The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

UNCERTAINTY-AWARE TRANSONIC FLOW PREDICTIONS WITH BAYESIAN DEEP LEARNING

Mehdi Anhichem[1], Sebastian Timme[1], Jony Castagna[2], Andrew J. Peace[3] and Moira Maina[3]

[1] University of Liverpool, Liverpool, L69 3GH, UK
m.anhichem@liverpool.ac.uk

[2] UKRI-STFC Hartree Centre, Warrington, WA4 4AD, United Kingdom

[3] Aircraft Research Association Ltd, Bedford, MK41 7PF, United Kingdom

Key words: Transonic flows, computational fluid dynamics, surrogate modelling, machine learning, epistemic uncertainty

Summary. Transonic flows pose unique challenges and exhibit distinctive phenomena due to the interaction of subsonic and supersonic flow regions. One notable characteristic is the formation of shock waves, that come with abrupt changes in the flow quantities, playing a crucial role in transonic aerodynamics. The unsteadiness in transonic flows arises from the dynamic nature of shock waves and their sensitivity to perturbations. A careful study of transonic flows to optimise aerodynamic designs, mitigate adverse effects, and enhance the overall stability and efficiency of commercial flight is essential. High-fidelity computational fluid dynamics (CFD) simulations can provide invaluable insights into such flows. Nevertheless, the prohibitive costs associated with these simulations limit their feasibility, particularly for tasks involving a large number of evaluations. There is a need for aerodynamic data models that possess both speed and accuracy, ensuring the ability to confidently predict viable design alternatives. Many engineering design problems encounter similar challenges and address them by employing approximate mathematical models known as surrogate models. A surrogate model functions as an agent that learns the relationship between a system input and its output, subsequently being used to predict outputs at unevaluated inputs. Given that these models approximate the quantity of interest across the design space, it becomes essential to assess the accuracy and uncertainty of their predictions throughout this space. Machine learning presents tools capable of offering prediction estimates along with associated uncertainties. The objective of this work is to leverage such model, specifically Bayesian neural network, and to propose specific improvements for transonic flows. It is applied to steady-state high-fidelity CFD simulations of an aerofoil geometry.

1 INTRODUCTION

Transonic flows, characterised by a mix of subsonic and supersonic regions, present unique challenges for aerodynamic analysis due to complex flow features such as shock waves and their interaction with boundary layers. High-fidelity CFD simulations are essential for capturing these intricate details with accuracy, particularly for the purpose of predicting pressure distributions across the aircraft surface. Research is being pursued to push the boundaries of CFD and to improve modelling of transonic flows [1, 2, 3]. Applying CFD may encounter difficulties, as it can be costly and time-consuming, particularly in contexts where multiple scenarios must be evaluated. As a result, aerodynamic data models are required to deliver both speed and accuracy to aerodynamic studies. Similar challenges arise in many engineering design problems, where they are often addressed by using surrogate models.

Machine learning is transforming scientific discovery by enabling the detection of patterns and predictions from large data sets [4], offering a powerful tool for surrogate modelling tasks. Machine learning such as neural networks are increasingly being adopted in the aerospace industry, offering new ways to optimise design, streamline manufacturing, enhance predictive maintenance, and improve flight operations [5]. The widespread adoption of deep neural networks across various fields, including fluid mechanics and aerodynamics, is largely due to their remarkable expressive power, as explained by the universal approximation theorem [6]. This theorem states that a neural network with at least one hidden layer and a non-linear activation function can approximate any continuous function to a desired degree of accuracy. Depending on the problem at hand, different types and architectures of neural networks can be employed in aerodynamic applications. For instance, a multilayer perceptron and a convolutional neural network were used in [7] and [8], respectively, to compute the aerodynamic forces of an aerofoil. Unsteady (multidisciplinary) phenomena, such as flutter speed or buffet pressure loads predictions, were considered in [9, 10, 11]. A comparative study on pressure distribution predictions for various geometries and input space dimensions using Gaussian processes, interpolation with proper orthogonal decomposition and deep neural networks is provided in [12].

As surrogate models are approximation techniques, questioning their prediction quality is crucial. This involves here epistemic uncertainty, which arises from the model's limited knowledge due to incomplete coverage of the design space by the training data. Bayesian methods help quantify this uncertainty, enabling Bayesian surrogate models to estimate both the output and its associated uncertainty. Bayesian Neural Network (BNN) is a type of neural network that incorporates Bayesian inference to make probabilistic predictions. Traditional neural networks make fixed predictions from learned weights and biases, while BNNs account for uncertainty by treating these parameters as random variables with prior distributions. BNNs have been used to optimise pressure tap locations in [13] and they demonstrate improvements over their non-Bayesian counterparts on various datas ets [14, 15]. The first application of BNNs for predicting transonic pressure distributions was presented in [16], with a main issue being the accurate prediction of the supersonic region over the aerofoil surface.

This paper proposes enhancements to BNNs for the specific prediction of pressure distributions under transonic flow conditions. It also addresses model calibration by evaluating the quality of the uncertainty estimates and evaluates a basic recalibration technique. These advancements are applied to Reynolds-averaged Navier-Stokes (RANS) simulation data for a two-dimensional aerofoil and are compared to a method from the literature based on proper orthogonal decomposition (POD). Progress is demonstrated on this test case, serving as a proof-of-concept for further exploration of BNN applications in transonic flow prediction.

2 METHODS

2.1 Bayesian Surrogate Models

Bayesian modelling applies Bayes' theorem to update knowledge about model parameters based on observed data. Let f represent a surrogate model that maps the input vector $\mathbf{x} \in \mathcal{D}$ to the observed scalar output y, where $\mathcal{D} \subseteq \mathbb{R}^d$ is the design space of dimension d. This can be expressed as $y = f(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the vector of model parameters. For example, in linear regression, these parameters include the intercept and slope, while in deep learning, they are the weights and biases of the neural network. In Bayesian modelling, instead of assigning a single value to parameters, probability distributions are used to capture information about the relationship between $\mathbf{x}$ and y. Given the prior distribution of the parameters $p(\boldsymbol{\theta})$ and the likelihood function $p(y|\mathbf{x}, \boldsymbol{\theta})$, Bayes' theorem is applied to compute the posterior distribution,

$$p(\boldsymbol{\theta}|y, \mathbf{x}) = \frac{p(y|\mathbf{x}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(y|\mathbf{x})} \tag{1}$$

with $p(y|\mathbf{x})$ being the marginal likelihood. It corresponds to the integrated likelihood function over the parameter space $\boldsymbol{\theta}$. Predictions at unseen inputs $\mathbf{x}^*$ are carried out through the posterior distribution by substituting it into the expression of the predictive distribution,

$$p(y^*|\mathbf{x}^*, \mathbf{x}, y) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|y, \mathbf{x})\, \mathrm{d}\boldsymbol{\theta} \tag{2}$$

In practice, computing the exact posterior distribution is often infeasible due to the complexity of the models and the large volume of data involved. In such cases, variational inference is commonly employed to approximate the true posterior distribution. This method involves substituting the true posterior p with a variational distribution q of known form. The goal is to optimise the parameters of q to minimise its divergence from p, thereby transforming the inference problem into an optimisation task. The Kullback–Leibler (KL) divergence [17] measures the difference between these distributions, and optimising this divergence yields both the model and variational parameters. This variational distribution is then used in place of the true posterior for making predictions, as shown in Eq. (2). More detailed discussions and applications can be found in [18, 19].

