Steady–state Flowfield Prediction in Transonic Regime Via Deep–Learning Framework

Gabriele Immordino*, Andrea Da Ronch†
Faculty of Engineering and Physical Sciences
University of Southampton, Southampton, United Kingdom, SO17 1BJ

Marcello Righi‡
School of Engineering
Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland, 8401

This article focuses on the development of a deep–learning framework for predicting distributed quantities around aircraft flying in the transonic regime. These quantities play a crucial role in determining aerodynamic loads and conducting aeroelastic analysis. Angle of attack and Mach number are chosen as the two independent parameters for the reduced–order models. A comparative assessment is conducted between the proposed deep-learning framework and the Proper Orthogonal Decomposition approach to identify strengths and weaknesses of each method. The accuracy of the data–driven machine–learning method in modelling transonic, steady–state aerodynamics is assessed against three benchmark cases of three-dimensional wings. Despite the challenges of the analyzed scenarios, promising results are obtained for each test case, showing the effectiveness of the model implemented. Furthermore, the article demonstrates the application of the method for aeroelastic analysis and uncertainty quantification. This quantifies robustness and versatility of the implemented model.

Nomenclature

Acronyms

$FCNN$	=	fully–connected neural network
LHS	=	Latin Hypercube Sampling
ML	=	machine–learning
$MAPE$	=	mean absolute percentage error
MSE	=	mean squared error
$NIPCE$	=	Non-Intrusive Polynomial Chaos Expansion

*PhD Student.
†Associate Professor, AIAA Senior Member.
‡Professor, AIAA Member, Lecturer at Federal Institute of Technology Zurich ETHZ

PDF = Probability Density Function

$POD+I$ = proper orthogonal decomposition with interpolation

ROM = reduced–order model

UQ = uncertainty quantification

Symbols

AoA = angle of attack, deg

c = mean chord, m

C_D = drag coefficient

C_F = skin friction coefficient

C_L = lift coefficient

C_{My} = pitching moment coefficient

C_P = pressure coefficient

M = Mach number

Re = Reynolds number

θ = elastic twist angle, deg

I. Introduction

Computational Fluid Dynamics (CFD) is widely used in engineering applications involving flow analysis. Despite a steady increase in available computational power, numerical simulations of turbulent flows, multiscale and multiphysics phenomena remain difficult to tackle, requiring millions of degrees of freedom and several days of computation on high performance computing architectures. In this perspective, a solution may be represented by reduced–order models (ROMs) which are a set of theoretical and numerical techniques enabling the reconstruction and represention of the nonlinear dynamic behaviour of a system with a small number of degree of freedom (DOF). They can provide a more straightforward formulation of a nonlinear problem, resulting in a better understanding of the physical phenomena as well as a significant reduction in computational time.

The ability to build lower–order, computationally efficient representations of higher–order dynamical systems is one of the most essential properties of ROMs. Using these representations, ROMs can efficiently model flow fields while using significantly less parameters. Conventional ROMs accomplish this by linearly projecting higher–order manifolds to lower–dimensional space using dimensionality reduction techniques, such as proper orthogonal decomposition (POD) or the dynamic mode decomposition (DMD). With these ROMs our understanding in different fluid flow areas has deepened and enabled us to control flow phenomena at low computational costs [1, 2]. However, linear models

cannot represent with adequate accuracy flow characterized by nonlinear phenomena, such as the shock waves and viscous effects herein considered, without significant tuning and a high number of modes. In fact, POD, Isomap and manifold learning proved to be an efficient first approach towards prediction of steady, turbulent aerodynamic flows [3–7]. Nonetheless, due to their inherently simplified formulation, they were not capable of accurately predicting strong discontinuities, such as sudden changes in pressure coefficient caused by shock waves.

To address this problem, the use of novel nonlinear ROM might be exploited, such as machine–learning techniques and deep–learning approaches. The latter introduce nonlinearity in the activation function which allows extracting key features from diverse types of data and achieving noteworthy results in various fields of research, including fluid dynamics [8–12]. Successful application of fully–connected neural network (FCNN) architecture may be found in the prediction of the dynamic response of a nonlinear system in subsonic [13] and transonic regime [14] as well as in aerodynamic loads estimation during dynamic stall [15]. Also, deep dense networks can serve as surrogate models for an efficient aircraft design and optimization [16, 17]. Before a standalone use of ANN for steady–state analysis, Zhiwei et al. [18] opted for a combination of POD, domain decomposition and ANN for predicting flow fields of transonic airfoils. The domain decomposition technique separates the hard-to-predict regions from the full field and POD has been adopted in the regions individually. The interpolation of the POD modes outside the training region was achieved with ANN. Du et al. [19] and Espinosa et al. [20] introduced a novel surrogate aerodynamic framework that enables rapid airfoil shape optimization. Their proposed approach utilizes data-driven neural network model, with the airfoil geometric parameters and flight conditions serving as the surrogate inputs. This framework represents a significant advancement in the field, offering a more efficient and effective approach to aerodynamics prediction. Andres et al. [21] conducted a thorough review of various algorithms used for predicting aerodynamic data. They concluded that two crucial factors for enhancing the accuracy of such predictions are machine–learning techniques and model parameters optimization. Castellanos et al. [22] demonstrated that deep neural networks outperform traditional linear techniques and highlighted the importance of ROM for problems where the data dimensionality is significantly larger than the parameters that influence the prediction process. The surrogate model developed from Zhou et al. [23] represents one of the most recent noteworthy results in this field. Their approach involves combining Convolutional Neural Network (CNN) with a Residual Network (ResNet) to reconstruct transonic flow field with strong discontinuity, based on inputs from flow conditions and aerodynamic shape. Through this work, the authors once again demonstrated the remarkable ability of ANN to effectively handle nonlinearities. In a similar vein, Massegur et al. [24] conducted a study comparing a local POD model with a Deep Neural Network (DNN) architecture embedded in a Convolutional Auto-Encoder for aerofoil icing predictions. This comparison underscores the robustness of neural network architectures in handling substantial variations in design outputs across the icing envelope. Physics–Informed Neural Networks (PINNs) have shown promising capabilities in flowfield prediction, by incorporating a physical term into the loss function during the network training process [25–27]. Even though further studies are required to thoroughly validate and assess

the potential of these methods, these approaches effectively combine the power of deep–learning with the governing equations of the underlying physics, enhancing the accuracy and reliability of flow predictions.

Keeping in mind these remarkable applications, the current study aims at developing an optimized deep–learning approach able to predict distributed quantities around aircraft flying in transonic regime. A comparative assessment of the proposed deep–learning framework with POD approach is introduced in order to highlight the robustness of the present model with respect to traditional well–established methods.

This research is rooted from the work of Sabater et al. [17] for transonic flowfield predictions, which laid the foundation for our current study. The innovation in this article lies in the utilization of a machine–learning approach designed to effectively manage a spectrum of physical phenomena and diverse geometric configurations within three established test cases involving 3D wings in transonic regime. Additionally, the research extends to the validation of the model robustness and versatility through an analysis of two practical applications. This dual focus on addressing complex scenarios and real-world applications underscores the unique contributions and broad applicability of the proposed machine–learning methodology.

The paper is organized as follows. Section II presents the implemented methodology. In Section III, the test cases are described, and the effectiveness of the proposed deep–learning framework is assessed. Section IV introduces the application of the proposed method for aeroelastic analysis and uncertainty quantification. Section V offers a summary of the findings and conclusions drawn from this study.

II. Methodology

This Section describes the various numerical techniques used to generate the reference simulation data and to build the reduced–order models.

A. Computational Fluid Dynamics Solver

The dataset used for the ROM approximations has been generated through Computational Fluid Dynamics (CFD) simulations. Reynolds-averaged Navier–Stokes (RANS) equations are discretized using SU2 v7.2.1 [28], a cell-centered finite volume method software that converts the partial differential equations into a set of ordinary differential equations. For all test cases, the one equation Spalart-Allmaras turbulence model is used for closure of the RANS equations. The convergence method is set to Cauchy method applied to the lift coefficient, considering a variation of 10^{-7} on the last 100 iterations. A $1v$ multigrid scheme is adopted for accelerating the convergence of CFD simulations. For the discretization of convective flows, the JST central scheme with artificial dissipation is employed. Gradients of flow variables are computed using the Green Gauss method. The linear solver chosen is the biconjugate gradient stabilization, with ILU preconditioner applied.

B. Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is a linear technique for dimensionality reduction. It was widely used for studying nonlinear problems, due to its capacity to extract, from a set of high–dimensional snapshots, the basis minimizing the error between the original snapshots and their orthogonal projection. This method consists of two steps. First, it generates a basis of orthonormal vectors, called modes, for the high–dimensional space of the data. Then, the original system is projected onto a low–dimensional space obtained by truncating the orthonormal basis to retain only the first few POD modes which embody the target energy of the system and express the essential information of data. POD may be formulated by taking the singular value decomposition of the snapshots matrix $\mathbf{X}$, which is either the pressure coefficient or skin friction coefficient surface field in our case study:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \sum_{i=1}^{n} \sigma_i u_i v_i^* \tag{1}$$

where the matrix $\mathbf{U}$ contains the spatial correlations in the data, $\mathbf{V}$ the temporal information and $\mathbf{\Sigma}$ the orthonormal eigenmodes. In the present work, truncation of the number of modes takes place by choosing a reduced rank $k < n$ so that only the k POD modes which account for 95% of the total energy are conserved upon reconstruction. Several approaches exist for determining the unknown POD coefficients within the latent space. In our investigation, we opted for a cubic interpolation technique, striking a balance between computational efficiency and accuracy [29, 30]. Henceforth, we employ the terminology POD+Interpolation (POD+I) to denote the POD coefficients obtained in the in the two–dimensional parameter space (Mach number and angle of attack) through cubic spline interpolation.

C. Artificial Neural Networks

Artificial Neural Network (ANN) is the most popular nonlinear tool in machine–learning which is usually used in supervised learning. It is a mathematical model inspired by the architecture of neural network in human brain. As in the brain, the signals are received by neurons, manipulated and passed to other neurons, as well as in ANN a neuron receives inputs through its connections; these are weighted, summed and passed through a nonlinear function called activation function. The output of each neuron can be expressed as:

$$y_{l,n} = f_l(w_{l,n} \cdot x_l + b_{l,n}) \tag{2}$$

where l denotes the layer number, n the neuron number in that layer, $w_{l,n}$ the weights and $b_{l,n}$ the constant term of the linear mapping for each neuron. The function $f_l()$ applied to the sum is the nonlinear activation function. The fully–connected neural network (FCNN), also called dense neural network or multilayer perceptron, is the most simple neural network architecture, where each neuron from one layer is fully–connected to all neurons in the other layer as shown in Figure 1.

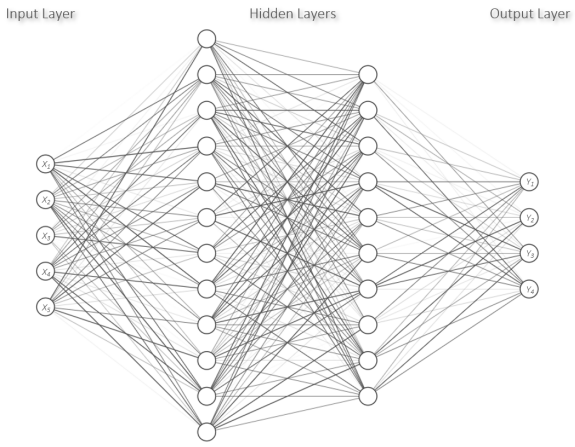


Fig. 1 Fully–connected network architecture.

In terms of layers our neural network is made of an input layer, which takes normalized input from the domain, hidden layers, that take input from another layer and pass output to another layer, and output layer, which makes a prediction.

The activation function is a crucial component of a neural network. It is connected to each neuron and defines how the weighted sum of the input is transformed into an output from a node in a layer of the network. The choice of activation function has a large impact on the capability and performance of the neural network, and, the best choice could differ according to the problem involved. Therefore, three types of activation function have been tested: `TanH`, `ReLu` and `LeakyReLu`. The Leaky Rectified Linear unit (`LeakyReLu`) generalizes the classic `ReLu` with a slope for negative values. This activation function helps to avoid dead neurons in the negative region as the gradient of the negative values is a non–zero value. In this case, the negative slope was set to $a = 0.02$ after a careful assessment aimed at achieving optimal performance.

$$f(x) = \max(ax, x) \tag{3}$$

Mean Squared Error (`MSE`) loss function was chosen to evaluate how the machine–learning algorithm models the data. It is a function that quantifies the difference between the predicted output of the algorithm and the true output. From the loss function, it is possible to derive the gradients which are used to update the weights. ADAptive Moment estimation (`Adam`) [31] was employed during the back–propagation phase for optimising neural network weights and minimising `MSE` loss function.

The dataset was divided into 3 distinct sets: the training set, containing the observations used to train the network, the validation set used to optimise the hyperparameters and avoid overfitting and the test set used to verify the generalisation ability of the network. Hyperparameters refer to variables that are not learned from the data but are set prior to the training process. They are essential for defining the architecture and behavior of the neural network.

The ROM was implemented in the deep–learning python library `TensorFlow/Keras`. The choice of neural network architecture fell on fully-connected neural network because it allows the algorithm to predict one specific surface value at a time, while taking into account the rest of the surface field. A schematic of the deep–learning framework is reported in Figure 2. Here, it is possible to notice that the ROM utilizes freestream conditions and mesh coordinates as input. From there, the ROM identification process takes place with the deep-learning framework. The output is a pointwise prediction of pressure coefficient and skin friction coefficient components.

