58th 3AF International Conference on Applied Aerodynamics
27 — 29 March 2024, Orléans – France

AERO2024-43-TEJERO

Towards real-time CFD: Novel deep learning architecture for transonic wall-bounded flows

Fernando Tejero[1*], Sanjeeth Sureshbabu[1], Luca Boscagli[1] and David MacManus[1]
[1] *Centre for Propulsion Engineering, School of Aerospace, Transport and Manufacturing, Cranfield University, UK*
[*] *Corresponging author: f.tejero@cranfield.ac.uk*

ABSTRACT

Low order models can be used to accelerate engineering design processes. These surrogates should meet the conflicting requirements of large design space coverage, high accuracy and fast evaluation. Within the context of aerospace applications at transonic conditions, this can be challenging due to the associated non-linearity of the flow regime. This work develops a deep learning based method for flow-field prediction. It preserves the spatial resolution of the underlying data, which enables the resolution of the boundary layer. This is an advance relative to current state-of-the-art for transonic flows. The architecture is demonstrated for a complex problem of aero-engine nacelles. The prediction of the primitive flow variables is within a root mean square error of 6×10^{-5}. The model is used to extract the nacelle drag, and its accuracy is about 6.8% relative to CFD computations. The overall method is an enabling and fast preliminary design capability with self-consistent data for multidisciplinary design studies.

1. INTRODUCTION

Low order models can be used for preliminary design studies of new engineering systems. Within an industrial environment, it is important to rapidly iterate thorough novel concepts and ideas to be competitive and remain at the forefront of technology [1]. Recent developments in neural networks have opened a wide range of research disciplines over the last years [2]. For example, they have been used for the prediction of integral values [3, 4], classification metrics [5], inverse design methods [6], reinforcement learning for flow control [7], aerodynamic geometry definition [8] or Reynolds Average turbulence modelling [9]. Other important discipline is for flow-field prediction, which presents an alternative and effective approach for analyzing and optimizing aerodynamic shapes [2]. In particular, there is a growing interest in their application for digital systems [10].

Within the context of machine learning for flow-field prediction, research has focused on developing methods for on-surface prediction as well as off-surface prediction. Sabater *et al.* [11] explored three techniques for on-surface pressure distribution of 2D airfoil and the NASA CRM aircraft configuration: Gaussian Process, proper orthogonal decomposition combined with a thin-plate spline interpolation technique, and deep artificial neural networks. All three methods used Mach number (M) and angle of attack as input degrees of freedom (DoF). The database covered a range of flight Mach numbers up to M = 0.85. The study concluded that while all methods successfully predicted flow-field distributions in subsonic conditions, the deep neural network proved most effective in capturing the shock-wave location and its strength at transonic regimes. Immordino *et al.* [12] used a fully connected neural network for the prediction of the surface pressure and skin friction distribution along wings and airframes. The method was tested on 3D cases: the Benchmark Super Critical Wing (BSCW), the ONERA M6 wing, and the NASA CRM aircraft configuration. Independent low-order models were created for each test case, with Mach number (M) and angle of attack (AoA) as the input parameters, where M and AoA varied within 0.70-0.84 and 0°-5°, respectively. The mean absolute percentage error (MAPE) on the validation dataset was approximately of 1% on skin friction coefficient across the three cases. However, larger discrepancies were identified for the pressure coefficient prediction with a MAPE of around 20% for the CRM configuration. This large difference reduced to 0.54% and 2.50% for the BSCW and ONERA M6 cases, respectively. Tejero *et al.* [13] ex-

plored the capabilities of a fully connected artificial neural network for on-surface prediction in aero-engine nacelle applications. For this study, a notable distinction from Sabater *et al.* [11] and Immordino *et al.* [12] was in the use of geometric parameters as degrees of freedom, at fixed operating conditions. This approach aligns more closely with an industrial design environment, where the operating conditions are typically constant, and the goal is to identify an aerodynamic shape that meets specific requirements such as low drag or a bespoke pressure distribution. For this study, 3D non- axisymmetric nacelles were considered in which the geometries were defined with 20 design variables at a flight Mach number of M = 0.85. Overall, the proposed method predicted the isentropic Mach number distribution along the nacelle with an uncertainty of 0.02.