2.2 Bayesian Neural Network with Monte Carlo Dropout

Deep learning, a subset of machine learning inspired by the brain's neural networks, tackles complex tasks by learning hierarchical data representations. It employs deep neural networks with multiple interconnected layers of artificial neurons that process and transform input data through activation functions. These networks are trained on large data sets, with internal parameters optimised via backpropagation to minimise prediction errors. Despite their widespread use, standard neural networks frequently suffer from overfitting, which complicates their generalisation, and they lack a measure of the model uncertainty. Bayesian deep learning addresses these issues effectively. Unlike standard neural networks, which provide deterministic predictions based on fixed weights and biases, Bayesian neural networks treat these parameters as random variables with prior probability distributions. During training, the prior distribution is updated to a posterior distribution using Bayes' rule, reflecting parameter uncertainty given the data. This posterior distribution allows for predictions that capture uncertainty, offering a range of possible outcomes rather than a single estimate. Bayesian inference techniques, such as Markov Chain Monte Carlo and variational inference, are used to derive the posterior distribution. For a comprehensive mathematical overview, refer to [20], and extensive reviews of training methods can be found in [21].

Monte Carlo Dropout (MCD), a popular variational inference technique introduced by [14], is widely used for training BNN. MCD approximates the posterior distribution over network weights by employing dropout regularisation, a process in which a random subset of neurons is deactivated during each forward pass throughout an epoch, which is a complete pass through the training data. This procedure effectively generates an ensemble of models, each with a unique subset of active neurons, and helps to prevent overfitting. While MCD is computationally efficient and easy to implement, it has limitations in capturing complex forms of uncertainty and can lead to challenges in producing well-calibrated uncertainty estimates. During training, the network parameters $\boldsymbol{\theta}$, including weights and biases, are optimised by minimising the cost function (or objective function),

$$\mathcal{L}_{\mathrm{MCD}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\|\boldsymbol{\theta}\|^2 \tag{3}$$

where λ is a decay of the L_2 regularisation term. The first term represents the mean squared error calculated on the training dataset y, which consists of N samples, and the corresponding predictions $\hat{y}$ from the neural network. During testing, instead of performing a single forward pass through the network, multiple forward passes are executed with different neurons randomly dropped out. The outputs from these passes are averaged to generate a final prediction for a test input. These outputs are also used to estimate the model's predictive uncertainty. Assuming a standard neural network with weights and biases $\boldsymbol{\theta}$, the output of the network for a given input $\mathbf{x}$ can be written as $\hat{y}(\mathbf{x};\boldsymbol{\theta})$. For the BNN model here, the T forward passes made with different dropout masks result in a set of outputs $\{\hat{y}(\mathbf{x};\boldsymbol{\theta}_t)\}_{t=1}^{T}$. The predictive distribution $p(y^*|\mathbf{x}^*,\mathbf{x},y) = \mathcal{N}(\mu_{\mathrm{MCD}}(\mathbf{x}^*),\sigma_{\mathrm{MCD}}(\mathbf{x}^*))$ at unseen point $\mathbf{x}^*$ over the network's outputs is approximated by averaging over this set of outputs. The mean and variance of the predictive distribution can be approximated by,

$$\mu_{\mathrm{MCD}}(\mathbf{x}^*) \approx \frac{1}{T}\sum_{t=1}^{T}\hat{y}(\mathbf{x}^*;\boldsymbol{\theta}_t) \tag{4}$$

$$\sigma^2_{\mathrm{MCD}}(\mathbf{x}^*) \approx \frac{1}{T-1}\sum_{t=1}^{T}\big(\hat{y}(\mathbf{x}^*;\boldsymbol{\theta}_t) - \mu_{\mathrm{MCD}}\big)^2 \tag{5}$$

This method offers computational efficiency by leveraging dropout layers, which are already available in common deep learning frameworks. The training process remains identical to that of standard neural networks with dropout layers. The key distinction arises during the prediction phase, where sampling is introduced to achieve a BNN. The BNN implementation is adapted from the work of [14] [1] and built on TensorFlow using the Keras API [22]. This implementation was applied to surface pressure distribution prediction in [16] and demonstrated promising results in predicting transonic flow features such as shock waves. However, a notable issue was the inaccurate prediction of the pressure coefficient in the supersonic region, leading to an imprecise evaluation of the shock, despite correctly predicting its location. In neural networks, parameter tuning is guided by the loss function, see Eq. (3), and here the idea is to embed more flow physics into the loss function by weighting it with data points specifically located in the supersonic region.

To know if a predicted pressure coefficient $\hat{y}_i$ is supersonic, it can be compared to the critical pressure coefficient. The critical pressure coefficient $C_{p,\text{crit}}$ is the pressure coefficient at which the local flow velocity first reaches the speed of sound, indicating the onset of sonic conditions on the surface [23]. For an aerofoil, it can be expressed as a function of the Mach number M:

$$C_{p,\text{crit}} = \frac{2}{\gamma M^2}\left[\left(\frac{(\gamma-1)M^2+2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} - 1\right] \tag{6}$$

where γ is the heat capacity ratio. When $\hat{y}_i$ falls below $C_{p,\text{crit}}$, the predicted value lies within the supersonic region. If $\hat{y}_i$ is in the supersonic region, an additional term is introduced into the loss function, proportional to the squared difference between y_i and $C_{p,\text{crit}}$. This penalises points within the supersonic region, adjusting the loss function to account for these specific points. The resulting loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{dropout}} + \mathcal{L}_{\text{crit}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 + \lambda\|\boldsymbol{\theta}\|^2 + \frac{k_{\text{crit}}}{N_{\text{crit}}}\sum_{i=1}^{N_{\text{crit}}}(C_{p,\text{crit}} - \hat{y}_i)^2 \tag{7}$$

where N_{crit} is the number of points in the supersonic region and k_{crit} is a scalar weighting this additional term. The latter can be treated as a hyperparameter of the neural network. Although it can be optimised using Bayesian methods along with the network architecture, in this case, it was selected through trial and error and set to two.

