, for the sake of analyzing lift and pitching moment curves, for each of the Mach numbers 0.54, 0.70, and 0.85, eight samples without control surface deflections were computed with angles of attack ranging from -2° to 5°.

In the case of the direct methods, a grid search approach was used to optimize the hyperparameters of RBF and MLPG. Tables 7 and 8 in the appendix display the optimal hyperparameters for these models. Using the selected hyperparameters, experiments for MLPG were repeated 10 times to account for the randomness of the training procedure and results display the obtained mean test metrics together with the standard deviation.

Regarding the indirect methods, the hyperparameters were selected as follows. For PODI, the hyperparameters are the same as those that were obtained in [19] for the prediction of the surface pressure distributions. The selected hyperparameters are shown in the appendix in Table 9. In the case of MLPL, the architecture of the model was chosen based on the previous best hyperparameters found in [19]. The batch size was changed to allow the training of one full snapshot simultaneously, so that the model processes $454,404$ points per batch. The learning rate, its decay factor and the number of epochs were changed accordingly to find a reasonable model. Table 10 shows the hyperparameters for this model.

For the MLPPR, the models start with the model state of MLPL and attempt to fine-tune the model for some additional epochs with the inclusion of the global loss. The model named MLPLC, standing for MLPL continued, refers to the model that is trained in an equal fashion as MLPPR but without the global loss. This is done in order to more fairly assess the benefit of the inclusion of the global loss penalty term, as the additional benefit could also stem alone

from training for more epochs. The optimal hyperparameters obtained using the framework Optuna [30] for MLPPR are shown in Table 11. As in the case of MLPG, experiments for MLPPR and MLPC were repeated 10 times.

4.1 LIFT COEFFICIENT

Table 2 shows the obtained metrics for all models, while Figure 3 displays the cumulative distribution of the absolute error. In order to show a more detailed view of the distribution, the results for MLPG, the worst performing model, are omitted in the plot. MLPL attains a MAE of 3.0×10^{-3}, while the MLPPR yields a MAE of 2.7×10^{-3}, representing a 10% decrease in this error metric. Meanwhile, MLPC, trained exactly as MLPPR but without the global loss term, obtains a MAE of 3.1×10^{-3} showing no improvement with respect to base model MLPL. The other models RBF, MLPG and PODI attain mean absolute errors of 6.1×10^{-3}, 13.3×10^{-3}, and 9.1×10^{-3}, representing an increase in this error metric of 103%, 343% and 203%. In all three error metrics MLPPR outperforms the other models, although the median of the absolute error remains unchanged with respect to MLPL. It can also be observed in the cumulative distribution of the absolute error how MLPL and MLPPR outperform the other models across almost all quantiles. Between MLPL and MLPPR the differences are much smaller and only minor improvements of MLPPR are present.

Regarding the prediction of the pressure coefficient distributions, MLPL achieves a mean absolute error of 12.3×10^{-3}, while on average MLPPR attains in this error metric a value of 12.4×10^{-3}. In comparison, PODI achieves a MAE of 19.8×10^{-3}, representing an increase of 61% with respect to MLPL. Hence, the MLPRR model is able to improve the lift coefficient prediction slightly compared to the MLPL model without negatively impacting the pressure distribution prediction.

Figure 4 shows the lift coefficient curves for the three selected Mach numbers for cases

Table 2: C_{L} test metrics

Metric	RBF	MLPG	PODI	MLPL	MLPPR	MLPC
MAE $\times 10^3$	6.1	13.3 ± 1.0	9.1	3.0	$\mathbf{2.7 \pm 0.1}$	3.1 ± 0.0
MedianAE $\times 10^3$	3.4	7.3 ± 1.0	4.9	**1.5**	$\mathbf{1.5 \pm 0.1}$	1.6 ± 0.0
MaxAE $\times 10^3$	30.1	83.6 ± 4.8	40.1	20.6	$\mathbf{19.2 \pm 0.3}$	21.1 ± 0.1