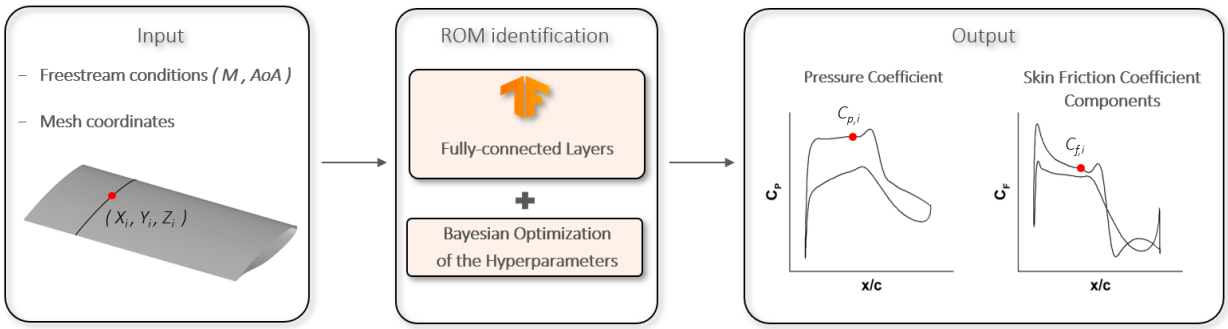


Fig. 2 Schematic of the deep–learning framework.

1. Bayesian optimization

An important step for improving the prediction capability of the model is the choice of a suitable set of hyperparameters. An optimization algorithm, which is able to explore the wide design space, is essential. Here, the goal of the algorithm is to optimize five hyperparameters: learning rate, batch size, activation function, number of neurons and layers. All the design variables are continuous, except for the activation function which is discrete. The design space for hyperparameters, including the possible values and step size for each continuous and discrete variable, is presented in Table 1. The selected continuous variables consist of the learning rate, number of hidden layers, number of neurons per hidden layer, and batch size. Additionally, activation function is chosen as discrete variable. The chosen ranges were intentionally set to be sufficiently large. Indeed, throughout the optimization process, it was observed that the hyperparameters converged to values below the upper limit of the specified ranges.

A Bayesian optimization strategy was employed for optimising the hyperparameters chosen. This automated technique handles the entire hyperparameter space as a black box, making it possible to determine the influence of the hyperparameters on the result without the need of manual studies or conventional sensitivity analyses.

The advantage of this type of optimization is that it creates a surrogate probability model of the objective function and finds the hyperparameters that perform best on the surrogate. It does not search the design space for every possible combination, but it follows an iterative process, where it starts by choosing the first hyperparameters randomly and then optimizes the next selection based on the performance of those ones in terms of best score on the validation set

metric. The hyperparameters found are used to evaluate the objective function, and the algorithm runs till the desired results are not achieved. Moreover, this Bayesian optimization approach treats discrete variables as continuous, first applying the optimization procedure and then rounding the suggested continuous point to the nearest discrete value in the search space before function evaluations. In our work, Keras-Tuner [32] seamlessly integrates Bayesian optimization with a underlying Gaussian process model [33, 34] into the hyperparameter search process. The acquisition function used is upper confidence bound (UCB) [34, 35]. 100 trials were run in order to ensure an adequate convergence of the Bayesian optimization strategy to the optimal set of hyperparameters. A trial indicates a full network training with specific hyperparameters. At the same time, we constrained the optimization algorithm by imposing 500 epochs on each trial to find the best architecture in order to limit the computational cost of each training phase. An epoch refers to a single complete pass through the entire training dataset where the network updates its weights and biases based on the calculated error. After the optimization phase is completed, the classic training procedure of the neural network takes place and the optimal weights, which minimize the loss function, are estimated.

Hyperparameter	**Value**	**Step size**
Learning rate	10^{-4} to 10^{-6}	$5 \cdot 10^{-6}$
Number of Hidden Layers	1 to 8	1
Number of Neurons per Hidden Layer	4 to 204	8
Batch size	1 to 8	1
Activation function	TanH - ReLu - LeakyReLu	–

Table 1 Hyperparameters design space.

III. Test cases

In this section, the capability of the machine–learning model to predict distributed aerodynamics quantities in transonic regime with different levels of geometric complexity and physical phenomena was assessed. Specifically, the model underwent testing on the Benchmark Super Critical Wing (BSCW) and ONERA M6 test cases, followed by evaluation on the wing-body configuration for NASA Common Research Model (CRM) (refer to Table 2).

This approach was undertaken to enhance the complexity of the geometry, incorporate diverse physics, and accommodate varying grid configurations. The progression ranged from a rectangular–wing to a swept–wing and eventually to a wing–body configuration. Additionally, the grid type changed from structured to unstructured. The physical phenomena also varied, encompassing changes in shock wave geometry, coupling between shock wave and separated boundary–layer, and interaction between the fuselage and wing, especially in the junction area, resulting in the generation of complex vortices, pressure changes and boundary–layer effects.

By conducting evaluations on these different test cases, the performance of the ML model was thoroughly examined under varying conditions, allowing for a comprehensive assessment of its predictive capabilities in transonic

aerodynamics.

Test case	Configuration	Mesh Type	Surface Mesh Points
BSCW	Wing	Structured	130,816
ONERA M6	Wing	Structured	149,700
CRM	Wing–Body	Unstructured	78,829

Table 2 Main features of each test case.

Angle of attack and Mach number were chosen as the two independent parameters for the ROMs. They are in the ranges of $[0, 5]$ $[deg]$ and $[0.70, 0.84]$, respectively. This range selection remains consistent across all test cases due to the analogous flow conditions characterizing the geometries. This choice ensures coherence throughout the analysis and guarantees a suitable level of complexity in the physics involved. The transonic regime leads to shock wave formation on the wing of each test case, whereas the high angles of attack to boundary–layer separation.

The necessary number of samples, n, for AoA and M are defined through Latin hypercube sampling (LHS) with a total of 70 points (Figure 3). Sixty percent of the samples (40 flight conditions with circles) are selected for training, 20% (15 flight conditions with squares) are selected for testing, and the remaining 20% (15 with diamonds) are left for validation.

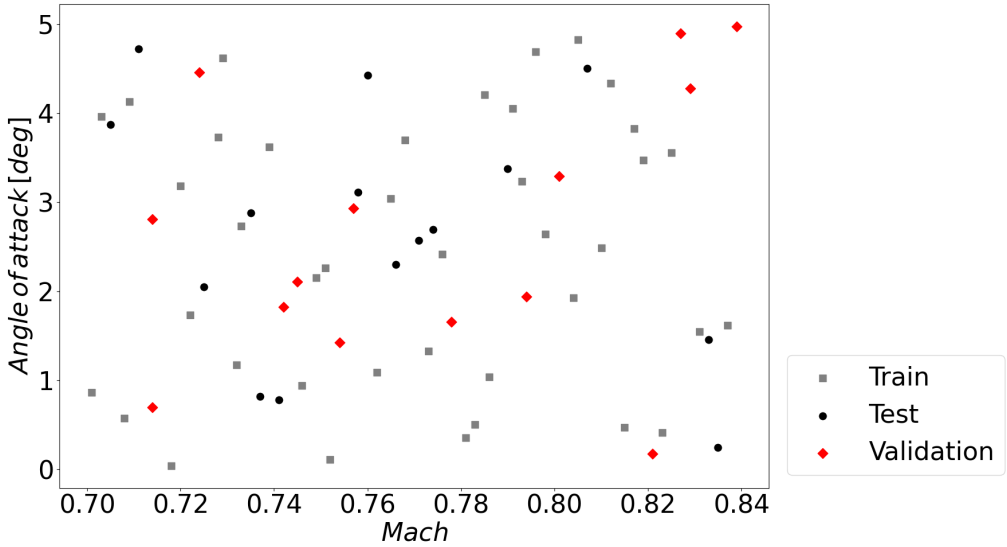


Fig. 3 Training, test and validation samples for Mach number and angle of attack.

A. Benchmark Super Critical Wing (BSCW)

The first test case is the Benchmark Super Critical Wing (BSCW), which is a transonic rigid semi–span wing with a rectangular planform and a supercritical airfoil shape from the AIAA Aeroelastic Prediction Workshop [36]. This wing

is elastically suspended on a flexible mount system with two degrees of freedom, pitch and plunge, and it has been developed for flutter analysis as it is characterized by shock wave motion, shock–induced boundary–layer separation and interaction between shock wave and detached boundary–layer. These three types of nonlinearity are challenging for the ROM predictions.

A mixed-type grid with $15.6 \cdot 10^6$ elements and 130,816 surface elements was generated, structured on the wing surface and in the first layers of the boundary–layer, voxel in the rest of the computational domain. A $y^+ = 1$ is adopted, after a preliminary mesh convergence study that ensured an adequate resolution of the boundary–layer and shock wave. The computational domain extends 100 chords from the solid wall to the farfield. An impression of the grid can be obtained from Figure 4.

The JST central scheme with artificial dissipation is adopted for the discretization of convective flows. The gradients of the flow variables are calculated using a Green Gauss method. The linear solver biconjugate gradient stabilisation is chosen, with ILU preconditioner applied. Aerodynamic coefficient calculations were performed considering a chord length of 0.4064 m and a scaling area of 0.3303 m^2.

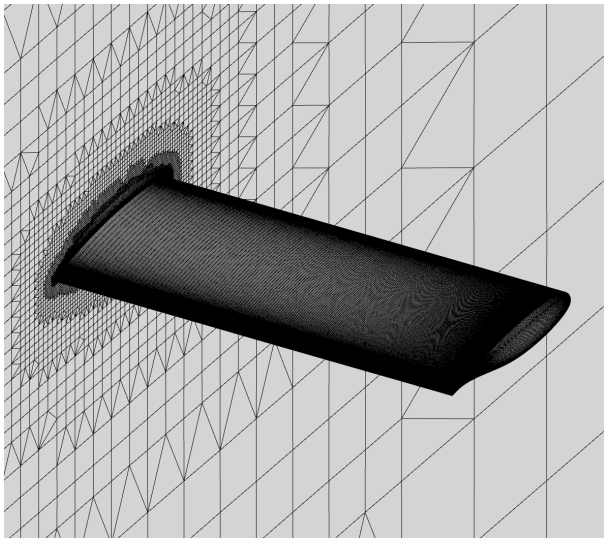


Fig. 4 Impression of the BSCW CFD grid.

Figure 5 reports the convergence study on the number of training samples of the neural network in terms of Mean Absolute Percentage Error (`MAPE`) for BSCW test case. `MAPE` was calculated averaging the absolute error of each prediction in the test set. This prediction error is determined weighted averaging the error on each grid point with the cell area. The power fit figure of values outside of training samples range was added as a dotted line. As expected, increasing the number of training samples reduces `MAPE` on test dataset of pressure coefficient. In particular, double the training samples reduces drastically C_P error of almost one order of magnitude. Therefore, 40 training samples are kept for the rest of the paper.

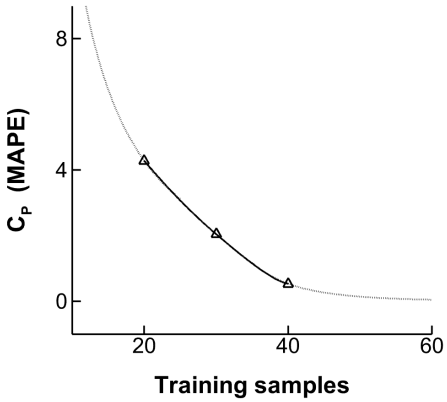
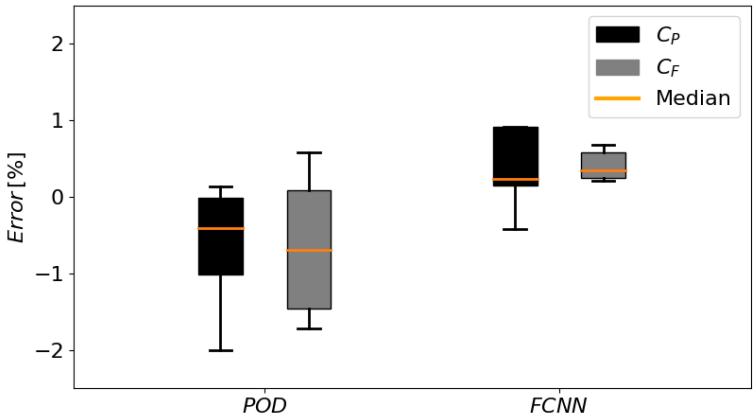


Fig. 5 Training samples convergence study on pressure coefficient of BSCW test dataset. Dotted line denotes the power fit figure of values outside of training samples range.

Table 3 shows `MAPE` on test dataset of pressure coefficient C_P and skin friction coefficient C_F. The error is normalized with respect to the cell area. Neural network behaves better with both quantities, showing its suitability for the complex aerodynamic problem analyzed.

An in-depth comprehension of error distribution is achieved through error percentile analysis. This entails reporting the maximum and minimum error values, highlighting the median with an orange line, and using boxes to represent the 25th and 75th percentiles. Notably, the error band of the FCNN model is narrower, and it exhibits a tendency to overpredict both C_P and C_F. This characteristic is advantageous for certification purposes as the model inclination to overpredict loads results in a conservative approach.



	MAPE [%]	
	FCNN	**POD+I**
C_P	0.5382	0.8958
C_F	0.6080	2.2753

Table 3 Error analysis on test dataset of distributed aerodynamics quantities for BSCW test case.

Figure 6 compares pressure coefficient contour of CFD data with the reconstructed surface field employing two different techniques, POD+I and FCNN. The comparison is conducted on a test sample at $M = 0.801$ and

$AoA = 3.295\ [deg]$, which is one of the most challenging prediction as reported by Righi et al. [37]. A remarkable agreement is found between CFD and FCNN which is able to predict the strong nonlinear pressure distribution in terms of shock wave position and size, in contrast with POD+I which overpredicts the size and misspredicts the position which is more outward. This is due to the POD+I formulation which assumes that the real flow field may be described as a linear superimposition of flow state. Consequently, making predictions in regions characterized by highly nonlinear phenomena, such as shock wave and detached boundary–layer, becomes extremely challenging. Same observations might be pointed out analyzing the skin friction contour in Figure 7. While both methods accurately predict the position of the shock wave, FCNN provides a precise prediction of the extent of the separated boundary–layer, as denoted by the violet contour in Figure 7.