2.3 POD+GP

The technique known in the literature as POD with interpolation is added as a reference in this paper. It combines the calculation of POD modes and their coefficients with the interpolation of these coefficients to make new predictions [24, 25, 26, 27, 12]. POD is a dimensionality reduction method that identifies the dominant spatial patterns or modes from a dataset. This technique, referred to as the Karhunen-Loève procedure or principal component analysis (PCA) in statistics, is widely used across various fields. Initially applied in fluid mechanics by [28] to extract coherent structures from turbulent flows, POD has since been utilized for analyzing turbulence [29] and other aerodynamic phenomena, such as shock buffet [30].

There are two primary methods for performing POD: the direct method using singular value decomposition and an alternative known as the snapshot method. The direct method provides accurate results but can be computationally intensive when the matrix being decomposed has more rows than columns. Let N_{snap} be the number of snapshots and N_{points} the number of points

[1] http://www.github.com/yaringal/DropoutUncertaintyExps

per snapshot. In aerodynamic applications, snapshots correspond to different timesteps or flow conditions while points correspond to the data points on the surface or in the volume. The data can be organised into an $N_{\text{points}} \times N_{\text{snap}}$ matrix denoted as $\boldsymbol{\mathcal{S}}$. Instead of applying the singular value decomposition on $\boldsymbol{\mathcal{S}}$, the method of snapshots considers the eigenvalue decomposition of $\boldsymbol{\mathcal{S}}^\top \boldsymbol{\mathcal{S}}$,

$$\boldsymbol{\mathcal{S}}^\top \boldsymbol{\mathcal{S}} \boldsymbol{\psi}_j = \lambda_j \boldsymbol{\psi}_j, \quad \boldsymbol{\psi}_j \in \mathbb{R}^{N_{\text{snap}}}, j \in [\![1; N_{\text{snap}}]\!] \tag{8}$$

where $\boldsymbol{\psi}_j$ are the eigenvectors and λ_j the eigenvalues of the matrix $\boldsymbol{\mathcal{S}}^\top \boldsymbol{\mathcal{S}}$. The POD modes are then computed and given as

$$\boldsymbol{\phi}_j = \boldsymbol{\mathcal{S}} \boldsymbol{\psi}_j \lambda_j^{-\frac{1}{2}}, \quad \boldsymbol{\phi}_j \in \mathbb{R}^{N_{\text{snap}}}, \tag{9}$$

or in matrix form,

$$\boldsymbol{\Phi} = \boldsymbol{\mathcal{S}} \boldsymbol{\Psi} \Lambda^{-\frac{1}{2}} = \boldsymbol{\mathcal{S}} \boldsymbol{\mathcal{V}}^{-1}, \quad \boldsymbol{\Phi} \in \mathbb{R}^{N_{\text{points}} \times N_{\text{snap}}} \tag{10}$$

where $\boldsymbol{\mathcal{V}}$ represents the resulting POD coefficients organised in an $N_{\text{snap}} \times N_{\text{snap}}$ matrix. In most scenarios involving the study of fluid mechanics, the snapshot matrix has a tall and narrow structure ($N_{\text{snap}} \ll N_{\text{points}}$), signifying the presence of numerous degrees of freedom compared to the number of snapshots. In this case, using the snapshot method provides a faster but slightly less accurate solution [29]. Equation 10 shows that the matrix $\boldsymbol{\mathcal{S}}$ can be expressed as a linear combination of the POD modes Φ,

$$\boldsymbol{\mathcal{S}} = \boldsymbol{\Phi} \boldsymbol{\mathcal{V}}. \tag{11}$$

More specifically, each individual snapshot can be reconstructed as a linear combination of the POD modes, denoted as $\boldsymbol{s}_i = \boldsymbol{\Phi} \boldsymbol{v}_i$, where $\boldsymbol{s}_i$ and $\boldsymbol{v}_i$ correspond to the i-th column of $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{V}}$, respectively.

POD can then be used as a regression technique and generate "unseen" snapshots by interpolating the POD coefficients. Essentially, to predict values within the design space that are not part of the original snapshot matrix, it is necessary to estimate the coefficients linked to each mode at the predicted inputs. POD with interpolation involves selecting a limited number r of POD modes that account for the majority of the system's energy (determined by the ratio of the sum of the eigenvalues of the selected modes to the total energy). The interpolation process then uses the information from $\boldsymbol{\mathcal{V}}$ to calculate the corresponding POD coefficients. When considering a limited number of modes, Eq. (11) can be written, $\boldsymbol{s}_i = \boldsymbol{\Phi}_r \boldsymbol{v}_i$ where the matrix $\boldsymbol{\Phi}_r$ contains r POD modes and $i \in [\![1; N_{\text{snap}}]\!]$. It can be expanded into:

$$\begin{cases} \boldsymbol{s}_i = v_{11} \boldsymbol{\phi}_1 + \cdots + v_{1r} \boldsymbol{\phi}_r \\ \vdots \\ \boldsymbol{s}_{N_{\text{snap}}} = v_{N_{\text{snap}}1} \boldsymbol{\phi}_1 + \cdots + v_{N_{\text{snap}}r} \boldsymbol{\phi}_r \end{cases} \tag{12}$$

where each snapshot is associated with a flow condition (M, α). To predict a snapshot $\boldsymbol{s}$ at an unseen flow condition, r GP are trained on the data sets $\{(M, \alpha)_i, v_{ij} | i \in [\![1; N_{\text{snap}}]\!]\}$ for $j \in [\![1; r]\!]$. In other words, each GP predict the coefficient associated with each POD mode at the flow condition. These GP are trained on the coefficients obtained from the POD decomposition, given int the column of Eq. (12). When employing GP for interpolation, each GP offers a prediction uncertainty linked to POD coefficients. The predicted snapshot is derived by multiplying the GP predictions with the chosen POD modes. Consequently, a snapshot emerges from the product of means with their corresponding POD modes, while an uncertainty estimate stems from the product of standard deviations with their corresponding POD modes,

$$\mu_{\text{POD+GP}}(\mathbf{x}^*) = \mu_1(\mathbf{x}^*)\boldsymbol{\phi}_1 + \cdots + \mu_r(\mathbf{x}^*)\boldsymbol{\phi}_r \tag{13}$$

$$\sigma_{\text{MCD}}(\mathbf{x}^*) = \sigma_1(\mathbf{x}^*)\boldsymbol{\phi}_1 + \cdots + \sigma_r(\mathbf{x}^*)\boldsymbol{\phi}_r \tag{14}$$

where $\mu_1, \ldots, \mu_r$ and $\sigma_1, \ldots, \sigma_r$ are, respectively, the posterior means and standard deviations of the r GP. Hence, under this configuration, the model designated here as POD+GP is Bayesian.