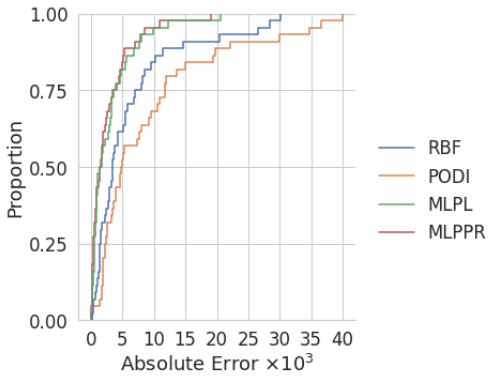


Figure 3: C_{L} absolute error cumulative distribution

Table 3: C_{L} mean absolute error $\times 10^3$

Mach	RBF	MLPG	PODI	MLPL	MLPPR
0.54	2.0	4.9	4.5	1.1	**1.0**
0.70	2.7	6.7	3.7	1.7	**1.5**
0.85	10.1	17.7	19.3	6.4	**6.2**

without control surface deflections. Table 3 shows for each Mach number the mean absolute errors over the eight angles of attack. In Figure 4a the Mach number corresponds to 0.54. All methods give accurate predictions and their mean absolute error is below 5×10^{-3}. MLPPR achieves the lowest MAE with a value of 1.0×10^{-3}, while MLPG attains the highest, 4.9×10^{-3}. MLPG is observed to underpredict at an angle of attack of $5°$, with an absolute error of 20.1×10^{-3}. In comparison, RBF obtains the lowest absolute error, 3.2×10^{-3}. At an angle of attack of $-1°$, shown in Figure 4d, PODI obtains the highest absolute error, 8.6×10^{-3}, while MLPPR obtains the lowest, 0.2×10^{-3}.

Figure 4b features a Mach number equal to 0.70, at the center of the sample space for this variable. For all methods, except PODI, the mean absolute error is greater than in the previous case. Again, MLPPR attains the lowest, 1.5×10^{-3}, and MLPG, the highest, 6.7×10^{-3}. As observed, MLPG overpredicts at an angle of $4°$, obtaining an absolute error of 18.5×10^{-3}, while MLPPR achieves the lowest error, 0.2×10^{-3}. At an angle of attack of $1°$, as shown in Figure 4e, MLPG deviates the most, with an absolute error of 4.7×10^{-3}, while MLPL is the closest to the ground truth value and attains an error of 0.2×10^{-3}. From a physical point of view, however, these differences are negligible since all model predictions except the MLPG provide accurate results which should suffice for nearly all follow on activities.