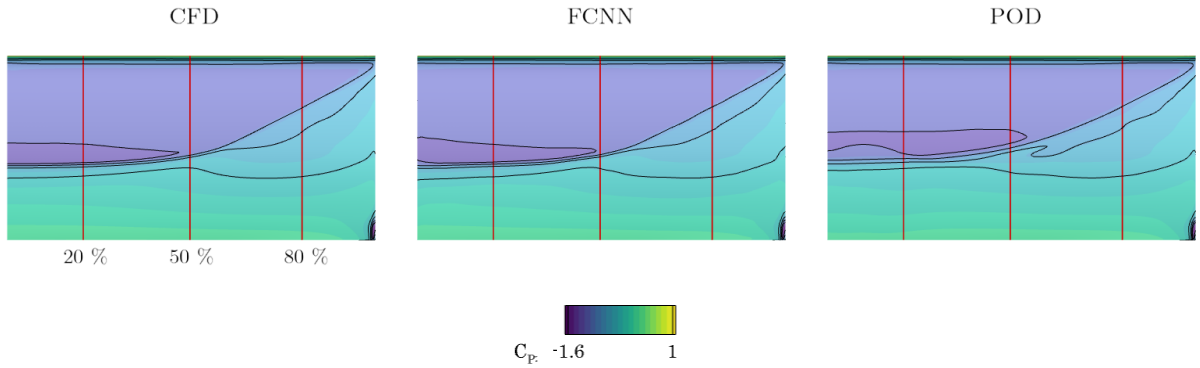


Fig. 6 Prediction of the pressure coefficient contour on the upper surface for BSCW test case at $M = 0.801$ and $AoA = 3.295\ [deg]$. Wing root is located on the left side. Incoming flow impinges onto the wing from the leading edge.

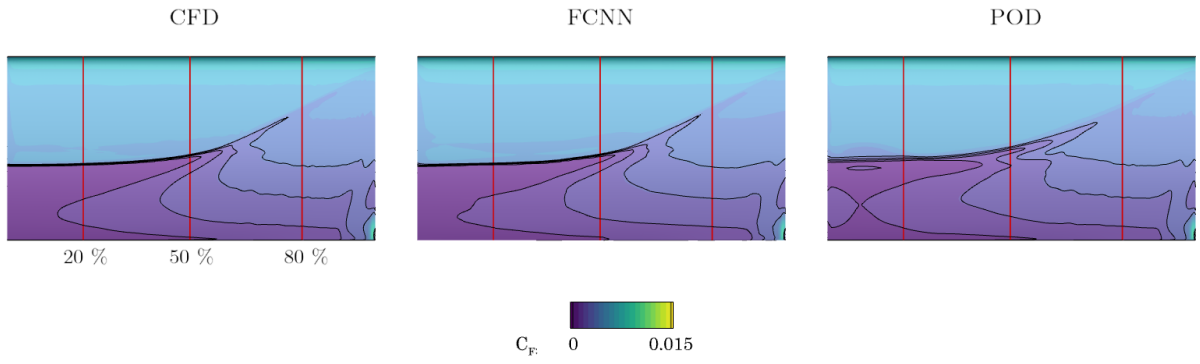


Fig. 7 Prediction of the skin friction magnitude coefficient contour on the upper surface for BSCW test case at $M = 0.801$ and $AoA = 3.295\ [deg]$. Wing root is located on the left side. Incoming flow impinges onto the wing from the leading edge.

Figures 8 and 9 highlight the pressure and skin friction coefficient distribution at three sections along the span of the BSCW wing at $M = 0.801$ and $AoA = 3.295\ [deg]$. The POD+I method captures the global trend of the variables but is unable to resolve any localized features, in contrast to the deep–learning model, which is able to estimate the

distribution with remarkable accuracy. In fact, we can notice a precise prediction of the pressure peaks in proximity of the shock location at 20% of the span which the POD+I approach is not able to capture correctly. This behaviour is likely due to the intrinsic linear formulation of POD+I which does not allow the model to taking into account strong nonlinearities.

The better estimation of the flowfield using the FCNN is clearly shown from the reconstruction of the aerodynamic forces. Table 4 shows lift coefficient C_L, drag coefficient C_D and pitching moment coefficient C_{My} obtained with each approach, with a FCNN error less than 1%. C_{My} has been computed with respect to 30% of the chord, considering the rigid mounting system of the BSCW that induces pitch oscillations around this specific location. Note that aerodynamic forces are calculated by integrating the pressure coefficient distribution and the skin friction coefficient distribution over the wing surface. The contribution of shock wave magnitude predominantly influences the C_D, which both methods predict accurately. As a result, the error in predicting C_D is comparatively lower than that of other coefficients. Conversely, the C_L and C_{My} are sensitive to the shock wave location. In the POD+I method, there is a misplacement of the shock wave position, evident in the considerably higher error in C_L compared to FCNN method. The disparity is even more pronounced in the case of C_{My}, which is closely tied to the shock wave position. Neural network, renowned for its capability in pattern recognition, excels in capturing the shock wave position, thereby achieving a remarkably small error in C_{My} prediction. The relatively larger error in predicting C_L compared to C_{My} could potentially be attributed to minor distributed errors across the surface.

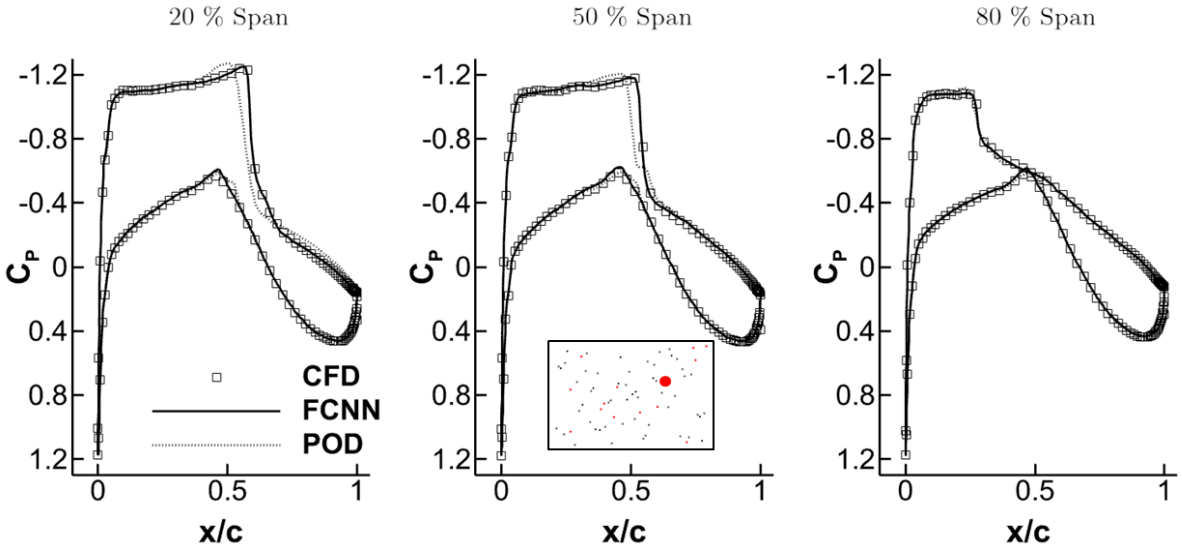


Fig. 8 Pressure coefficient sections of BSCW at $M = 0.801$ and $AoA = 3.295$ $[deg]$.

Figure 10 compares the pressure coefficient distribution obtained with wind tunnel measurements [38] in two sections of the wing at $M = 0.74$ and $AoA = 0.2$ $[deg]$ with the prediction of the presented methods. The flow conditions in this case involve the absence of boundary–layer separation and a small shock wave on the upper surface

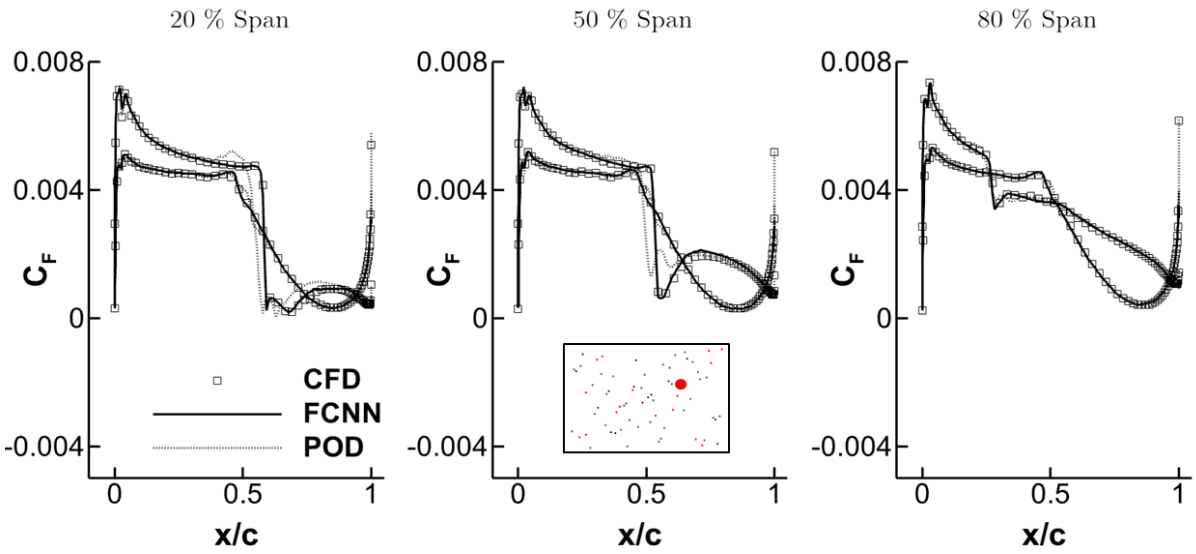


Fig. 9 Skin friction coefficient sections of BSCW at $M = 0.801$ and $AoA = 3.295\ [deg]$.

	CFD	FCNN		POD+I	
		Value	Error (%)	Value	Error (%)
C_L	0.5256	0.5**186**	1.3318	0.5**122**	2.5495
C_D	0.0423	0.042**2**	0.2364	0.04**17**	1.4184
C_{My}	-0.0653	-0.06**49**	0.6125	-0.06**29**	3.6753

Table 4 Aerodynamic coefficients prediction for BSCW test case at $M = 0.801$ and $AoA = 3.295\ [deg]$.

of the wing. Notably, we include experimental data in the figures to demonstrate that under specific conditions, the surrogate models might exhibit superior performance compared to the full–order model. This improvement can be attributed to the insights gained during training from data points spanning a wide flight envelope (cite Fahrat papers). FCNN closes follow CFD in both sections, whereas POD+I misspredicts the pressure distribution at 60% of the span close to the leading edge showing small fluctuations.

B. ONERA M6

The second test case is the ONERA M6 wing, which represents a classic validation test case due to the relatively simple geometry and complicated flow physics. It was developed for studying high Reynolds number flows characterized by separated flow, shock–boundary–layer interaction and complex λ–shaped shock wave.

A mixed-type grid with $18.8 \cdot 10^6$ elements and 149,700 surface elements was generated, structured on the wing surface and in the first layers of the boundary–layer, voxel in the rest of the computational domain. A $y^+ = 1$ is adopted. The computational domain extends 100 chords from the solid wall to the farfield. An impression of the grid can be obtained from Figure 11.

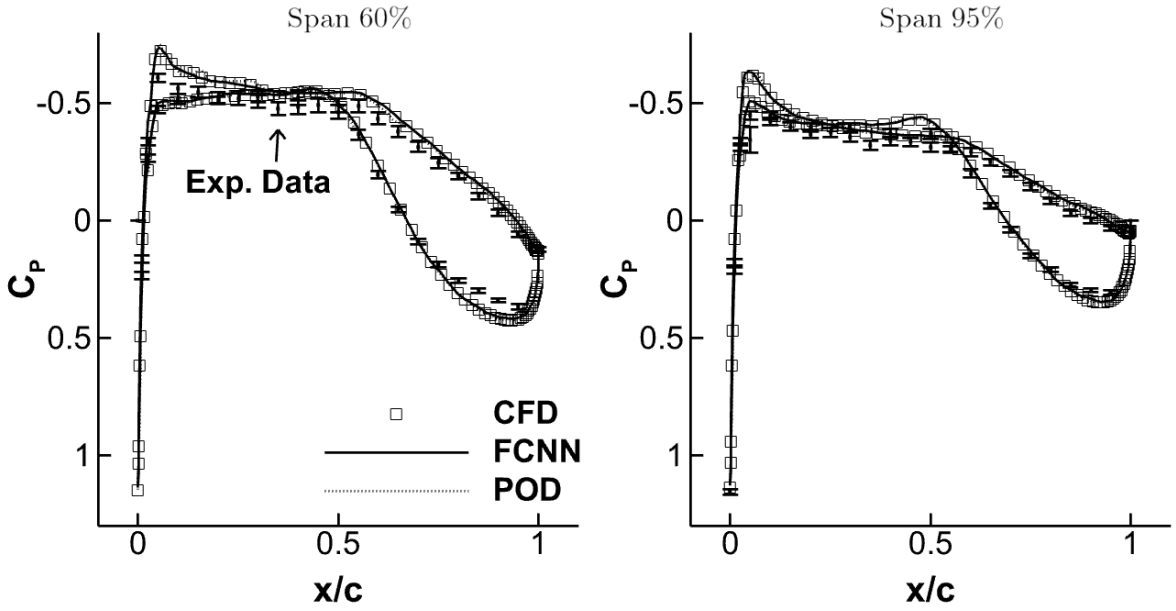


Fig. 10 Comparison of pressure coefficient sections of BSCW at $M = 0.74$ and $AoA = 0.2$ $[deg]$ with experimental data [38].

The JST central scheme with artificial dissipation is adopted for the discretization of convective flows. The gradients of the flow variables are calculated using a Green Gauss method. The linear solver biconjugate gradient stabilisation is chosen, with ILU preconditioner applied. A chord of 0.6461 m and a scaling area of 0.7532 m^2 have been considered for the aerodynamic forces calculation.