2.4 Evaluation Criteria

At each input $\mathbf{x}^*$, Bayesian surrogate models provide a prediction and a measure of its uncertainty. To assess both quantities, metrics for the accuracy of both the model's predictions and its uncertainty estimates are introduced. The mean-squared error (MSE), is considered to evaluate the models' performance:

$$\text{MSE} = \frac{1}{N_t}\sum_{n=1}^{N_t}(y_n - \hat{y}_n)^2 \tag{15}$$

where N_t is the number of testing points. Lower MSE values are preferred, as they signify a better fit of the model. We also evaluate the 99^{th} percentile of the squared errors, denoted as SE_{99}, which reveals the value below which 99% of the squared errors fall. This percentile provides insight into the error distribution and the robustness of the model. While accurate predictions are essential for surrogate models, Bayesian models also require assessing the reliability of their uncertainty estimates. Ideally, a well-calibrated Bayesian model's uncertainty estimates should accurately reflect the true data distribution, meaning that, for example, a 90% posterior confidence interval should contain the true simulation outcome approximately 90% of the time. In the context of regression [32, 33], the surrogate model's uncertainty estimates are deemed well-calibrated when, as $N_t \to \infty$,

$$\frac{1}{N_t}\sum_{n=1}^{N_t}\mathbb{I}\{y_n \leq F_n^{-1}(p)\} \to p \text{ for all } p \in [0,1] \tag{16}$$

where $\mathbb{I}$ is the Boolean function and F_n is the cumulative distribution function (CDF) targeting y_n. Here, F_n^{-1} denotes the quantile function $F_n^{-1}(p) = \inf\{y : p \leq F_n(y)\}$. In other words, as the dataset grows to an infinite size, the empirical CDF should converge with the predicted CDF. To assess the quality of calibration, we measure the proportion of test data observations that fall within the confidence intervals predicted by the quantile function. We then visualize the calibration quality in Bayesian models using a calibration plot, where perfectly calibrated uncertainty estimates would align exactly with the diagonal line.

2.5 Model Recalibration

Model predictions are derived exclusively from the training data. The uncertainty estimates are not always well-calibrated, meaning that the predicted standard deviations may not align with the empirical errors observed in the data. When model predictions are systematically overconfident or underconfident, recalibration can be applied to scale the predicted standard deviations. Standard deviation scaling is a technique used to adjust the predicted uncertainties of a model to better reflect the true uncertainties observed in the data. If the model's standard deviations are well-calibrated, the standardised residuals should follow a standard normal distribution. The standardised residuals vector is computed on the validation set. First, the following operations are performed element-wise:

$$\frac{\mathbf{y} - \boldsymbol{\mu}_{\text{MCD}}}{\boldsymbol{\sigma}_{\text{MCD}}} \tag{17}$$

Then, the standard deviation of this vector, denoted as $\boldsymbol{\sigma}_{\text{emp}}$ for empirical standard deviation, serves as a scaling factor for the predicted standard deviations. The recalibrated standard deviation $\boldsymbol{\sigma}_{\text{recalibrated}}$ is then simply calculated by:

$$\boldsymbol{\sigma}_{\text{recalibrated}} = \boldsymbol{\sigma}_{\text{emp}} \times \boldsymbol{\sigma}_{\text{MCD}} \tag{18}$$

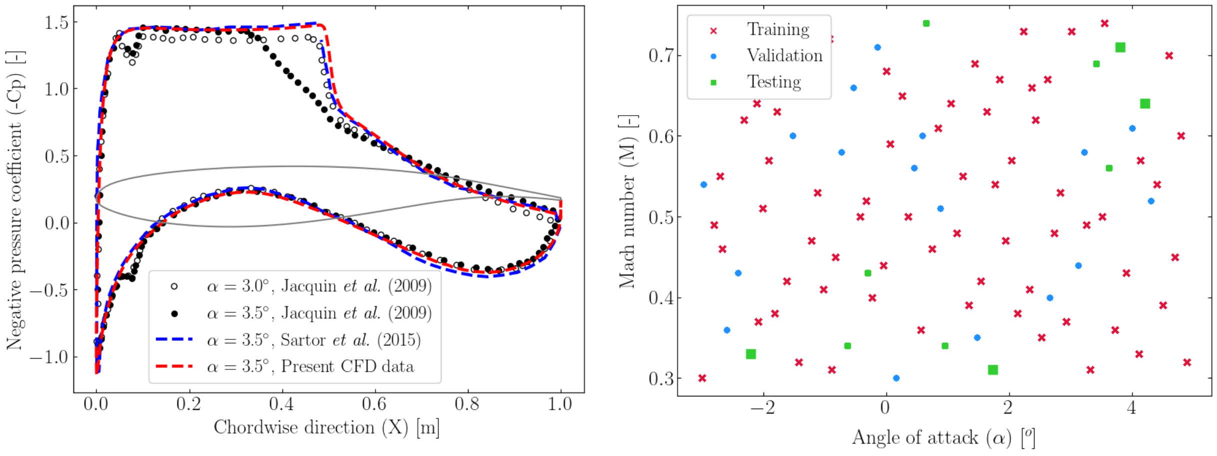
This approach effectively incorporates additional data information into the model calibration. Nevertheless, more sophisticated ad-hoc recalibration techniques, such as percentile-based isotonic regression [32], could offer further refinement.

3 TEST CASE

ONERA's OAT15A profile has been studied widely since its first experimental investigation in [34], such as for numerical shock-buffet stability analyses in [35, 36, 37]. To solve the non-linear governing equations, the TAU code of the German Aerospace Center (DLR), commonly used in the European aerospace industry, was chosen. It employs a cell-vertex second-order finite-volume spatial discretisation. The inviscid fluxes of the governing RANS equation are discretised via a central scheme that employs matrix artificial dissipation, whereas a first-order Roe upwind scheme is used for those of the turbulence model. For the computation of gradients of flow variables, required for viscous fluxes and the turbulence model's source term, the Green–Gauss theorem is employed. The no-slip adiabatic condition is strongly enforced on the solid walls of the aerofoil, and the far field is considered to be a free-stream flow through a characteristic boundary condition. The code's standard backward Euler scheme with the lower-upper symmetric Gauss–Seidel method, together with local time-stepping and multigrid for convergence acceleration, is chosen for obtaining steady-state solutions converged by seven orders of magnitudes in the density residual norm.