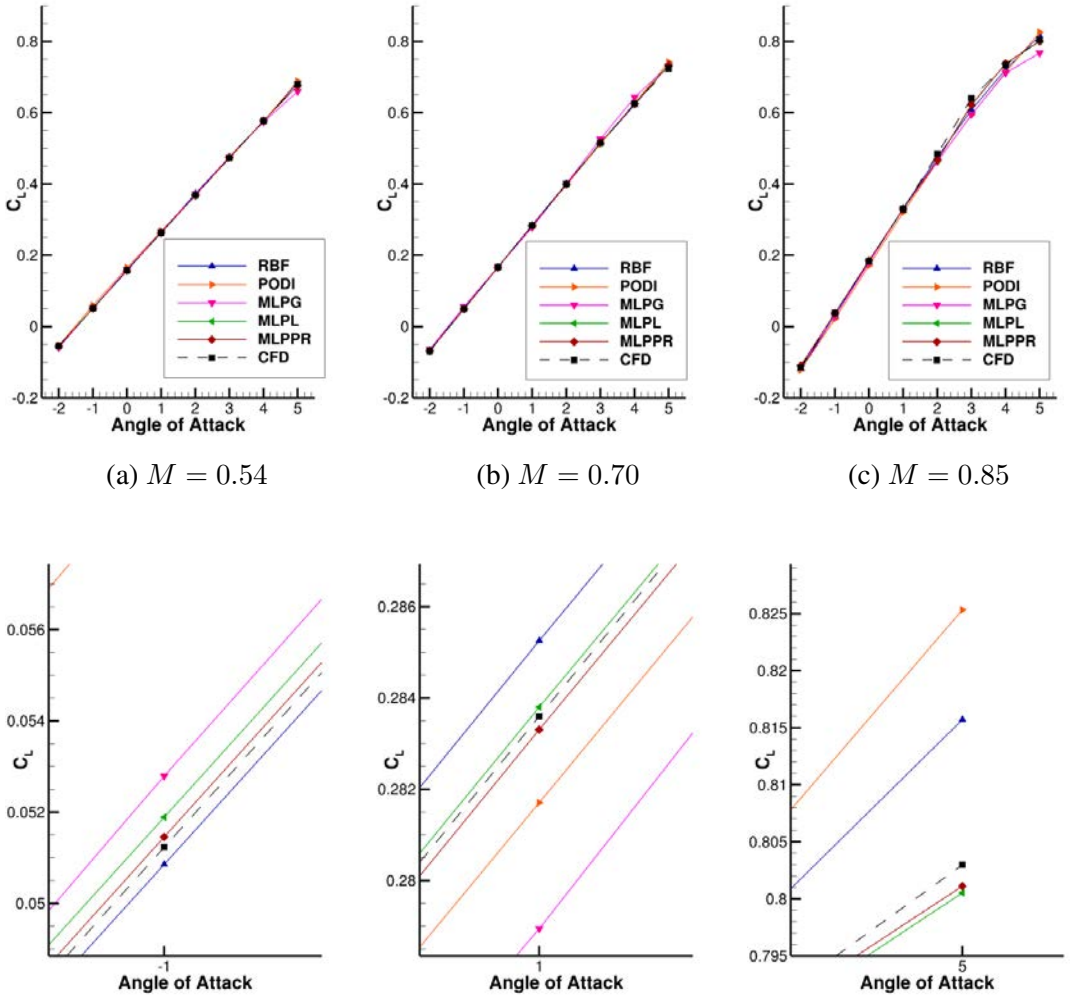
Figure 4c shows the lift curve for a Mach number equal to 0.85. For this Mach number, the mean absolute error is the highest for all models. MLPPR still attains the lowest mean absolute error, 6.2×10^{-3}, while PODI attains the highest, 19.3×10^{-3}. The predictions at an angle of attack of $3°$ differ significantly from the reference solution. At this angle of attack, PODI yields the highest absolute error, 46.7×10^{-3}, while MLPPR attains the lowest, 18.1×10^{-3}. It is also observed that MLPG deviates significantly for higher angles. At an angle of $5°$, shown in Figure 4f, MLPG deviates the most, with an absolute error of 35.7×10^{-3}. In comparison, MLPPR, attains the lowest error, 1.9×10^{-3}. In contrast to both previous cases, at this Mach number the MLPRR model has a benefit from a physical perspective as it indicates the lift decrease due to occurring separation on the upper wing surface and provides a more robust $C_{\mathrm{L,max}}$ prediction.

4.2 PITCHING MOMENT COEFFICIENT

Table 4 shows the obtained metrics for all models, while Figure 5 displays the cumulative distribution of the absolute error. Again the results for MLPG, the worst performing model, are omitted in the plot. The MLPL attains a MAE of 44.5×10^{-4}, while the MLPPR yields a MAE of 43.6×10^{-4}, representing a 2% decrease in this error metric. Meanwhile, MLPC, trained exactly as MLPPR but without the global loss term, obtains a MAE of 44.8×10^{-4} showing no improvement with respect to the MLPL base model. The other models RBF, MLPG and PODI attain mean absolute errors of 46.1×10^{-4}, 102.6×10^{-4}, and 55.6×10^{-4}, representing an increase in this error metric of 4%, 131% and 25%. MLPPR outperforms all other models

with respect to the mean absolute error, while MLPLC attains the lowest median absolute error and RBF the lowest maximum absolute error. However, it can be observed in the cumulative distribution of the absolute error that MLPL and MLPPR outperform RBF in most quantiles.