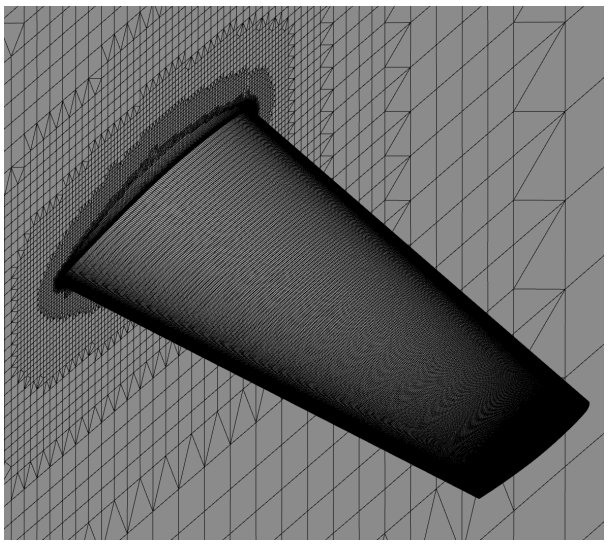
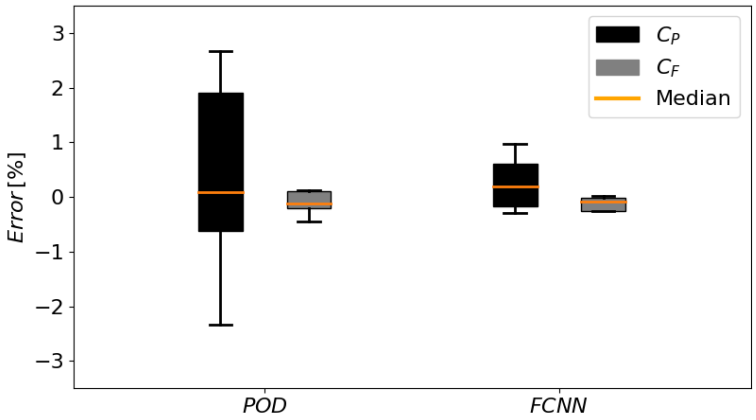


Fig. 11 Impression of the ONERA M6 CFD grid.

Table 3 shows `MAPE` on test dataset of distributed aerodynamics quantities. The error is normalized with respect to the cell area. Similar to the previous test case, FCNN consistently outperforms the POD+I method, exhibiting an

error lower than 0.19% for C_P and 0.53% for C_F compared to the POD+I method. A detailed examination of the error percentile analysis graph reveals a narrower error spread for FCNN, underscoring its higher accuracy on the test dataset in comparison to the POD+I method.



	MAPE [%]	
	FCNN	**POD+I**
C_P	2.4949	2.6836
C_F	0.8470	1.3745

Table 5 Error analysis on test dataset of distributed aerodynamics quantities for ONERA M6 test case.

Figure 12 shows the surface pressure coefficient field on a new prediction at $M = 0.8395$ and $AoA = 3.06\ [deg]$ employing two different techniques, POD+I and FCNN. This particular condition was selected due to the availability of experimental data [39]. It is possible to notice a complex λ–shaped shock wave due to the swept leading edge and a minimum of the airfoil curvature at about 30% chord. The comparison between CFD and ROMs shows an overall good prediction of the shock wave in both methods. In Figure 13, the reconstruction of the skin friction coefficient also demonstrates a good agreement in terms of the shape and position of the λ shock wave. However, FCNN outperforms POD+I in terms of magnitude close to the tip area.

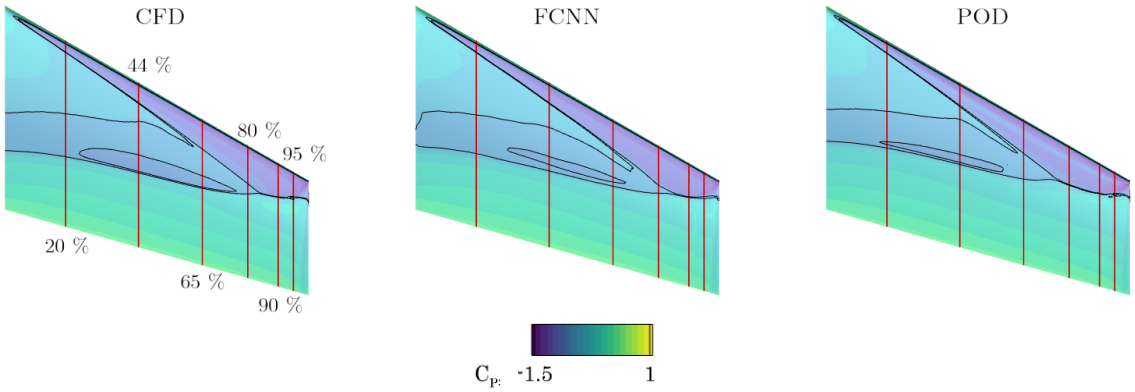


Fig. 12 Prediction of the pressure coefficient contour on the upper surface for ONERA M6 test case at $M = 0.8395$ and $AoA = 3.06\ [deg]$.

The pressure coefficient distribution at several station along the wing span at $M = 0.8395$ and $AoA = 3.06\ [deg]$ is

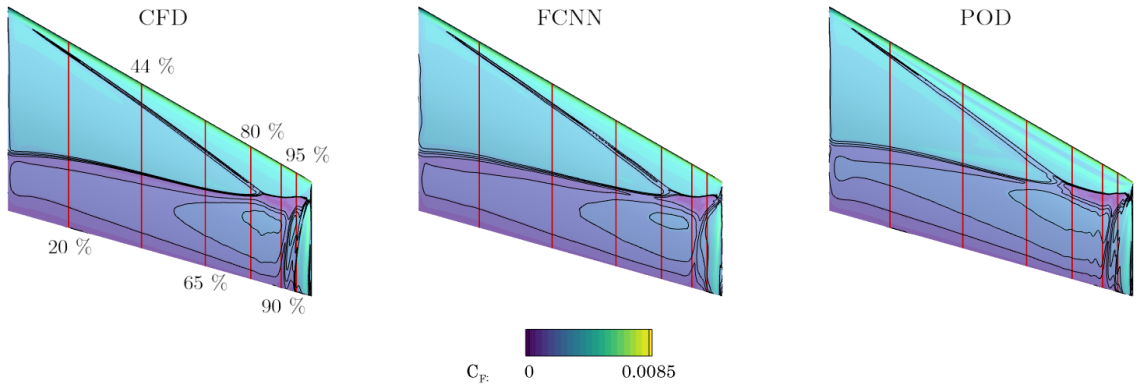


Fig. 13 Prediction of the skin friction magnitude coefficient contour on the upper surface for ONERA M6 test case at $M = 0.8395$ and $AoA = 3.06\ [deg]$.

reported in Figure 14, where a comparison with available experimental data [39] takes place. In contrast with Figure 12, the better behaviour of FCNN is clearly shown. At each span station, the neural network closely follows CFD results, whereas POD+I shows big C_P oscillations in proximity of the shock wave. This non-smooth pressure field results from POD modes with small eigenvalues, which get dramatically magnified throughout the interpolation phase. This is further emphasized by examining the skin friction reconstruction presented in Figure 15. The current results have been based on POD modes that encompass 95% of the total kinetic energy. Reducing the number of modes only resulted in poorer outcomes. Conversely, increasing them for accounting 99% of the total energy does not impact significantly on the results. Therefore, we have made the decision to retain a high number of modes in order to achieve the highest level of accuracy in the predictions.

In fact, the accurate pressure and skin friction estimation of FCNN brings to a better prediction of aerodynamic coefficients, as shown in Table 6. Specifically, the FCNN exhibits excellent prediction capabilities for the lift coefficient, surpassing of almost ten times the performance of POD+I. Additionally, the error of FCNN in the drag and pitching moment coefficients is less than half of what is achieved with POD+I. The reason behind the lower error of the FCNN on the lift coefficient compared to the other two coefficients can be understood by examining Figure 12 and 13. These figures reveal that the contour of the C_F exhibits non–smooth isolines in the CFD solution, which suggests a simulation that may not have converged effectively. However, this issue does not impact the deep–learning algorithm, which is capable of providing accurate solutions even in regions with complex physical phenomena. This highlights the significant advantage of using deep–learning to regularize CFD data.

C. NASA Common Research Model (CRM)

The third test case is the NASA Common Research Model (CRM), which is a transonic wing–body model from the AIAA CFD Drag Prediction Workshop [40]. The main features are a conventional low–wing configuration and a fuselage representative of a wide body commercial aircraft.

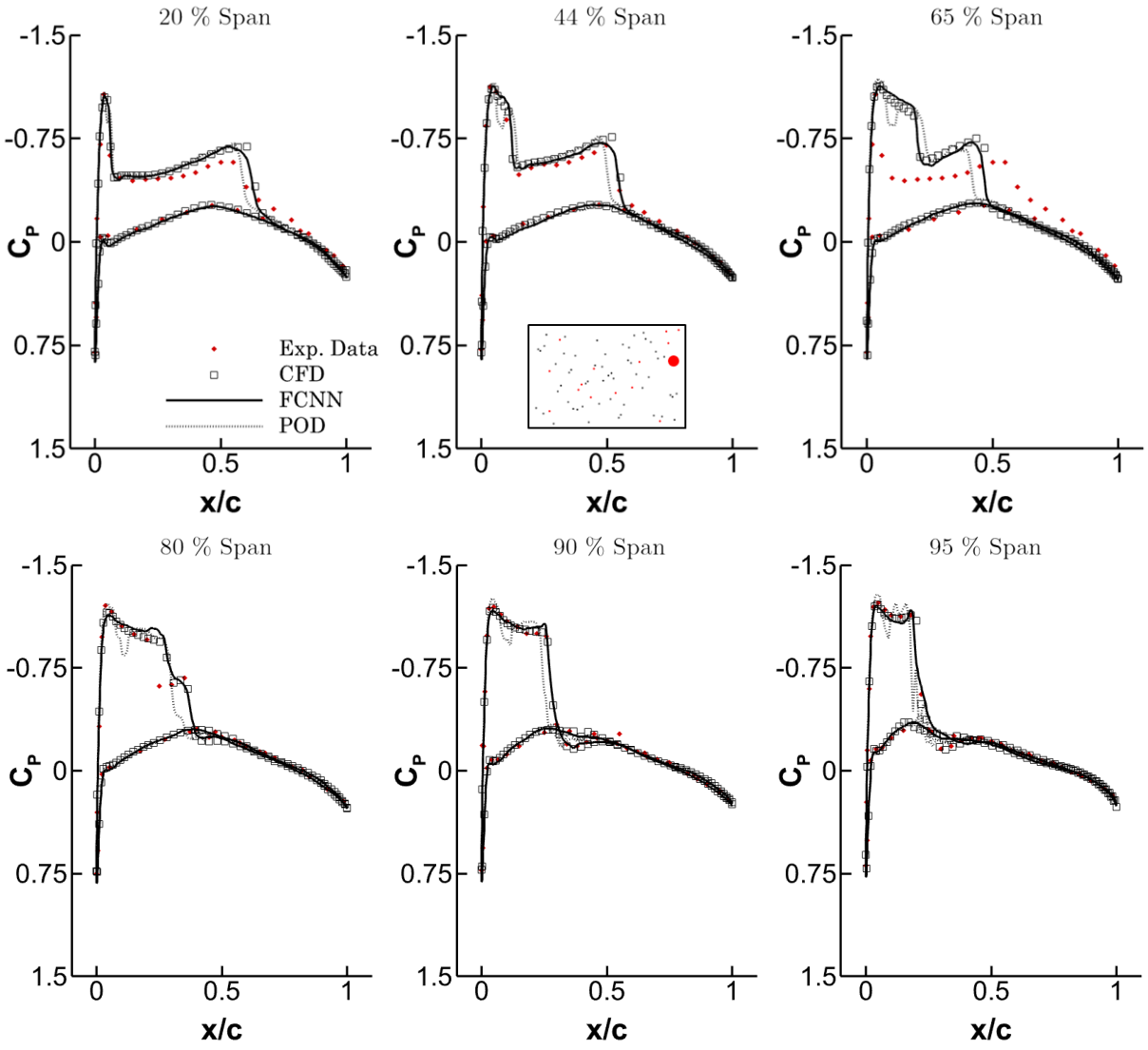


Fig. 14 Comparison of pressure coefficient sections ONERA M6 at $M = 0.8395$ and $AoA = 3.06\ [deg]$ with experimental data [39].

	CFD	FCNN		POD+I	
		Value	Error (%)	Value	Error (%)
C_L	0.2763	0.276**1**	0.0724	0.2**698**	2.3525
C_D	0.0183	0.01**90**	3.8251	0.01**69**	7.6503
C_{My}	-0.1281	-0.12**67**	1.0929	-0.12**29**	4.0593

Table 6 Aerodynamic coefficients prediction for ONERA M6 test case at $M = 0.8395$ and $AoA = 3.06\ [deg]$.

The computational grid was derived from the DLR one created for the AIAA Drag Prediction Workshop [41]. The unstructured grid has $8.8 \cdot 10^6$ elements and 157,374 surface elements. The computational domain extends 100 chords from the fuselage to the farfield. A $y^+ = 1$ is adopted. An impression of the grid can be obtained from Figure 16.