For the two-dimensional aerofoil simulations herein, a baseline mesh was generated that has a quasi-structured region in the near-field with an O-type topology around the blunt trailing edge, while unstructured triangular cells are employed towards the circular far-field boundary with a radius of 100 chord lengths. The domain is discretised with approximately 80,000 control volumes altogether, and the aerofoil surface is divided into 626 elements. Figure 1a illustrates the pressure coefficient in comparison to corresponding experimental data from [34] at two specific angles of attack, specifically around the onset of shock buffet. Additionally, it includes numerical results for the same two-dimensional aerofoil obtained from [37], using a very similar version of the turbulence model. The simulations at $\alpha = 3.5°$ demonstrate good agreement with the wind-tunnel data at $\alpha = 3.0°$. This discrepancy is often reported and attributed to the particular choice of turbulence model.

Along with the aerofoil surface coordinates (X, Z), the Mach number M and angle of attack α are considered as input space parameters. The Reynolds number is assumed constant at



(a)

(b)

Figure 1: (a) Pressure coefficient results at Mach number $M = 0.73$ and Reynolds number $Re = 3 \times 10^6$ compared with experiments from [34] and steady-state two-dimensional aerofoil results from [37] and (b) and design of experiment for present OAT15A test case.

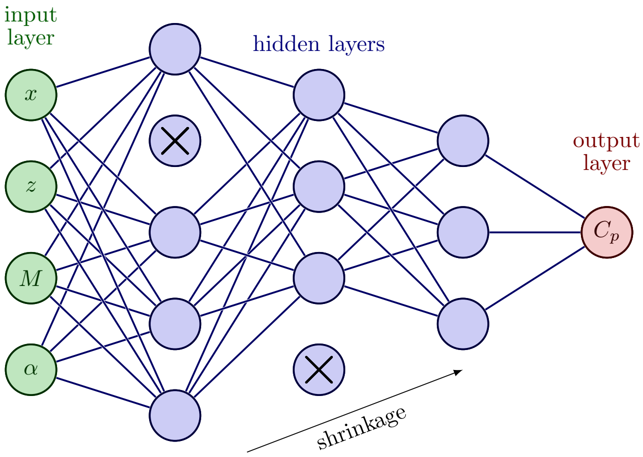


Figure 2: Bayesian neural network with MCD architecture example.

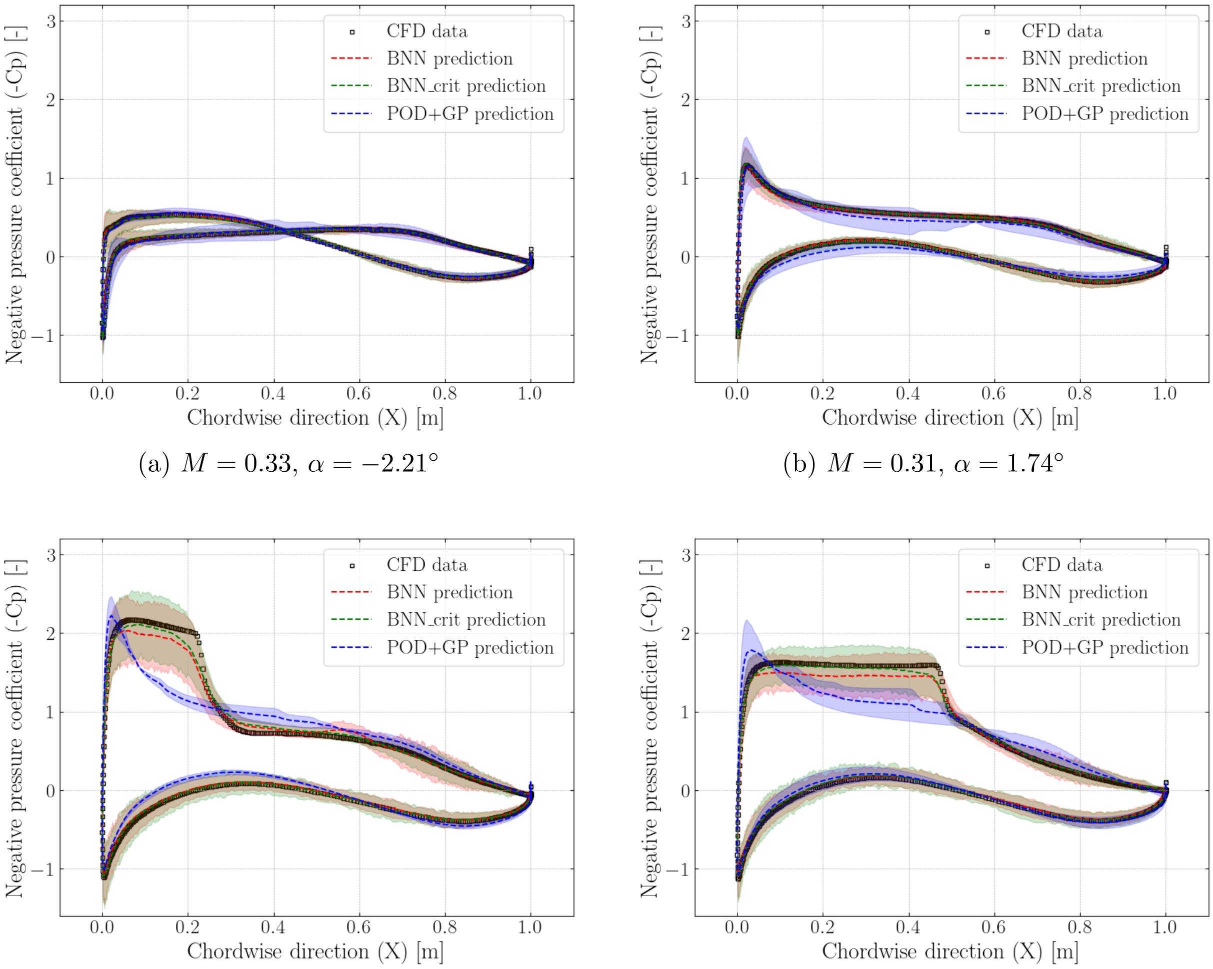
$Re = 3 \times 10^6$ throughout. Figure 2 presents the BNN architecture for this four-dimensional input space with C_p as output. Following a Halton sequence [38], as outlined in [12], 98 steady-state simulations have been carried out within the parameter ranges $M = [0.3, 0.75]$ and $\alpha = [-3°, 5°]$, leading to a total of 61,348 data points. Each M–α combination has a complete set of (X, Z) coordinates. Figure 1b shows the sample locations of the numerical experiments which are divided into three subsets; specifically 70% for training, 20% for validation and 10% for testing.