Figure 6 shows the pitching moment coefficient curves for the selected Mach numbers. Table 5 shows for each Mach number the mean absolute error over the eight angles of attack. In Figure 6a the Mach number corresponds to 0.54. MLPL achieves the lowest mean absolute



(a) $M = 0.54$ (b) $M = 0.70$ (c) $M = 0.85$

(d) $M = 0.54, \alpha = -1°$ (e) $M = 0.70, \alpha = 1°$ (f) $M = 0.85, \alpha = 5°$

Figure 4: Comparison of the predicted lift coefficients to the CFD reference samples without control surface deflections

Table 4: C_{my} test metrics

Metric	RBF	MLPG	PODI	MLPL	MLPPR	MLPLC
MAE $\times 10^4$	46.1	102.6 ± 8.7	55.6	44.5	$\mathbf{43.6 \pm 0.0}$	44.8 ± 0.0
MedianAE $\times 10^4$	33.7	59.6 ± 10.9	35.9	23.7	25.0 ± 0.1	$\mathbf{23.6 \pm 0.0}$
MaxAE $\times 10^4$	**163.7**	683.9 ± 112.4	465.1	301.7	297.3 ± 0.3	301.5 ± 0.1

error, 8.9×10^{-4}, while MLPG yields the highest, 39.2×10^{-4}. MLPG is observed to deviate significantly at various angles of attack. It is also noticeable that RBF overpredicts at an angle of $5°$, attaining an absolute error of 147.1×10^{-4}. In comparsion, MLPPR achieves the lowest error, 7.6×10^{-4}. At an angle of attack of $3°$, as shown in Figure 6d, MLPG has the highest absolute error of 43.0×10^{-4}, while MLPL has the lowest, 3.5×10^{-4}. At an angle of attack of $-2°$, as shown in Figure 6e, MLPG presents the highest error, 49.8×10^{-4}, while MLPL attains the lowest error, 1.2×10^{-4}.

Figure 6b features a Mach number equal to 0.70. For all methods, except MLPG, the mean absolute error is greater than in the previous case. PODI achieves the lowest mean absolute error, 21.6×10^{-4}, while MLPG attains the highest, 31.7×10^{-4}. At an angle of attack of $5°$ the models show higher deviations from the ground truth value. PODI presents the highest absolute error, 104.1×10^{-4}, while MLPG the lowest, 38.4×10^{-4}. Throughout all models except the MLPG at some specific angles of attack yield reasonable results for following aerodynamic and fight mechanic analysis with correct trends as well as accurate enough moment estimates.

Figure 6c shows the pitching moment curve for a Mach number equal to 0.85. As in the case of lift, this Mach number proves to be the most challenging and the mean absolute error is the highest for all models. This directly results from the underlying physics with shock induces separation and reverse shock motion especially at high angles of attack. MLPR achieves the lowest mean absolute error, 67.7×10^{-4}, while MLPG attains the highest, 150.0×10^{-4}. The predictions at an angle of attack of $3°$ differ significantly from the reference solution, as the pitching moment starts deviating from the linear trend due to an underlying inverse shock motion. At this angle of attack, PODI yields the highest absolute error, 303.5×10^{-4}, while MLPL attains the lowest, 185.7×10^{-4}. For higher angles of attack, significant deviations can also be observed for RBF, MLPG and PODI. At an angle of $4°$, shown in Figure 6f, RBF has the higest error, 118.0×10^{-4}, while MLGP attains the lowest, 12.3×10^{-4}. For an angle of $5°$ the