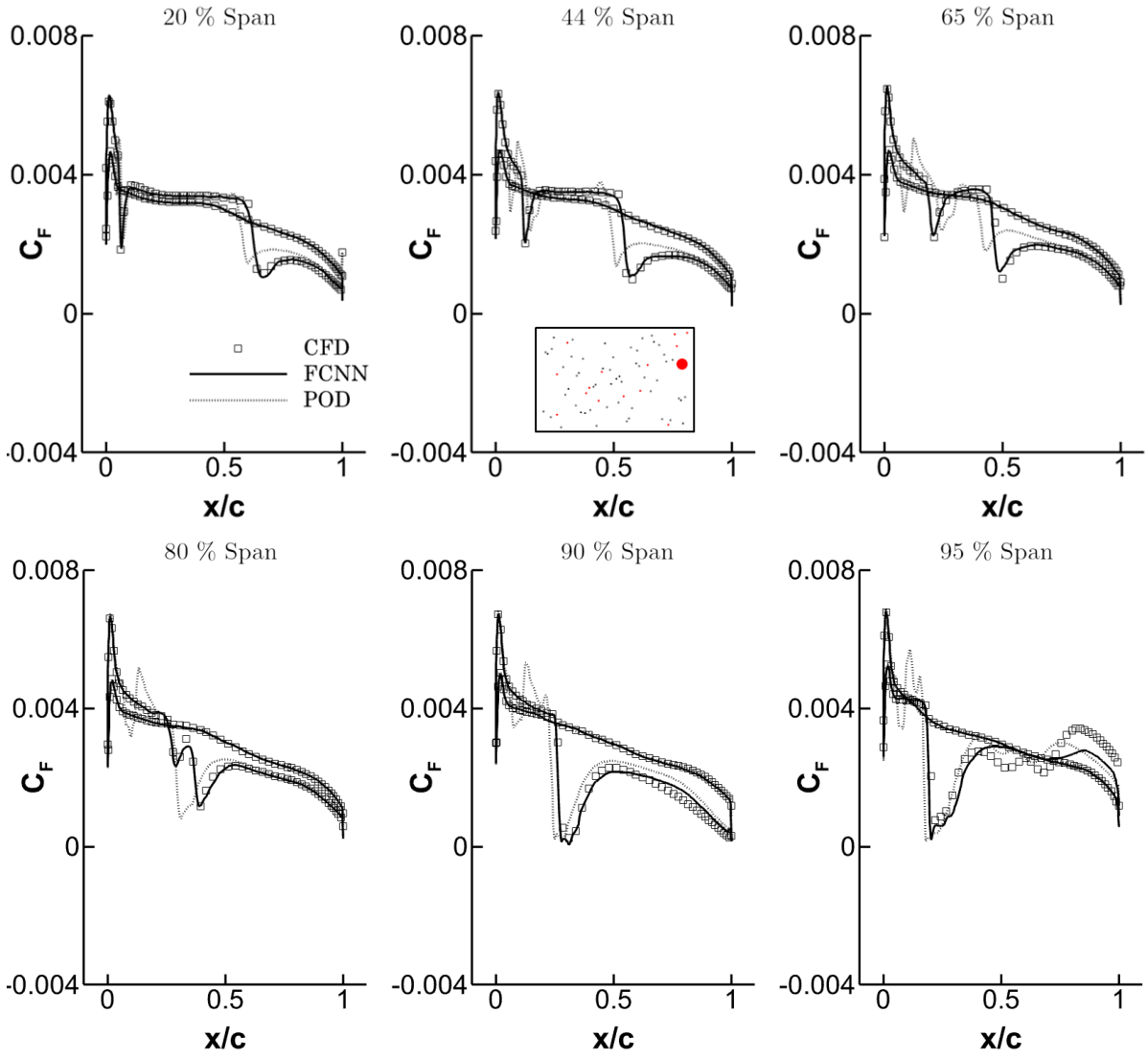


Fig. 15 Skin friction coefficient sections of ONERA M6 at $M = 0.8395$ and $AoA = 3.06$ $[deg]$.

The Lax–Friedrichs central scheme with low artificial dissipation set to 0.02 is adopted for the discretization of convective flows. The gradients of the flow variables are calculated using a Green Gauss method. The linear solver biconjugate gradient stabilisation is chosen, with ILU preconditioner applied. A chord of 0.1412 m and a scaling area of 0.1266 m^2 were considered for aerodynamic coefficients calculation.

This test case represents the most challenging prediction for our deep–learning framework due to the complex geometry, physics and grid configuration. It differs from the others as it has a remarkable amount of surface points in regions with mostly linear flow behavior, such as vast areas of the fuselage, which is advantageous to the linear POD+I technique. Consequently, we chose to quantify the `MAPE` exclusively on the wing, which exhibits high nonlinearity. The results in Table 7 reveal that FCNN outperforms POD+I in both coefficients for the wing. Additionally, when examining the error percentile analysis graph, it becomes evident that FCNN exhibits a lower error spread compared to POD+I.

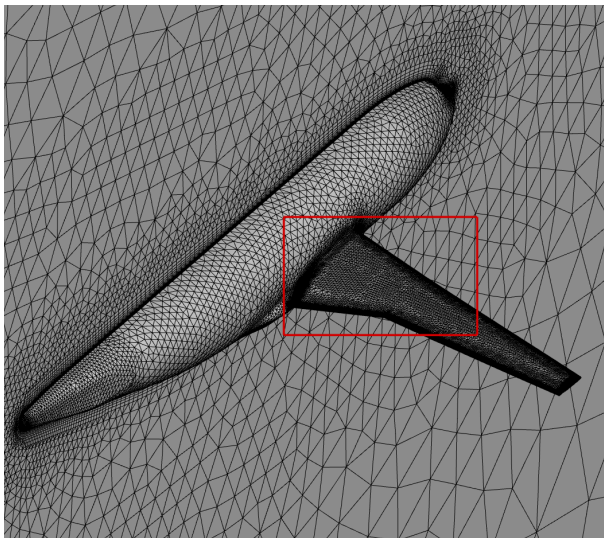
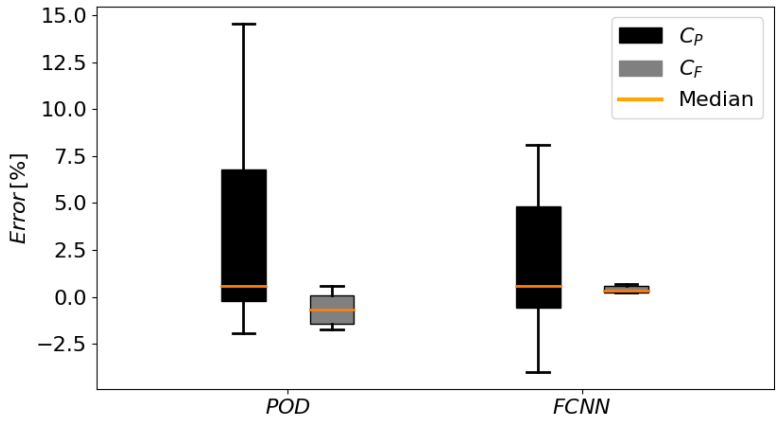


Fig. 16 Impression of the CRM CFD grid. More focus on the red rectangle in Figure 17 and 18.

This underscores the superior performance and accuracy of the developed deep–learning framework in capturing the nonlinearities.



	MAPE [%]	
	FCNN	**POD+I**
C_P	1.7385	3.7136
C_F	1.1942	1.2142

Table 7 Error analysis on test dataset of distributed aerodynamics quantities for CRM test case.

Surface pressure contour predictions are shown in Figure 17 on a test sample at $M = 0.827$ and $AoA = 4.898\ [deg]$. This specific condition has been selected due to the complex physics involved, in terms of boundary–layer detachment and shock wave formation on the upper wing surface. These phenomena interplay with the vortex generated at the wing-fuselage junction, making the prediction particularly challenging. In the absence of local nonlinearities, the surface pressure distribution can be accurately predicted by both ROMs. Therefore, except for the area next to the wing, the prediction error on the fuselage is almost nil.

Examining the predictions for C_P (see Figures 17 and 19), some discrepancies can be observed in the wing–fuselage

junction and the kink area of the wing. However, the rest of the wing and the fuselage demonstrate more accurate predictions, which explains why the error for C_L is lower for FCNN compared to POD+I (refer to Table 8). Conversely, for C_F, the network demonstrates its ability to regularize the surface field, particularly near the trailing edge, as illustrated in Figures 18 and 20. The interaction between the fuselage vortex and the wing leads to significant pressure gradients. This vortex interaction alters the configuration of the shock, as can be noticed when examining cross–sectional slices along the span (Figure 19). At 20% position, no shock is present. However, as we progress toward the wing tip, the shock emerges and intensifies. As shown in Figure 19, the POD+I model is unable to predict the location of the suction peak and shock wave on the wing. As FCNN can handle nonlinearities, it offers considerably better predictions around the shock area. A more complicated network architecture might be used to address discrepancies that exist close to the wing–fuselage junction and in the kink area, which can arise due to irregular and non–uniform data distributions [42].

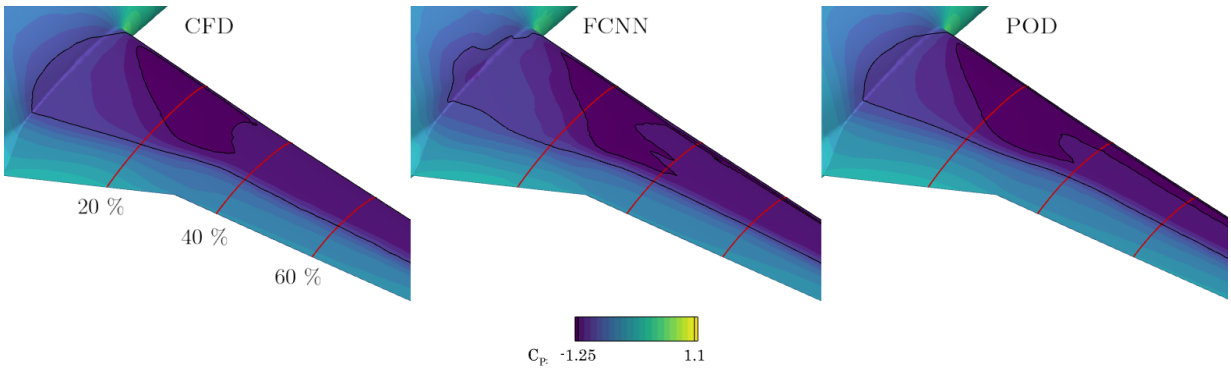


Fig. 17 Prediction of the pressure coefficient contour on the upper surface for CRM test case at $M = 0.827$ and $AoA = 4.898\ [deg]$.

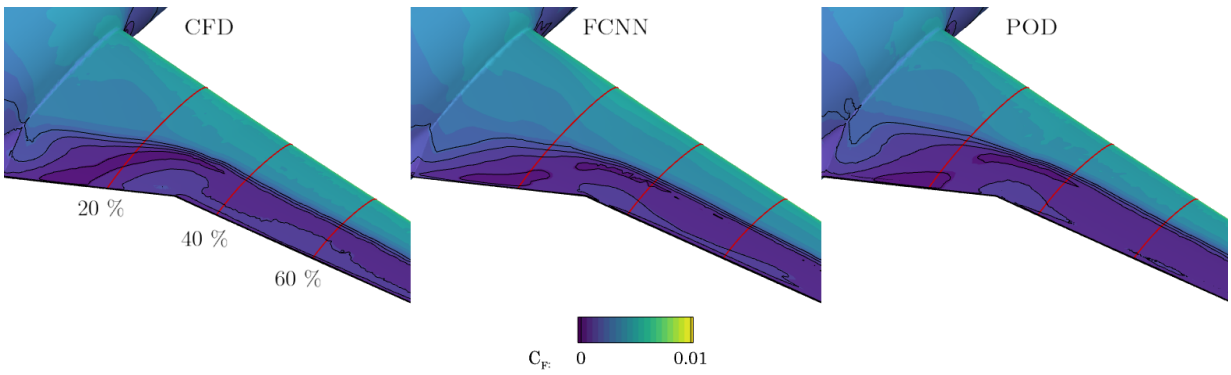


Fig. 18 Prediction of the skin friction magnitude coefficient contour on the upper surface for CRM test case at $M = 0.827$ and $AoA = 4.898\ [deg]$.

Overall, the behavior of the FCNN remains reasonable, especially in the prediction of aerodynamic coefficients, as demonstrated in Table 8. Although the error for C_L is higher compared to POD+I, the neural network proficiency in pattern recognition becomes evident in the accurate prediction of the pitching moment which is closely linked to the

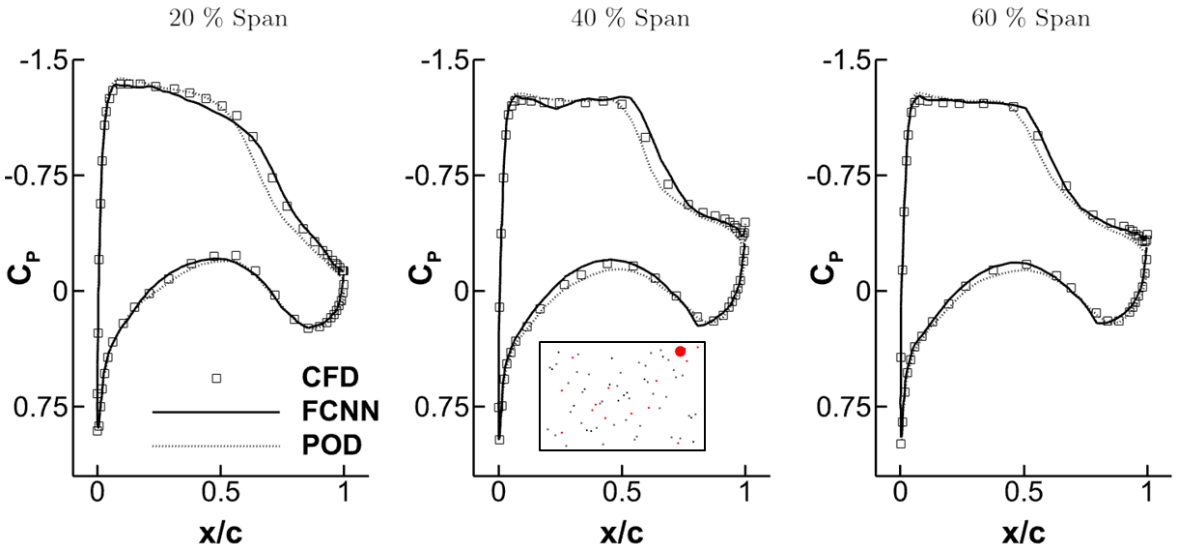


Fig. 19 Pressure coefficient sections of CRM at $M = 0.827$ and $AoA = 4.898$ $[deg]$.