Whilst the applications for on-surface predictions have been demonstrated for relatively complex aerodynamic shapes, e.g. NASA CRM airframe, the research for off-surface prediction has been limited to simple shapes at subsonic and moderate transonic conditions. Duru *et al.* [14] developed a deep learning approach employing convolutional neural networks (CNN) to predict the flow field around airfoils at fixed M = 0.7. The architectures were trained to predict the static pressure and Mach number distributions. Although the study showcased the ability of CNNs for this design space with non-linear features such as shock-waves, the limitations related to near-wall resolution were acknowledged. Sekar *et al.* [15] integrated both fully connected and convolutional neural networks for airfoil applications at subsonic flow regimes with low Reynolds numbers below 2500. For this work, the CNN handled the geometry parameterization, while the fully connected network was used for flow estimation. The method had a prediction error of around 3% in both pressure and velocity fields. The main drawback of this architecture is the limited control in the definition of the aerodynamic shape, which is defined by the CNN model. To overcome these challenges, Kashefi *et al.* [16] introduced a point-cloud deep learning framework. This innovative approach utilizes the grid vertices, normally required in traditional Computational Fluid Dynamics (CFD) domains, as a point-cloud input within a PointNet architecture [17]. This enables the representation of the aerodynamic geometries with the same resolution as a CFD-based simulation, and eliminates the need for data interpolation into a Cartesian grid. The effectiveness of this method was successfully demonstrated in incompressible laminar flow scenarios across a diverse range of aerodynamic shapes. The root mean square error was typically within 0.1% for both velocity and pressure fields. It is important to note that training a PointNet model is computationally very expensive and it may become unfeasible for mesh sizes that are required for transonic applications. For example, the demonstration from Kashefi *et al.*[16] was based on a coarse mesh with approximately 1k cells count, which is orders of magnitude lower than the expected computational domains for transonic flows.

Deep learning is a viable approach for accelerating aerodynamic shape optimization during the initial phases of the design process. The development of accurate low-order models for off-surface flow-field prediction has the potential to change existing design practices within an industrial environment. However, current machine learning architectures are not suitable for complex aerodynamic shapes at transonic conditions due to limited resolution near the wall (CNN limitation) or large memory requirement during the training process (PointNet limitation). This study builds on previous work from the authors in which a novel architecture, named Point-Enhanced Convolutional Neural Network (PCNN), was developed by combining the benefits of CNN and PointNets models. This preserves the spatial resolution of the underlying CFD methods, conserves the spatial correlation in the domain (CNN feature) and reduces the memory requirements to train the models. The PCNN is used in compact aero-engine nacelle applications at transonic conditions. This is a challenging design problem due to the non-linear aerodynamics associated to this design space. As part of this work, the predicted flow-fields are used to derive nacelle drag. Relative to other applications such as airfoils [14, 15], nacelle drag poses several challenges as the pre-entry flow, wake region and force acting on the nacelle surface are required for its calculation. As such, the novelty of this work is in the assessment of the PCNN method for a complex transonic cases, and the derivation of integral parameters from the flow-field predictions. The contribution to knowledge is in the quantification of the method to predict nacelle drag and its comparison with current state-of-the-art low order models. The main benefit of the PCNN capability is that the low order model is based on self-consistent data. This is an enabling and fast preliminary design capability for multidisciplinary optimisation studies.

2. METHODOLOGY

2.1 Data generation

The underlying CFD data used in the training of the deep learning models was generated with a well-established method developed by Tejero *et al.* [18, 19, 20] for 2D aero-engine nacelles. A comprehensive description of the different modules was previously presented [18, 19], and therefore only a brief overview is provided in this paper.

The geometry of the different housing components of the aero-engine, i.e. intake, nacelle and exhaust system, are defined with the intuitive class shape transformation (iCST) method [19]. The nacelle shape is controlled with