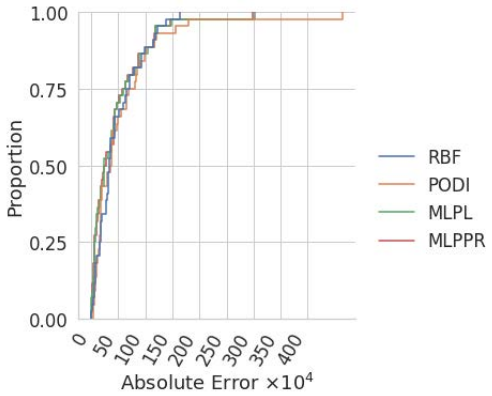
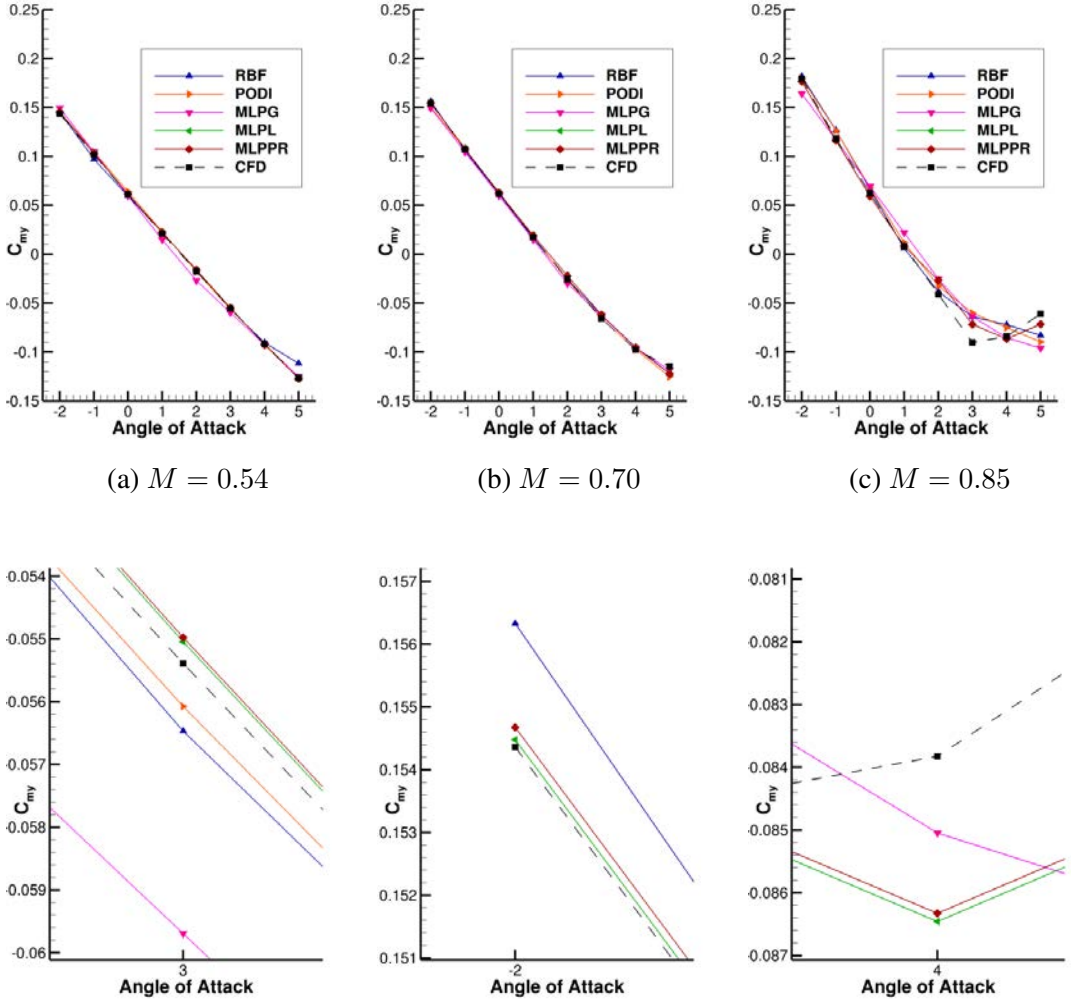


Figure 5: C_{my} absolute error cumulative distribution

Table 5: C_{my} mean absolute error $\times 10^4$

Mach	RBF	MLPG	PODI	MLPL	MLPPR
0.54	34.7	39.2	14.9	**8.9**	10.1
0.70	22.2	31.7	**21.6**	24.9	25.7
0.85	99.7	150.0	118.1	67.9	**67.7**



(a) $M = 0.54$ (b) $M = 0.70$ (c) $M = 0.85$

(d) $M = 0.54, \alpha = 3°$ (e) $M = 0.70, \alpha = -2°$ (f) $M = 0.85, \alpha = 4°$

Figure 6: Comparison of the predicted pitching moment coefficients to the CFD reference samples without control surface deflections

indirect deep learning models are the only ones able to capture the reversing trend in pitching moment increase.