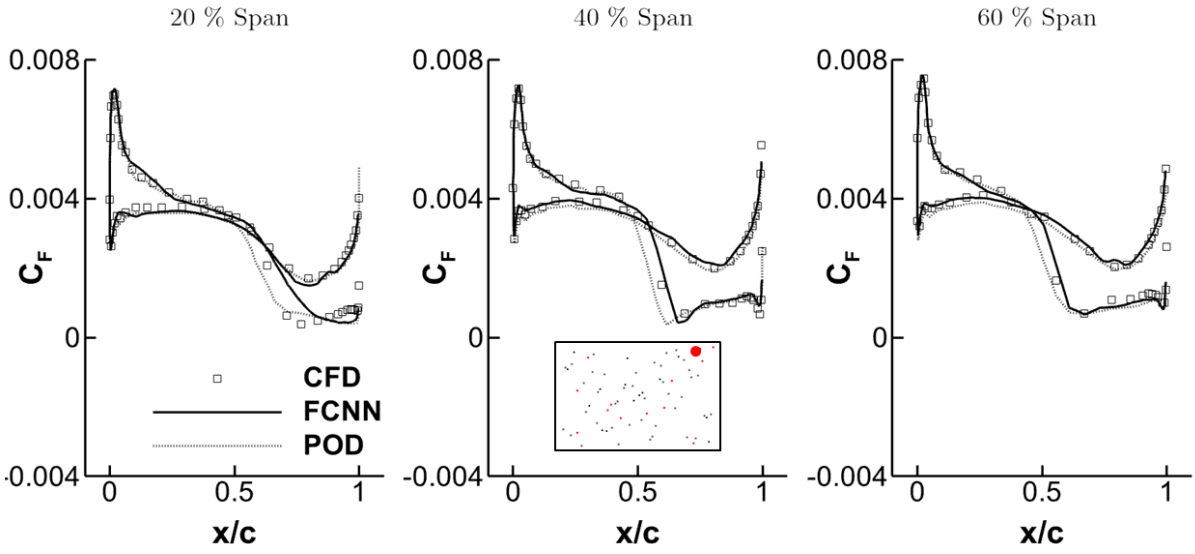


Fig. 20 Skin friction coefficient sections of CRM at $M = 0.827$ and $AoA = 4.898$ $[deg]$.

shock wave placement. This leads to substantial benefits for stability analysis.

	CFD	**FCNN**		**POD+I**	
		Value	Error (%)	Value	Error (%)
C_L	0.4927	0.4**860**	1.3598	0.4**874**	1.0807
C_D	0.0821	0.0**787**	4.1413	0.0**773**	5.8738
C_{My}	-0.0349	-0.03**48**	0.2865	-0.0**238**	31.7103

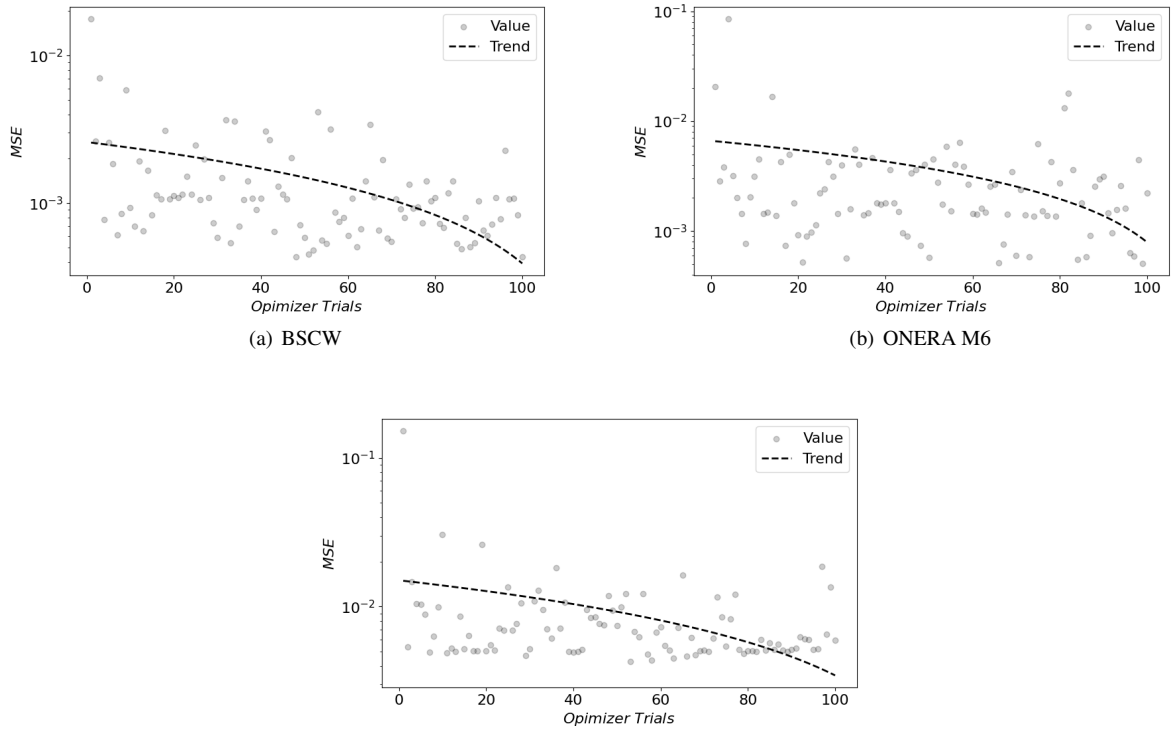
Table 8 Aerodynamic coefficients prediction for CRM at $M = 0.827$ and $AoA = 4.898$ $[deg]$.

It can be concluded that the deep–learning framework excels in predicting localized phenomena, such as strong

shocks or flow separation,while also effectively handling data regularization in complex geometric regions. This effectiveness is credited to its nonlinear formulation, which enhances prediction accuracy. Furthermore, a noteworthy aspect of the FCNN model is its robustness in extrapolating beyond the sampling area, which sets it apart from the POD+I approach.

D. Optimal hyperparameters of the network

The Bayesian optimization history of the FCNN hyperparameters tuning is shown in Figure 21. The plot depicts the evolution of `MSE` across multiple trials for each test case. Transparent points represent the `MSE` value at the end of the training process at each trial. The black dashed line denotes the trend of the error during the optimization. Notably, the graph highlights a consistent reduction in error throughout the optimization process, emphasizing the effectiveness of the tuning procedure in identifying the combination of hyperparameters that yield minimal `MSE` on the validation dataset.



(a) BSCW

(b) ONERA M6

(c) CRM

Fig. 21 Hyperparameter optimization history of the FCNN for each test case.

Table 9 shows the hyperparameters of the neural network after the optimization process, where a relation between the number of surface mesh points assigned in the input layer and the neural network parameters is highlighted.

The optimum learning rate and activation function are similar between BSCW and ONERA M6 test case. This suggests that an optimal region for certain hyperparameters may exist for test cases characterized by similar physical

phenomena. Hence, these hyperparameters have been fixed in order to improve the hyperparameter optimization performance by reducing the design space.

It is important to emphasize that the optimization of the neural network architecture is dependent on the quality of the data. In the analysis of BSCW test case, the algorithm was tested also using few CFD simulations that could not reach a low level of residuals. Among the 70 samples, 4 simulations did not attain a high level of convergence. We conducted training both with and without these simulations, yielding distinct outcomes. This enabled us to observe that, when the poorly converged simulations were included in the training set, the optimizer tends to converge towards a more complex network architecture. This complexity was reflected in an observable increase in the number of parameters, rising from 170,756 to 241,865.

Another noteworthy observation is that, in regions where convergence cannot be easily achieved, the model demonstrates its capability to provide a smooth solution, effectively regularizing the data. Consequently, under specific circumstances, the surrogate model might exhibit superior performance compared to the full–order model (cite Fahrat papers).

Hyperparameter	**Optimal Value**		
	BSCW	ONERA M6	CRM
Learning rate	10^{-4}	10^{-4}	10^{-4}
Number of Hidden Layers	7	6	4
Batch size	6	6	6
Activation function	LeakyReLu	LeakyReLu	LeakyReLu
Number of parameters	170,756	158,180	68,392
Input surface points	130,816	149,700	78,829

Table 9 Optimal hyperparameters for the three test cases.

E. Computing Cost Saving

A comprehensive computational cost analysis was conducted to assess the efficiency of reduced-order methods compared to the high-order approach, as presented in Table 10. In CFD simulations, it was found that an average of 900 CPU hours is required for a single run, while generating the entire dataset demands approximately 65,000 CPU hours. The prediction time for a single sample is comparable between the two ROMs. However, the FCNN, which delivers highly accurate predictions, necessitates computationally demanding optimization and training phases. As a result, while the POD+I method may suffice during the initial design stages, when precise and reliable results are paramount, employing neural network becomes an extremely valuable approach despite its greater computational requirements.

One of the primary obstacles we faced throughout this study was the considerable computational cost associated with each high-fidelity simulation. Consequently, it is crucial to adopt a philosophy aimed at minimizing the amount of training data necessary to develop an accurate model. Neglecting the costs associated with data generation can pose a

potential risk, as it may compromise the computational benefits gained from employing a reduced–order model. Hence, striking a balance between model accuracy and computational efficiency is essential for successful implementation.

Test case	CFD (CPU hours)		FCNN (GPU hours)			POD+I (CPU hours)	
	Simulation		Optimization	Training	Prediction	Training	Prediction
	(70 runs)	(1 run)	(100 trials)	(1 model)	(1 sample)	(1 model)	(1 sample)
BSCW	42,000	600	10	0.6	0.0001	0.002	0.0002
ONERA M6	126,000	1,800	11	1.6	0.0001	0.002	0.0002
NASA CRM	28,000	400	7	0.4	0.0001	0.002	0.0002

Table 10 Computing cost comparison between ROMs and CFD for each test case.

IV. Applications of the Framework

To evaluate the practical suitability of the proposed method, two applications are introduced. Both concern assessments which require several repeated evaluations of the aerodynamic forces/moments, which are carried out with the deep–learning framework. Specifically, we focus on the BSCW test case. The first application is the assessment of the static aeroelastic displacement in a flow regime in which aerodynamic forces depend nonlinearly on the displacements. The second application concerns the propagation of aleatoric uncertainties of Mach and angle of attack onto the aerodynamic forces. In the second case, we utilized neural network to generate high number of samples in a brief timeframe and employed Polynomial Chaos Expansion (PCE) to examine statistical quantities. Here, we demonstrated the efficiency of the method and, importantly, we verified that the obtained results align with physical principles. By assessing performance and robustness of the framework in these applications, we gain valuable insights into the method potential and ability to tackle complex challenges encountered in practical settings. In fact, the applications show that the deep–learning framework proposed is both effective (it provides accurate data), efficient (at low cost) and robust (it lowers the dependence on data quality in every sample).

A. Static Deformation of BSCW

The presented deep–learning framework has been coupled with the structural model of the pitch system of the BSCW test case to perform aeroelastic analysis. The calculation of the twist angle θ is derived from the equation $K \cdot \theta = M$, wherein K represents the stiffness matrix and M is the pitching moment. Due to the interdependence of the pitching moment and the twist angle, the nonlinear characteristics of the problem necessitate an iterative approach for resolution. Table 11 provides the structural properties of the BSCW wing test case [38, 43, 44]. In this analysis, the structural damping is neglected, and both the elastic axis and the center of gravity are located at the mid–chord position. A schematic of the half-span BSCW is shown in Figure 22.

The influence of increasing dynamic pressure on aeroelastic behavior is noteworthy as it enhances the impact of

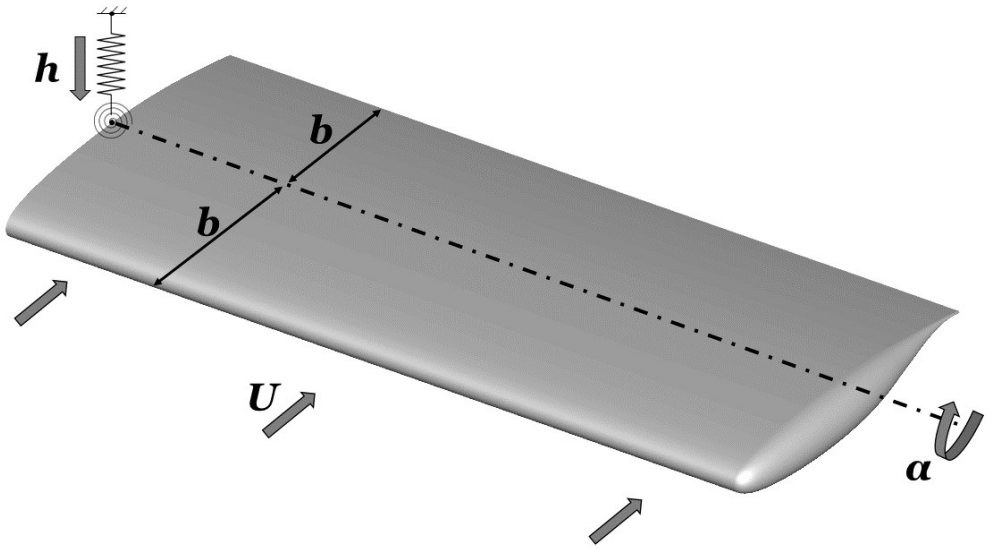


Fig. 22 Schematic of the half-span BSCW subject to pitch and plunge motion.

BSCW properties	
K_α	4018.64 Nm rad^{-1}
K_h	38484.12 N m^{-1}
c	0.4064 m

Table 11 Structural properties of the BSCW model.

pressure in relation to structural stiffness. Figure 23 illustrates this effect by presenting the incremental elastic twist angle θ $[deg]$ predicted by the deep–learning framework as a function of dynamic pressure for rigid angles of attack ranging from 0 to 5 $[deg]$ at $M = 0.74$. The circular symbols in the figure represent CFD results obtained from Heeg and Chwalowski [45].