4 RESULTS

To evaluate the proposed improvements, a BNN model was trained on the presented test case, alongside another BNN model incorporating a modified loss function, as explained in Eq. (7) and

Table 1: Evaluation metrics of Bayesian models on test set.

Model	MSE	SE_{99}
BNN	0.0046	0.0992
BNN_crit	**0.0024**	**0.0516**
POD+GP	0.0156	0.3086



(a) $M = 0.33$, $\alpha = -2.21°$

(b) $M = 0.31$, $\alpha = 1.74°$

(c) $M = 0.64$, $\alpha = 4.21°$

(d) $M = 0.71$, $\alpha = 3.81°$

Figure 3: Comparison between predictions made by the models on four points of the test set (highlighted in Fig. 1b with bigger symbols).

referred to as BNN_crit. These models are compared against the POD+GP model, commonly used as a reference in the literature. The architecture for the BNN models (see Fig. 2), adopted from [16], was determined through Bayesian optimisation [39]. It features three hidden layers with 384 neurons in the first layer and a shrinkage rate of 0.661. The final BNN architectures are

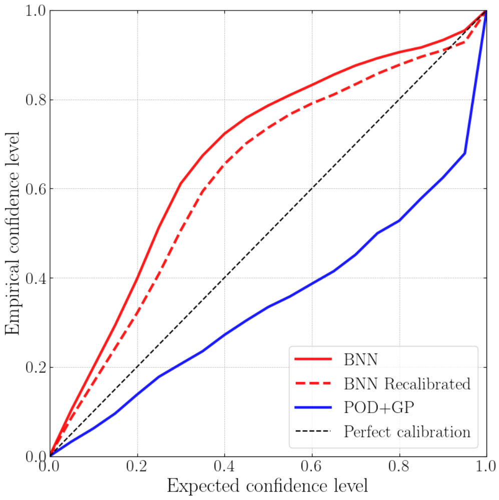
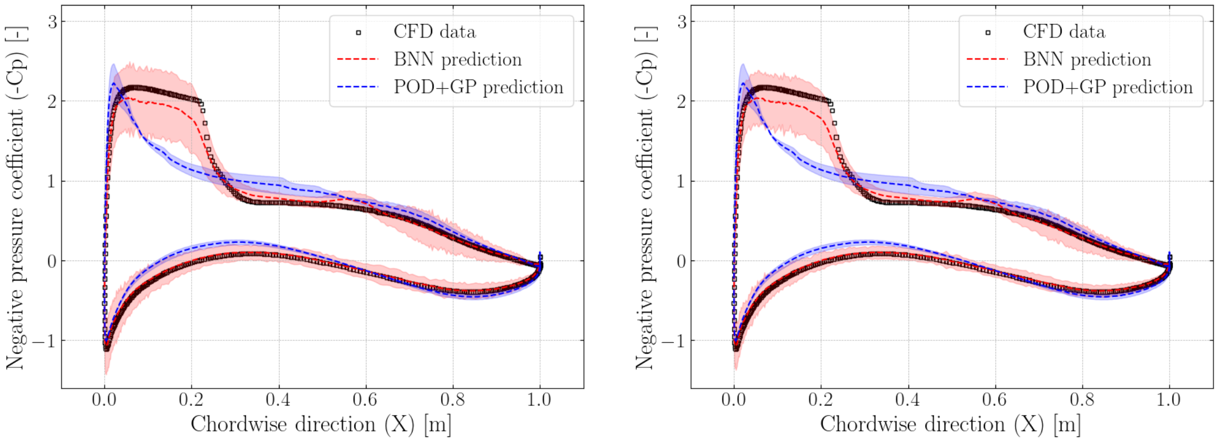


Figure 4: Calibration plot assessing the different model standard deviations on the OAT15A test case.

trained for 1,000 epochs using the Adam optimiser [40], with the default learning rate of 1×10^{-3} applied to minimise the mean-squared error. Following training, the predictive distribution is approximated by running T forward passes through the trained network, each using different dropout masks. As demonstrated in [16], both the predicted mean and standard deviation converge within a 1% range of μ_{MCD} after approximately 1,000 samples, although 2,000 samples are used here to achieve a more refined predictive distribution.

Table 1 summarises the results for the metrics for the test set. Predictions on the test set have been carried out to evaluate the models' accuracy on unseen data. The results reveal differences in performance between the approaches. In general, BNN models demonstrate a more than threefold reduction in MSE and SE_{99} compared to POD+GP, showing its superior capability to closely approximate the correct values. These differences in accuracy are evident in Fig. 3, where at higher Mach numbers and angles of attack (Figs. 3c and 3d), significant discrepancies are observed in the ability of the POD+GP model to predict the shock location on the aerofoil's upper surface. This aligns well with the observations reported in [12, 16]. When comparing the two BNN models, identical predictions are observed at lower Mach numbers and incidence angles (Figs. 3a and 3b), as expected, since no supersonic regions are present at these flow conditions. However, in the transonic regime, BNN_crit provides a closer match to the reference CFD data when predicting shock waves. BNN_crit accurately captures the supersonic regions, while the standard BNN overestimates the pressure coefficient. Both models predict the shock location similarly, but discrepancies arise in the transition between the supersonic plateau and the shock for both models. These findings generalise to the entire test set, with a reduction by a factor of two in MSE and SE_{99} achieved by incorporating an additional term in the loss function. This demonstrates the effectiveness of this approach in improving transonic flow predictions. Alternative formulations of the loss function accounting for supersonic (and/or separated) flow features can be considered in the future.

A key advantage of Bayesian models is their ability to estimate epistemic uncertainty, shown as transparent regions in Fig. 3. This provides a measure of confidence in the predicted val-



(a) Before recalibration.

(b) After recalibration.

Figure 5: A comparison of standard deviation predictions made by the models at $M = 0.64$ and $\alpha = 4.21°$ before and after recalibration.

ues, especially in areas distant from the training data. For a thorough uncertainty analysis, a calibration plot is presented in Fig. 4, following the methodology outlined by [32], as described in Section 2.4. Uncertainty is evaluated by comparing the expected probabilities of specific outcomes with the empirically observed rates across twenty defined intervals. The model's calibration plot before recalibration, presented in Section 2.5, are represented by a solid red line, while the recalibrated model's calibration plot is depicted using a red dashed line. The BNN tends to overestimate uncertainty, while the POD+GP underestimates it, as reflected in the confidence intervals in Figs. 3 and 5. After applying recalibration (via standard deviation scaling), the recalibrated model, shown by the dashed line, moves closer to the diagonal, indicating better calibration. Although the improvement is modest, it demonstrates the potential of this approach to enhance Bayesian surrogate model predictions, especially since a simple recalibration method was used here. Fig. 5 shows the recalibration's effect on the confidence intervals, which become smaller in Fig. 5b compared with Fig. 5a, particularly in the supersonic region where the standard deviation was initially overestimated.