4.3 Computational Cost

For practical applications it is not only important that the data-driven models provide accurate predictions of aerodynamic quantities, but also that they do so in significantly less time than the time-consuming high fidelity CFD computations. Therefore, we provide an overview in Table 6 of the training and evaluation times of the models to help assess the feasibility of their use in time-critical scenarios. RBF and PODI were trained exclusively on an Intel(R) Xeon(R) W-2135 CPU @ 3.70GHz with 12 cores. On the other hand, MLPG was trained using an NVIDIA Quadro P4000 8BG GPU, while MLPL and MLPPR were trained on a NVIDIA A100 40 GB GPU as they have a higher memory requirement. The two direct methods, RBF and MLPG, take for training 4 and 25 seconds, respectively. PODI is the indirect method that requires the least training time, taking only 7 seconds. On the other hand, the training of MLPL

takes about 6 hours and MLPPR requires the same time plus 180 seconds for the fine-tuning of the model. The evaluation time for all models is significantly under one second and is thus negligible with respect to the CFD simulation wall time of 19 hours using 64 cores on DLR's High Performance Computing System CARA. The direct methods, as they do not predict the surface pressure coefficient distribution, are the ones that compute the global coefficient prediction the fastest. For PODI the evaluation time is dominated by the computation of the global coefficient, while for MLPL and MLPPR the computation of the pressure coefficient distribution takes most of the prediction time.

Table 6: Computational cost in hours for C_{L} models

	CFD	RBF	MLPG	PODI	MLPL	MLPPR
Training	–	1×10^{-3}	7×10^{-3}	2×10^{-3}	6	6
One Evaluation	19	6×10^{-8}	3×10^{-7}	2×10^{-5}	5×10^{-5}	5×10^{-5}

5 CONCLUSIONS

This work studied the effect that is achieved in the prediction of global coefficients with the use of a recently proposed physics-based regularization term that penalizes a multilayer perceptron for local predictions (MLPL) of surface pressure coefficients when the predicted distributions yield global coefficients that deviate from the ground truth. This approach (MLPPR) was tested for an industrial relevant 3D configuration known as the NASA Common Research Model (NASA CRM) transport aircraft. It was compared to the same model without the regularization term (MLPL), and to proper orthogonal decomposition coupled with interpolation (PODI). These were considered as indirect methods that first predict a surface pressure coefficient distribution and use the available integral formulas to calculate the global coefficients. A comparison was also made to two direct methods that only yield predictions of global coefficients, namely, radial basis function interpolation (RBF) and multilayer perceptron for global predictions (MLPG).

We found that the deep learning indirect methods, MLPL and MLPPR, provide in general the best predictions for the lift and pitching moment coefficients. When using the physics-based regularization term for the fine-tuning of a trained multilayer perceptron, only limited improvements are achieved in the predictions of the lift and pitching moment coefficients. PODI, despite being trained to predict the pressure coefficient distribution as well, did not achieve the same accuracy as MLPL and MLPPR. With regards to the direct methods, the training of a multilayer perceptron to predict the global coefficients proved to be challenging given the relative small amount of samples. Hence, in our experiments this network yielded the highest errors. In contrast, the radial basis function interpolation approach was able to give more accurate predictions, despite being trained with the same amount of samples as the multilayer perceptron MLPG. Regarding computational cost, all methods are capable of evaluating new samples in nearly real-time. RBF is the method with the least training and evaluation time, while MLPL and MLPPR require much more training time, since they predict the pressure coefficient distribution but feature comparable evaluation times.

Future research could focus on the use of the studied physics-based regularization to train the multilayer perceptron from scratch, instead of as a fine-tuning technique. Moreover, the in-

corporation of adaptive weighting mechanisms to dynamically adapt the penalization parameter of the global loss is a promising direction to pursue. In addition, more complex architectures such as graph neural networks could also be extended with a physics-based regularization.