In static aeroelastic solutions with rigid angles of attack equal to or less than 1.15 $[deg]$, the dynamic pressure induces a negative twist, resulting in a reduction of the angle of attack (nose down). However, for angles of attack greater than 1.15 $[deg]$, the induced elastic twist angle becomes positive (nose up) and increases with dynamic pressure. This phenomenon can be attributed to a shift in the center of pressure, resulting in a change of sign for the pitching moment. This, in turn, leads to an increase in the aerodynamic load as the angle of attack rises. A remarkable agreement might be notice between the presented results and computational ones from Heeg and Chwalowski [45].

To conclude, we want to point out that a static aeroelastic solution has been computed within the presented framework in 0.002 GPU hours. The problem is nonlinear and necessitates iterative resolution. This stands in stark contrast to the method proposed by Heeg and Chwalowski [45], which consumed approximately 1000 CPU hours with a gradient method. Unlike traditional method that would necessitate running a CFD simulation at each step, this approach demonstrates efficiency and effectiveness in handling aeroelastic simulations.

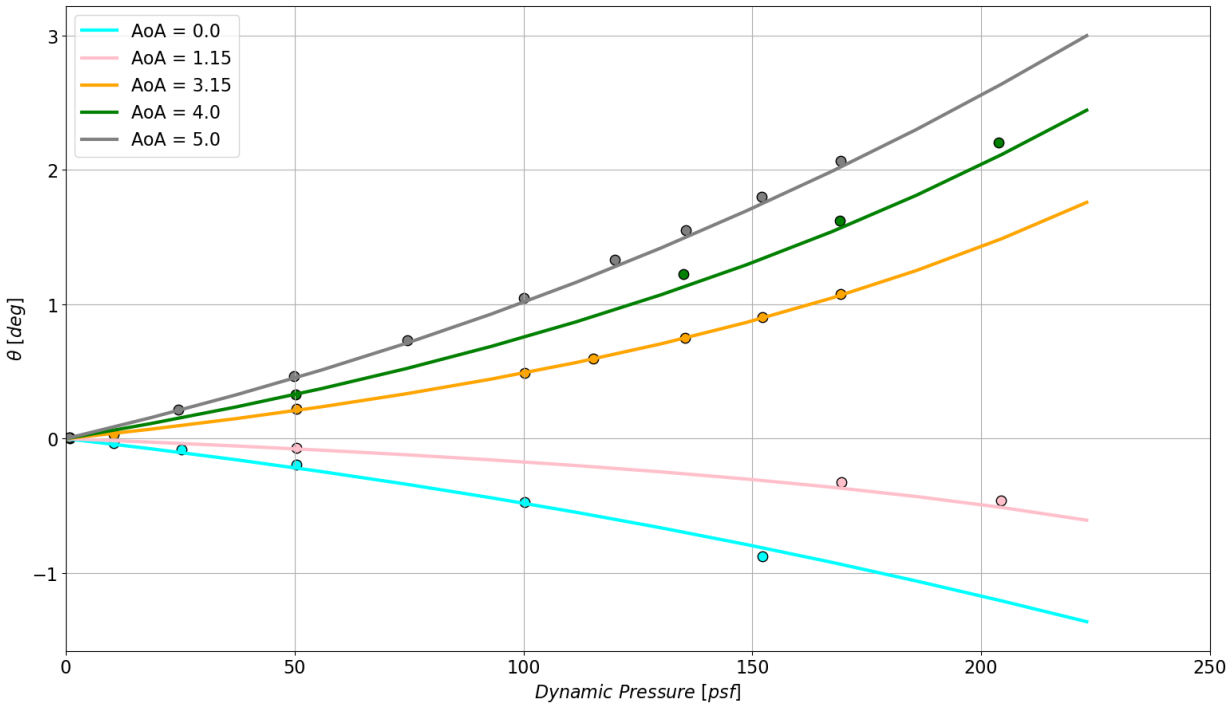


Fig. 23 Static elastic twist angles of BSCW at $M = 0.74$ for different initial AoAs.

B. Propagation of Uncertainties of BSCW

The aim of this study is to effectively address the uncertainties associated with flow parameters. This research is motivated by the significant variability observed in the results of the Second Aeroelastic Prediction Workshop, particularly the substantial scatter in the test case results [46].

This study leverages the developed deep–learning framework to efficiently generate an high number of samples, and Non-Intrusive Polynomial Chaos Expansion (NIPCE) technique [47–50] to approximate the probability distribution associated with the random variables. This technique has been successfully applied to similar problems, as documented in the literature [51–53]. In the expansion, the variable of interest Y, which depends on a vector of random variables ξ, is expressed as a series expansion:

$$Y(\xi) = \sum_{i=o}^{p} \alpha_i \Psi(\xi) \tag{4}$$

Here, Ψ represents multivariate basis functions, often Hermite polynomials for normally distributed random variables [50, 54], and α_i are the spectral coefficients. The number of terms in the expansion, denoted by p, is determined by the number of random variables n and the chosen order of the NIPCE s, and can be calculated using $p = \frac{(n+s)!}{n!s!}$.

To determine the spectral coefficients α_i, a minimum of m experiments are required, assuming oversampling with $m > n$. The m values of Y are sampled using the Latin Hypercube Sampling (LHS) method. In this study, the random variables Mach number and angle of attack have been selected for analysis (refer to Table 12). It is assumed that the

random variables follow a normal distribution with given standard deviation, 0.01 for M and 0.1 for AoA. These values are chosen to align with the order of magnitude of the uncertainty typically encountered during standard wind tunnel campaigns [55–57]. The NIPCE is constructed using Hermite functions.

RV	Distribution	Std deviation
$Mach$	Normal	0.01
AoA	Normal	0.1

Table 12 Freestream Random Variables, their distribution and standard deviation values used to quantify uncertainty.

Figure 24 displays the Probability Density Function (PDF) of the lift coefficient at $M = 0.80$ and $AoA = 3.0\ [deg]$, which was identified as a condition with significant variability by Righi et al. [37]. It is important to note that the PDF retains its shape and characteristics as the NIPCE order increases. Moreover, as the NIPCE order is augmented, the PDF reveals pronounced nonlinearities that become more prominent.

A comparison of the PDFs at various Mach numbers and AoAs is showcased in Figure 25. In the low M-AoA region, the PDF exhibits a linear behavior, as evidenced by their symmetric distribution. However, as the Mach number and AoA increase, the PDF displays pronounced nonlinearity, characterized by the observed asymmetry in the curves. Notably, at $M = 0.84$ and $AoA = 5\ [deg]$, this asymmetry is less pronounced. This phenomenon can be attributed to the complete separation of the boundary–layer and the relatively less intense interaction with the shock wave, lending support to an explanation grounded in linear theory.

Subsequently, we conduct a sensitivity analysis utilizing Sobol indices to precisely measure the contribution of individual input variables and their interactions to the overall variability in model output. These indices assist in discerning the relative importance of each factor. The Sobol indices, depicted in Figure 26, reveal the impact of Mach variations. It is evident that as the Mach number increases, the influence of Mach, treated as a random variable, becomes more prominent, primarily due to the intensified compressibility effects. This dominance of Mach is particularly significant at high Mach numbers, where the flow experiences separation, diminishing the influence of the AoA in comparison. These findings align well with previously published results by Heeg et al. [46], lending credibility to our observations.

This study demonstrated that our model could generate the PDF at each condition in only 0.01 GPU hours, highlighting the efficiency and resource–saving benefits of our approach within this context. Crucially, the obtained results consistently align with physical principles, underscoring the reliability of the approach. Moreover, it is pertinent to acknowledge that, although our present focus did not encompass the incorporation of uncertainty into the neural network, this remains a potential avenue for future exploration and refinement of the proposed methodology.

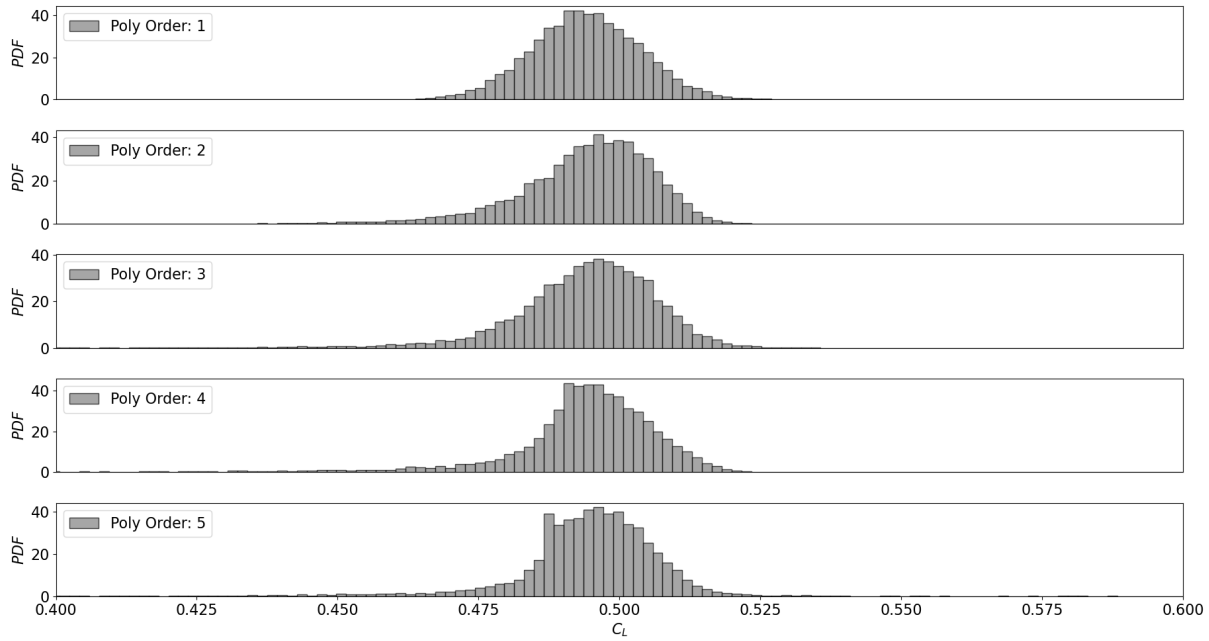


Fig. 24 Probability Density Function of lift coefficient at $M = 0.80$ and $AoA = 3.0\ [deg]$, polynomials are consistent up to $p = 5$, 10,000 samples, 2 Random Variables.

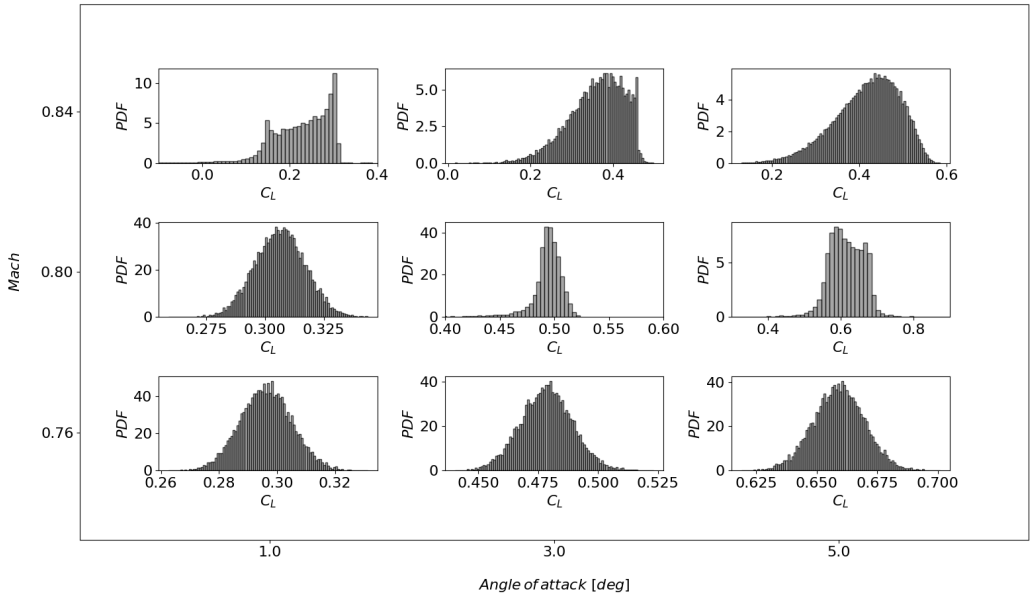


Fig. 25 Probability Density Function of lift coefficient, $p = 4$, 10,000 samples, 2 Random Variables.

V. Conclusions

A deep–learning framework was developed to capture nonlinear phenomena occurring in transonic regime. It was shown that our method is able to predict distributed quantities which are critical for the determination of aerodynamic

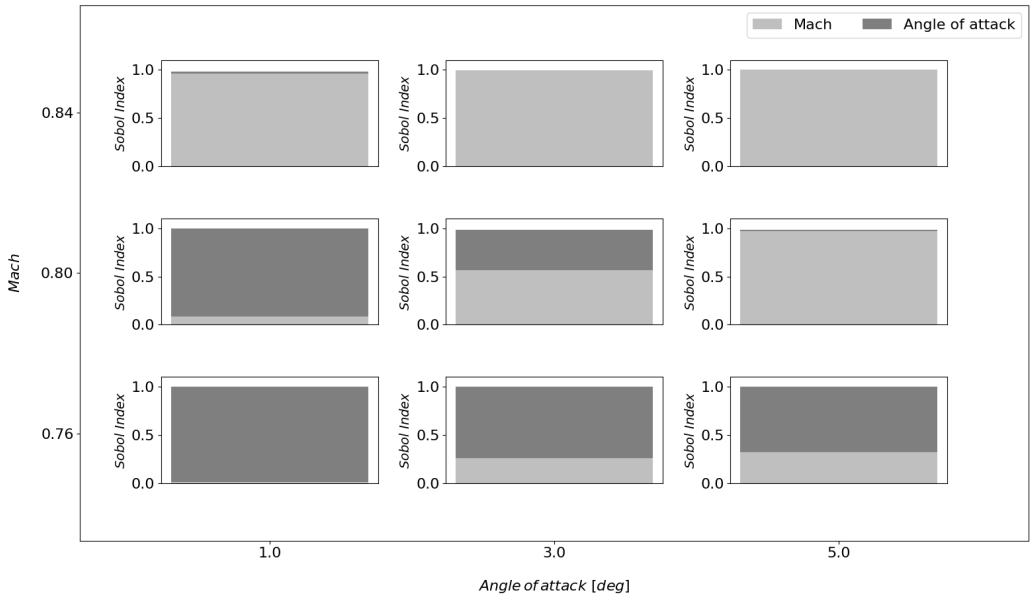


Fig. 26 Sobol indices of the two random variables for lift coefficient.

loads and aeroelastic analysis.