5 CONCLUSIONS

This paper builds upon the comparison conducted in [16] and contributes to the broader effort to develop uncertainty-aware surrogate models for surface pressure distributions in transonic flows. It introduces a specific enhancement to a BNN model, which is tested on a two-dimensional case, along with the assessment of a simple recalibration technique.

Neural networks have demonstrated significant potential for predicting flow features in transonic conditions due to their capability to capture complex non-linearities, offering advantages over more traditional methods like POD+GP. Despite this, challenges remain, as discrepancies in predictions persist in the supersonic region and shock resolution. The incorporation of a penalty term based on the critical pressure coefficient has led to some improvements, indicating a promising direction for further advancements in transonic flow modelling. The modification of the loss function is not specific to BNN and can be applied to any type of multilayer perceptron. However, the definition of $C_{p,\mathrm{crit}}$ provided here is specific to aerofoils [23], complicating its application to three-dimensional configurations.

This paper also proposes the evaluation of a recalibration technique as a proof-of-concept for this application. Note that this recalibration method can be applied to any type of Bayesian surrogate model, as it is an ad-hoc process that directly adjusts the model's outputted standard deviation. While more sophisticated recalibration methods are planned for future exploration, the integration of additional data-driven information through standard deviation scaling has demonstrated a positive impact on model calibration. This approach has improved the reliability of predictions and helped reduce the model's dependency on the specific training data, thereby enhancing the robustness of the model.

ACKNOWLEDGEMENTS

The first author is grateful for the financial support by the Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Distributed Algorithms (grant number 2447391 as part of EP/S023445/1) in partnership with Aircraft Research Association Ltd. We thank the German Aerospace Center for access to the TAU flow solver and the University of Liverpool for computing time on the high-performance computing system. We also acknowledge the technical support provided by the UK Research and Innovation (UKRI) Science and Technology Facilities Council (STFC) Hartree centre. We thank ONERA for making the OAT15A aerofoil geometry available. The simulation data that support the findings of this study are available from the authors upon reasonable request.

REFERENCES

[1] Masini, L. and Timme, S. and Peace, A. J. 2020. "Scale-Resolving Simulations of a Civil Aircraft Wing Transonic Shock-Buffet Experiment." AIAA Journal, **58**, no. 10, 4322-4338. http://doi.org/10.2514/1.J059219

[2] Herr, M. and Radespiel, R. and Probst, A. 2023. "Improved Delayed Detached Eddy Simulation with Reynolds-stress background modelling." Computers & Fluids, **265**, 106014. http://doi.org/10.1016/j.compfluid.2023.106014

[3] Petrocchi, A. and Barakos, G. 2024. "Transonic buffet simulation using a partially-averaged Navier–Stokes approach." Aerospace Science and Technology, **149**, 109134. http://doi.org/10.1016/j.ast.2024.109134

[4] Hey, T. and Tansley, S. and Tolle, K.M. 2009. The fourth paradigm: data-intensive scientific discovery. Redmond, WA: Microsoft Research.

[5] Brunton, S.L. and Kutz, J.N. and Manohar, K. and Aravkin, A.Y. and Morgansen, K. and Klemisch, J. and Goebel, N. and Buttrick, J. and Poskin, J. and Blom-Schieber, A.W. and Hogan, T. and McDonald, D. 2021. "Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning." AIAA Journal, **59**, no. 8, 2820-2847. http://doi.org/10.2514/1.J060131

[6] Hornik, K. and Stinchcombe, M. and White, H. 1989. "Multilayer feedforward networks are universal approximators." Neural Networks, **2**, no. 5, 359-366. http://doi.org/10.1016/0893-6080(89)90020-8

[7] Moin, H. and Zeeshan Iqbal Khan, H. and Mobeen, S. and Riaz, J. 2022. "Multilayer feedforward networks are universal approximators." 2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST). Islamabad, Pakistan: IEEE. http://doi.org/10.1109/IBCAST54850.2022.9990112

[8] Zhang, Y. and Sung, W.J. and Mavris, D.N. 2018. "Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient." 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. Kissimmee, FL: American Institute of Aeronautics and Astronautics. http://doi.org/10.2514/6.2018-1903

[9] Wang, Y. and Wang, Y. 2021. "Flutter speed prediction by using deep learning." Advances in Mechanical Engineering, **13**, 168781402110622. http://doi.org/10.1177/16878140211062275

[10] Tekaslan, H.E. and Demiroglu, Y. and Nikbay, M. 2022. "Surrogate Unsteady Aerodynamic Modeling with Autoencoders and LSTM Networks." AIAA SCITECH 2022 Forum. San Diego, CA: American Institute of Aeronautics and Astronautics. http://doi.org/10.2514/6.2022-0508

[11] Zahn, R. and Weiner, A. and Breitsamter, C. 2023. "Prediction of wing buffet pressure loads using a convolutional and recurrent neural network framework." CEAS Aeronautical Journal, **15**, 61–77. http://doi.org/0.1007/s13272-023-00641-6

[12] Sabater, C. and Stürmer, P. and Bekemeyer, P. 2022. "Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques." AIAA Journal, **60**, no. 9, 5249-5261. http://doi.org/10.2514/1.J061234

[13] Shen, W. and Huan, X. and Zhou, B.Y. and Gauger, N.R. 2020. "Towards Design of Airfoil Pressure Tap Locations for Real-Time Predictions Under Uncertainty Using Bayesian Neural Networks." AIAA SCITECH 2020 Forum. Orlando, FL: American Institute of Aeronautics and Astronautics. http://doi.org/10.2514/6.2020-0906

[14] Gal, Y. and Ghahramani, Z. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." Proceedings of The 33rd International Conference on Machine Learning. New-York, NY: PMLR. http://proceedings.mlr.press/v48/gal16.pdf

[15] Wang, K. and Du, H. and Jia, R. and Jia, H. 2022. "Performance Comparison of Bayesian Deep Learning Model and Traditional Bayesian Neural Network in Short-Term PV Interval Prediction." Sustainability, **14**, 12683. http://doi.org/0.3390/su141912683