6 ACKNOWLEDGEMENT

The funding of parts of these investigations from the LUFO VI-1 project SMARTfly (SMARTfly - Smart Modeling of flying Transport Vehicles - Entwicklung exakter und effizienter Modellierungen und Simulationsmethoden für den Entwurf von Fluggeräten und Triebwerken, FKZ: 20X1909A) by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) is gratefully acknowledged.

This work was supported by the Helmholtz Association's Initiative and Networking Fund on the HAICORE@KIT partition.

REFERENCES

1. Lucia, D. J., Beran, P. S. & Silva, W. A. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences* **40,** 51–117. ISSN: 0376-0421. https://www.sciencedirect.com/science/article/pii/S0376042103001131 (2004).
2. Bui-Thanh, T, Damodaran, M. & Willcox, K. *Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics* in *21st AIAA Applied Aerodynamics Conference* (2003), 4213.
3. Iuliano, E. & Quagliarella, D. Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. *Computers & Fluids* **84,** 327–350 (2013).
4. Franz, T., Zimmermann, R., Görtz, S. & Karcher, N. Interpolation-based reduced-order modelling for steady transonic flows via manifold learning. *International Journal of Computational Fluid Dynamics* **28** (Mar. 2014).
5. Fossati, M. Evaluation of aerodynamic loads via reduced-order methodology. *AIAA Journal* **53,** 2389–2405 (2015).
6. Ripepi, M. *et al.* Reduced-order models for aerodynamic applications, loads and MDO. *CEAS Aeronautical Journal* **9,** 171–193 (2018).
7. Bengio, Y. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning* **2,** 1–127. https://doi.org/10.1561/2200000006 (2009).
8. Santos, M., Mattos, B. & Girardi, R. *Aerodynamic Coefficient Prediction of Airfoils Using Neural Networks* in *46th AIAA Aerospace Sciences Meeting and Exhibit* (American Institute of Aeronautics and Astronautics, Jan. 2008). https://doi.org/10.2514/6.2008-887.
9. Yilmaz, E. & German, B. *A Convolutional Neural Network Approach to Training Predictors for Airfoil Performance* in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (American Institute of Aeronautics and Astronautics, June 2017). https://doi.org/10.2514/6.2017-3660.
10. Kou, J. & Zhang, W. Layered reduced-order models for nonlinear aerodynamics and aeroelasticity. *Journal of Fluids and Structures* **68,** 174–193. ISSN: 0889-9746. https://www.sciencedirect.com/science/article/pii/S0889974616301852 (2017).