The analyzed scenarios involved inherent difficulties because of the three–dimensional nature of the flow and the existence of nonlinear phenomena, such as shock waves and boundary–layer separation. Despite these challenges, promising results were obtained for the wing–only test cases, showing the effectiveness of the model implemented. On the other hand, the wing–body test case highlighted some limitation of the method implemented. In fact, it can be concluded that structured grid, or rather ordered points, impacts in a positive way on the capability prediction of the deep–learning framework developed.

We have also shown that the proposed approach provides a robust and reliable means to compute the static twist deformation of the BSCW, while reducing computational complexity and time demands of traditional aeroelastic simulations. It not only enables a deeper understanding of the wing structural behavior but also contributes to optimizing wing design for enhanced performance.

We employed Non-Intrusive Polynomial Chaos Expansion (NIPCE) to quantify the impact of uncertainties from freestream variables in BSCW test case. The resulting confidence intervals were found to be significant, providing valuable insights that may help explain the considerable variability observed in past Aeroelastic Prediction Workshops. Notably, the PDFs obtained exhibited varying deviations from the Gaussian distribution, thereby highlighting the nonlinear nature of the underlying physics.

The impact of the present work on the aerospace community may be summarized in three points. First, the

optimization of architecture of the neural network is an essential idea to speed up the process of setting up the hyperparameters and to enhance the model accuracy. Second, regularization of the data is a considerable advantage of the presented algorithm. In fact, in regions where achieving convergence is challenging, the model showcases its ability to deliver a smooth solution, effectively regularizing the data. Finally, the adaptability of the proposed approach extends to other significant areas of research, including unsteady aerodynamics and shape optimization problems. This adaptability stems from the optimization of the neural network architecture, which enables the identification of the optimal set of hyperparameters for each specific test case.

Taking the proposed steady–state background as a starting point, the deep–learning framework will be applied to unsteady responses based on angle of attack step input for pattern recognition.

Acknowledgement

The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. The project has been supported by Digitalization Initiative of the Zurich Higher Education Institutions (DIZH) grant from Zurich University of Applied Sciences (ZHAW).

References

[1] Lassila, T., Manzoni, A., Quarteroni, A., and Rozza, G., "Model order reduction in fluid dynamics: challenges and perspectives," *Reduced Order Methods for modeling and computational reduction*, 2014, pp. 235–273.

[2] Lucia, D. J., Beran, P. S., and Silva, W. A., "Reduced-order modeling: new approaches for computational physics," *Progress in aerospace sciences*, Vol. 40, No. 1-2, 2004, pp. 51–117.

[3] Zimmermann, R., and Görtz, S., "Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model," *The Aeronautical Journal*, Vol. 116, No. 1184, 2012, pp. 1079–1100.

[4] Zimmermann, R., Vendl, A., and Görtz, S., "Reduced-order modeling of steady flows subject to aerodynamic constraints," *AIAA journal*, Vol. 52, No. 2, 2014, pp. 255–266.

[5] Franz, T., Zimmermann, R., Görtz, S., and Karcher, N., "Interpolation-based reduced-order modelling for steady transonic flows via manifold learning," *International Journal of Computational Fluid Dynamics*, Vol. 28, No. 3-4, 2014, pp. 106–121.

[6] Franz, T., "Reduced-order modeling for steady transonic flows via manifold learning," Ph.D. thesis, Deutsches Zentrum für Luft-und Raumfahrt, 2016.

[7] Ripepi, M., Verveld, M. J., Karcher, N., Franz, T., Abu-Zurayk, M., Görtz, S., and Kier, T., "Reduced-order models for aerodynamic applications, loads and MDO," *CEAS Aeronautical Journal*, Vol. 9, No. 1, 2018, pp. 171–193.

[8] Brenner, M., Eldredge, J., and Freund, J., "Perspective on machine learning for advancing fluid mechanics," *Physical Review Fluids*, Vol. 4, No. 10, 2019, p. 100501.

[9] Brunton, S. L., Noack, B. R., and Koumoutsakos, P., "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, Vol. 52, 2020, pp. 477–508.

[10] Fukami, K., Fukagata, K., and Taira, K., "Assessment of supervised machine learning methods for fluid flows," *Theoretical and Computational Fluid Dynamics*, Vol. 34, No. 4, 2020, pp. 497–519.

[11] Kutz, J. N., "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, Vol. 814, 2017, pp. 1–4.

[12] Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T., and Yeh, C.-A., "Modal analysis of fluid flows: Applications and outlook," *AIAA journal*, Vol. 58, No. 3, 2020, pp. 998–1022.

[13] Torregrosa, A., García-Cuevas, L., Quintero, P., and Cremades, A., "On the application of artificial neural network for the development of a nonlinear aeroelastic model," *Aerospace Science and Technology*, Vol. 115, 2021, p. 106845.

[14] Sun, D., Wang, Z., Qu, F., and Bai, J., "A deep learning based prediction approach for the supercritical airfoil at transonic speeds," *Physics of Fluids*, Vol. 33, No. 8, 2021, p. 086109.

[15] Wang, X., Kou, J., and Zhang, W., "A new dynamic stall prediction framework based on symbiosis of experimental and simulation data," *Physics of Fluids*, Vol. 33, No. 12, 2021, p. 127119.

[16] Zhang, X., Xie, F., Ji, T., Zhu, Z., and Zheng, Y., "Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 373, 2021, p. 113485.

[17] Sabater, C., Stürmer, P., and Bekemeyer, P., "Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques," *AIAA Journal*, 2022, pp. 1–13.

[18] Zhiwei, S., Chen, W., Zheng, Y., Junqiang, B., Zheng, L., Qiang, X., and Qiujun, F., "Non-intrusive reduced-order model for predicting transonic flow with varying geometries," *Chinese Journal of Aeronautics*, Vol. 33, No. 2, 2020, pp. 508–519.

[19] Du, X., He, P., and Martins, J. R., "Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling," *Aerospace Science and Technology*, Vol. 113, 2021, p. 106701.

[20] Espinosa Barcenas, O. U., Quijada Pioquinto, J. G., Kurkina, E., and Lukyanov, O., "Surrogate Aerodynamic Wing Modeling Based on a Multilayer Perceptron," *Aerospace*, Vol. 10, No. 2, 2023, p. 149.

[21] Andrés-Pérez, E., and Paulete-Periáñez, C., "On the application of surrogate regression models for aerodynamic coefficient prediction," *Complex & Intelligent Systems*, Vol. 7, 2021, pp. 1991–2021.

[22] Castellanos, R., Varela, J. B., Gorgues, A., and Andrés, E., "An assessment of reduced-order and machine learning models for steady transonic flow prediction on wings," *ICAS 2022*, 2022.

[23] Zhou, H., Xie, F., Ji, T., Zhang, X., Zheng, C., and Zheng, Y., "Fast transonic flow prediction enables efficient aerodynamic design," *Physics of Fluids*, Vol. 35, No. 2, 2023, p. 026109.

[24] Massegur, D., Clifford, D., Da Ronch, A., Lombardi, R., and Panzeri, M., "Low-Dimensional Models for Aerofoil Icing Predictions," *Aerospace*, Vol. 10, No. 5, 2023, p. 444.

[25] Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E., "Physics-informed neural networks (PINNs) for fluid mechanics: a review," *Acta Mechanica Sinica*, Vol. 37, No. 3, 2021, p. 1727–1738.

[26] Rohlfs, L., and Weiss, J., "Assimilating Velocity Fields from BOS Measurements in Supersonic Flows Using Physics Informed Neural Networks," *AIAA AVIATION 2023 Forum*, 2023, p. 4363.

[27] Sureshbabu, S., Tejero, F., Sanchez-Moreno, F., MacManus, D. G., and Sheaf, C., "Deep-Learning Methods for Non-Linear Transonic Flow-Field Prediction," *AIAA AVIATION 2023 Forum*, 2023, p. 3719.

[28] "SU2 Code," , 2022. URL `https://su2code.github.io/`.

[29] Malouin, B., Trépanier, J.-Y., Gariépy, M., et al., "Interpolation of transonic flows using a proper orthogonal decomposition method," *International Journal of Aerospace Engineering*, Vol. 2013, 2013.

[30] Tan, B.-T., Willcox, K. E., and Damodaran, M., "Applications of proper orthogonal decomposition for inviscid transonic aerodynamics," 2003.

[31] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al., "KerasTuner," `https://github.com/keras-team/keras-tuner`, 2019.

[33] Snoek, J., Larochelle, H., and Adams, R. P., "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems*, Vol. 25, 2012.

[34] Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M., "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[35] Auer, P., "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, Vol. 3, No. Nov, 2002, pp. 397–422.

[36] Heeg, J., "Overview of the aeroelastic prediction workshop," *51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2013, p. 783.

[37] Righi, M., Düzel, S., Anderegg, D., Da Ronch, A., Massegur Sampietro, D., and Immordino, G., "ROM-Based Uncertainties Quantification of Flutter Speed Prediction of the BSCW Wing," *AIAA SCITECH 2022 Forum*, 2022, p. 0179.

[38] Raveh, D. E., Yossef, Y. M., and Levy, Y., "Analyses for the second aeroelastic prediction workshop using the EZNSS code," *AIAA Journal*, Vol. 56, American Institute of Aeronautics and Astronautics Inc., 2018, pp. 387–402. https://doi.org/10.2514/1.J055960.

[39] Schmitt, V., "Pressure distributions on the ONERA M6-wing at transonic mach numbers, experimental data base for computer program assessment," *AGARD AR-138*, 1979.

[40] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Abridged summary of the third AIAA computational fluid dynamics drag prediction workshop," *Journal of aircraft*, Vol. 45, No. 3, 2008, pp. 781–798.

[41] Vassberg, J., Tinoco, E., Mani, M., Brodersen, O., Eisfeld, B., Wahls, R., Morrison, J., Zickuhr, T., Laflin, K., and Mavriplis, D., "Summary of DLR-F6 wing-body data from the third AIAA CFD Drag Prediction Workshop," *RTO AVT-147 Paper*, Vol. 57, 2007.

[42] Massegur Sampietro, D., and Da Ronch, A., "Recurrent Multi-Mesh Convolutional Autoencoder Framework for Spatio-Temporal Aerodynamic Modelling." *AIAA AVIATION 2023 Forum*, 2023, p. 3845.

[43] Heeg, J., Chwalowski, P., Raveh, D. E., Dalenbring, M. J., and Jirasek, A., "Plans and example results for the 2nd AIAA aeroelastic prediction workshop," *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 0437.

[44] Heeg, J., Chwalowski, P., Schuster, D., and Dalenbring, M., "Overview of the Aeroelastic Prediction Workshop," *51st AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics Inc, AIAA, 2013. https://doi.org/10.2514/6.2013-783.

[45] Heeg, J., and Chwalowski, P., "Investigating the transonic flutter boundary of the benchmark supercritical wing," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 0191.

[46] Heeg, J., Chwalowski, P., Raveh, D. E., Jirasek, A., and Dalenbring, M., "Overview and data comparisons from the 2nd Aeroelastic Prediction Workshop," *34th AIAA Applied Aerodynamics Conference*, 2016, p. 3121.

[47] Wiener, N., "The homogeneous chaos," *American Journal of Mathematics*, Vol. 60, No. 4, 1938, pp. 897–936.

[48] Ghanem, R. G., and Spanos, P. D., "Spectral stochastic finite-element formulation for reliability analysis," *Journal of Engineering Mechanics*, Vol. 117, No. 10, 1991, pp. 2351–2372.

[49] Xiu, D., and Karniadakis, G. E., "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM journal on scientific computing*, Vol. 24, No. 2, 2002, pp. 619–644.

[50] Sudret, B., "Global sensitivity analysis using polynomial chaos expansions," *Reliability engineering & system safety*, Vol. 93, No. 7, 2008, pp. 964–979.

[51] Hosder, S., Walters, R. W., and Balch, M., "Point-collocation nonintrusive polynomial chaos method for stochastic computational fluid dynamics," *AIAA journal*, Vol. 48, No. 12, 2010, pp. 2721–2730.

[52] Stanford, B. K., and Massey, S. J., "Uncertainty Quantification of the Fun3d-Predicted Nasa Crm Flutter Boundary," *SciTech Forum*, 2017.

[53] Stanford, B. K., and Roy, S., "Sizing and Topology Design of an Aeroelastic Wingbox Under Uncertainty," *International Forum on Aeroelasticity and Structural Dynamics*, 2019.

[54] Rahman, S., "Wiener–Hermite polynomial expansion for multivariate Gaussian probability measures," *Journal of Mathematical Analysis and Applications*, Vol. 454, No. 1, 2017, pp. 303–334.

[55] Kammeyer, M., "Wind tunnel facility calibrations and experimental uncertainty," *20th AIAA Advanced Measurement and Ground Testing Technology Conference*, 1998, p. 2715.

[56] Owen, F. K., and Owen, A. K., "Measurement and assessment of wind tunnel flow quality," *Progress in aerospace sciences*, Vol. 44, No. 5, 2008, pp. 315–348.

[57] Amiri, K., Soltani, M. R., and Haghiri, A., "Steady flow quality assessment of a modified transonic wind tunnel," *Scientia Iranica*, Vol. 20, No. 3, 2013, pp. 500–507.