[16] Anhichem, M. and Timme, S. and Castagna, J. and Peace, A. and Maina, M. 2024. "TBayesian Approaches for Efficient and Uncertainty-Aware Prediction of Pressure Distributions." AIAA SCITECH 2024 Forum. Orlando, FL: American Institute of Aeronautics and Astronautics. http://doi.org/10.2514/6.2024-0253

[17] Kullback, S. and Leibler, R.A. 2022. “On Information and Sufficiency.” The Annals of Mathematical Statistics, **22**, no. 1, 79-86. http://doi.org/10.1214/aoms/1177729694

[18] Wainwright, M.J. and Jordan, M.I. 2007. “Graphical Models, Exponential Families, and Variational Inference.” Foundations and Trends in Machine Learning, **1**, no. 1-2, 1-305. http://doi.org/10.1561/2200000001

[19] Blei, D.M. and Kucukelbir, A. and McAuliffe, J.D. 2017. “Variational Inference: A Review for Statisticians.” Journal of the American Statistical Association, **112**, no. 518, 859-877. http://doi.org/10.1080/01621459.2017.1285773

[20] Blundell, C. and Cornebise, J. and Kavukcuoglu, K. and Wierstra, D. 2015. “Weight Uncertainty in Neural Network.” Proceedings of the 32nd International Conference on Machine Learning. Lille, France: PMLR. http://proceedings.mlr.press/v37/blundell15.pdf

[21] Magris, M. and Iosifidis, A. 2023. “Bayesian learning for neural networks: an algorithmic survey.” Artificial Intelligence Review, **56**. http://doi.org/10.1007/s10462-023-10443-1

[22] Abadi, M. and Agarwal, A. and Barham, P. and Brevdo, E. and Chen, Z. and Citro, C. and Corrado, G.S. and Davis, A. and Dean, J. and Devin, M. and Ghemawat, S. and Goodfellow, I. and Harp, A. and rving, G. and Isard, M. and Jia, Y. and Jozefowicz, R. and Kaiser, L. and Kudlur, M. and Levenberg, J. and Mane, D. and Monga, R. and Moore, S. and Murray, D. and Olah, C. and Schuster, M. and Shlens, J. and Steiner, B. and Sutskever, I. and Talwar, K. and Tucker, P. and Vanhoucke, V. and Vasudevan, V. and Viegas, F. and Vinyals, O. and Warden, P. and Wattenberg, M. and Wicke, M. and Yu, Y. and Zheng, X. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from https://www.tensorflow.org/.

[23] Houghton, E.L. and Carpenter, P.W. and Collicott, S.H. and Valentine, D.T. 2013. Aerodynamics for Engineering Students (Sixth Edition). Butterworth-Heinemann.

[24] Zimmermann, R. and Görtz, S. 2012. “Improved extrapolation of steady turbulent aerodynamics using a non-linear POD-based reduced order model.” The Aeronautical Journal, **116**, no. 1184, 1079-1100. http://doi.org/10.1017/S0001924000007491

[25] Fossati, M. and Habashi, W.G. 2013. “Multiparameter analysis of aero-icing problems using proper orthogonal decomposition and multidimensional interpolation.” AIAA Journal, **51**, no. 4, 946-960. http://doi.org/10.2514/1.J051877

[26] Zimmermann, R. and Vendl, A. and Görtz, S. 2014. “Reduced-order modeling of steady flows subject to aerodynamic constraints.” AIAA Journal, **52**, no. 2, 255-266. http://doi.org/10.2514/1.J052208

[27] Fossati, M. 2015. “Evaluation of Aerodynamic Loads via Reduced-Order Methodology.” AIAA Journal, **53**, no. 8, 2389-2405. http://doi.org/10.2514/1.J053755

[28] Lumley, J.L. 1967. The Structure of Inhomogeneous Turbulent Flows. Atmospheric Turbulence and Radio Wave Propagation. Moscow, USSR: Publishing House Nauka.

[29] Sirovich, L. 1987. “Turbulence and the Dynamics of Coherent Structures. Part I: Coherent Structures.” Quarterly of Applied Mathematics, **45**, no. 3, 561-571. https://www.jstor.org/stable/43637457

[30] Masini, L. and Timme, S. and Peace, A. J. 2020. “Analysis of a civil aircraft wing transonic shock buffet experiment.” Journal of Fluid Mechanics, **884**, A1. http://doi.org/10.1017/jfm.2019.906

[31] Bisgard, J. 2021. Analysis and linear algebra: the singular value decomposition and applications. Providence, RI: American Mathematical Society.

[32] Kuleshov, V. and Fenner, N. and Ermon, S. 2018. “Accurate Uncertainties for Deep Learning Using Calibrated Regression.” arXiv, abs/1807.00263. http://arxiv.org/abs/1807.00263

[33] Scalia, G. and Grambow, C.A. and Pernici, B. and Li, Y. and Green, W.H. 2020. “Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction.” Journal of Chemical Information and Modeling, **60**, no. 6, 2697-2717. http://doi.org/10.1021/acs.jcim.9b00975

[34] Jacquin, L. and Molton, P. and Deck, S. and Maury, B. and Soulevant, D. 2009. “Experimental Study of Shock Oscillation over a Transonic Supercritical Profile.” AIAA Journal, **47**, no. 9, 1985-1994. http://doi.org/10.2514/1.30190

[35] Crouch, J. D. and Garbaruk, A. and Strelets, M. 2019. “Global instability in the onset of transonic-wing buffet.” Journal of Fluid Mechanics, **881**, 3-22. http://doi.org/10.1017/jfm.2019.748

[36] He, W. and Timme, S. 2021. “Triglobal infinite-wing shock-buffet study.” Journal of Fluid Mechanics, **925**, A27. http://doi.org/10.1017/jfm.2021.678

[37] Sartor, F. and Mettot, C. and Sipp, D. 2015. “Stability, receptivity, and sensitivity analyses of buffeting transonic flow over a profile.” AIAA Journal, **53**, no. 7, 1980-1993. http://doi.org/10.2514/1.J053588

[38] Halton, J. H. 1964. “Algorithm 247: Radical-inverse quasi-random point sequence.” Communications of the ACM, **7**, no. 12, 701-702. http://doi.org/10.1145/355588.365104

[39] Garnett, R.. 2023. Bayesian Optimization. Cambridge University Press.

[40] Kingma, D.P. and Ba, J. 2015. “Adam: A Method for Stochastic Optimization.” 3rd International Conference on Learning Representations, ICLR 2015. San Diego, CA: Conference Track Proceedings. http://arxiv.org/abs/1412.6980