11. Rozov, V. & Breitsamter, C. Data-driven prediction of unsteady pressure distributions based on deep learning. *Journal of Fluids and Structures* **104,** 103316. ISSN: 0889-9746. https://www.sciencedirect.com/science/article/pii/S0889974621000992 (2021).
12. Guo, X., Li, W. & Iorio, F. *Convolutional Neural Networks for Steady Flow Approximation* in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, San Francisco, California, USA, 2016), 481–490. ISBN: 9781450342322. https://doi.org/10.1145/2939672.2939738.
13. Jin, X., Cheng, P., Chen, W.-L. & Li, H. Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Physics of Fluids* **30,** 047105. https://doi.org/10.1063/1.5024595 (Apr. 2018).
14. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K. & Kaushik, S. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics* **64,** 525–545 (2019).
15. Thuerey, N., Weißenow, K., Prantl, L. & Hu, X. Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows. *AIAA Journal* **58,** 25–36 (2020).
16. Wu, H., Liu, X., An, W., Chen, S. & Lyu, H. A deep learning approach for efficiently and accurately evaluating the flow field of supercritical airfoils. *Computers and Fluids* **198,** 104393. ISSN: 0045-7930 (2020).
17. Campomanes, C. S., Stürmer, P. & Bekemeyer, P. *Fast Predictions of Aircraft Aerodynamics using Deep Learning Techniques* in *AIAA Aviation 2021 Forum* (Aug. 2021).
18. Hines, D. & Bekemeyer, P. *Data-driven reduced order modeling for aerodynamic flow predictions* in *Eccomas Congress 2022* (2022). https://elib.dlr.de/189313/.
19. Hines, D. & Bekemeyer, P. Graph Neural Networks for the Prediction of Aircraft Surface Pressure Distributions. *Under review* (2023).
20. Dias Ribeiro, M., Stadtner, M. & Bekemeyer, P. Unsteady Reduced Order Model with Neural Networks and Physics-informed Regularization for Aerodynamic Applications. *Under review* (2023).
21. Andres, E. & Paulete-Periáñez, C. On the application of surrogate regression models for aerodynamic coefficient prediction. *Complex Intelligent Systems* **7** (Mar. 2021).
22. Balla, K., Sevilla, R., Hassan, O. & Morgan, K. An application of neural networks to the prediction of aerodynamic coefficients of aerofoils and wings. *Applied Mathematical Modelling* **96,** 456–479. ISSN: 0307-904X. https://www.sciencedirect.com/science/article/pii/S0307904X21001426 (2021).
23. Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. Design and Analysis of Computer Experiments. **4,** 409–423 (1989).
24. Sirovich, L. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics* **45,** 561–571 (1987).
25. Schwamborn, D., Gerhold, T. & Heinrich, R. The DLR TAU-code: recent applications in research and industry (2006).

26. Bekemeyer, P. *et al. Data-Driven Aerodynamic Modeling Using the DLR Smarty Toolbox* in *AIAA Aviation 2022 Forum* (June 2022).
27. Keye, S. & Mavriplis, D. J. *Summary of Data from the Sixth AIAA CFD Drag Prediction Workshop: Case 5 (Coupled Aero-Structural Simulation)* in *55th AIAA Aerospace Sciences Meeting* (American Institute of Aeronautics and Astronautics, Jan. 2017). https://doi.org/10.2514/6.2017-1207.
28. Keye, S., Togiti, V. K. & Brodersen, O. P. *DLR Results of the Sixth AIAA Computational Fluid Dynamics Drag Prediction Workshop* in *35th AIAA Applied Aerodynamics Conference* (American Institute of Aeronautics and Astronautics, June 2017). https://doi.org/10.2514/6.2017-4232.
29. Keye, S. & Rudnik, R. *Validation and Assessment of Turbulence Model Impact for Fluid-Structure Coupled Computations of the NASA CRM* in *5th CEAS Air & Space Comference* (2015). https://elib.dlr.de/102153/.
30. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. *Optuna: A Next-generation Hyperparameter Optimization Framework* in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019).

7 APPENDIX

Table 7: RBF hyperparameters

Hyperparameter	C_{L}	C_{my}
interpolator	Gaussian	Gaussian Variable Exponent
augmenatation	Linear	Quadratic
scale	True	True
regularization	True	False

Table 8: MLPG hyperparameters

Hyperparameter	C_{L}	C_{my}
Initial learning rate	1.0×10^{-3}	1.0×10^{-3}
LR decay factor	0.999	0.999
Epochs	2000	2000
Batch size	15	15
Dimension of hidden layers	16	16
Hidden layers	5	6
Activation	ReLU	ReLU

Table 9: PODI hyperparameters

Hyperparameter	Value
mode retention	99.0%
interpolator	Gaussian Variable Exponent
augmentation	Linear
scale	False
regularization	True

Table 10: Base MLPL hyperparameters

Hyperparameter	Value
Initial learning rate	1.0×10^{-3}
LR decay factor	0.999
Epochs	5000
Batch size	$454,404$ (1 snapshot)
Dimension of hidden layers	128
Hidden layers	12
Activation	ELU

Table 11: MLPPR continuation hyperparameters

Hyperparameter	C_{L}	C_{my}
Learning rate	1.0×10^{-6}	1.0×10^{-6}
Global penalty $\lambda_{\mathrm{global}}$	1.0×10^{3}	1.0×10^{4}
Epochs	25	25
Trained Layers	All	Last 2