seven design variables: L_{nac}, r_{hi}, r_{te}, f_{max}, r_{max}, r_{if} and β_{nac} (Figure 1). This geometry control has been successfully used in the past to assess the normalised effect of nacelle length (L_{nac}/r_{hi}) [21], exhaust compactness (r_{te}/r_{hi}) [19] and intake compactness (r_{hi}/r_{fan}) [22] on the nacelle drag characteristics. The computational grids follow a well-established multiblock structured strategy, with a cell count of approximately 30,000 cells. The first cell height is adjusted so satisfy y^+ of approximately 50. The numerical simulations are performed with a viscous and compressible steady Favre-averaged Navier-Stokes equations solver. A double-precision, implicit, density-based approach with the k-ω SST turbulence closure is used. The spatial discretisation is based on a second-order upwind method with a Green-Gauss node-based scheme. The dynamic viscosity is calculated with Sutherland's law and the thermal conductivity is derived from kinetic theory. The freestream is modeled at the pressure-far-field boundary condition, with imposed Mach number, static pressure, and static temperature. All the surface walls, i.e. nacelle, intake, spinner, bypass duct, and core duct surfaces, are represented as no-slip adiabatic walls. The fan-face is defined with a pressure-outlet boundary condition, wherein the mass flow is set according to the required mass flow capture ratio (MFCR). Inlets for the bypass and core ducts use pressure-inlet conditions with prescribed total pressure and temperature values. This numerical approach has been previously validated [19, 23] and used throughout a range of aero-engine nacelle studies [21, 22].

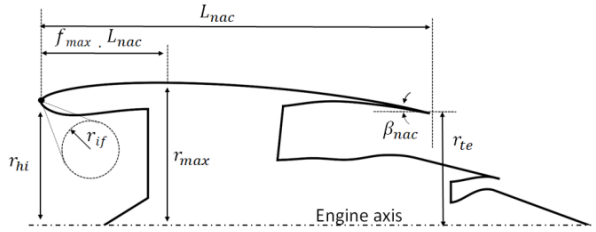


Figure 1: Aero-engine nacelle geometry definition

2.2 Deep learning architecture

The PCNN architecture comprises two main sections: a shared Multi-Layer Perceptron (MLP) and a Convolutional Neural Network (CNN), denoted as Section-I and Section-II, respectively (Figure 2). The input to the PCNN model is a two-dimensional array of dimensions $N \times d$, where N represents mesh points and d is the domain's dimensionality. Within Section-I, the model employs three blocks (Block-I-I, Block-II-I, and Block-III-I as illustrated in Figure 2), following a PointNet architecture. The first block (Block-I-I) initiates with two shared MLP layers (MLP1) each having 64 filters (N_f). Subsequently, a second set of shared MLP layers (MLP2) is used with increasing number of filters ($N_f = 64, 128, 1024$). The output dimension of Block-I-I is $N \times 1024$. Due to the discrepancy between the number of grid points (N) and 1024, direct use of a max-pooling layer is not feasbile [16]. The PCNN addresses this by transposing the output from Block-I-I, acknowledging that the preservation of permutation invariance, as in the original PointNet [16], is unnecessary for this work given the fully structured grid used during the data generation. This, however, highlights a limitation of the PCNN model for datasets generated with unstructured grid strategies. With permutation invariance not required, the output from Block-I-I is transposed and serves as input for Block-II-I (Figure 2). Max-pooling is applied along the rows, identifying dominant features in each of the N rows. This yields an output of $N \times 1$, which is then expanded to $N \times 1024$ by tiling the original tensor. The output from tiling is concatenated with the output from MLP1, preserving the geometric information. This concatenated output serves as input for Block-III-I, that uses four shared MLP layers (MLP3) with decreasing depths from $N_f = 1024$ to $N_f = 512, 256, 128$. The final layer's output is reshaped into $N_m \times N_n \times 128$ tensors, satisfying the condition $N = N_m \times N_n$. The selection of N_m and N_n values is guided by N_m/N_n proximity to 1 and $N_m > N_n$. In cases where N is a prime number, padding can be added to the original mesh points. Section-II of the PCNN architecture takes the output generated in Section-I (Figure 2) as input and consists of two blocks (Block-I-II and Block-II-II). Block-I-II involves two encoding (CNN1, CNN2) and two decoding layers (DeCNN1, DeCNN2), with CNN1 using 128 filters, reduced to 64 in CNN2. For the decoding layers, DeCCN1 and DeCNN2 are composed by 64 and 128 filters, respectively. Block-II-II comprises a single encoder and decoder (CNN3 and DeCNN3), structured as CNN1 and DeCNN2. The output of Block-II-II is a tensor of size $N_m \times N_n \times 128$, that is then reshaped to a 2D tensor with a column of size N and depth 128. Finally, a 1D CNN is used to transform the 2D tensor in an array of $N \times 1$ that contains the predicted field-data from the input point cloud of the PCNN model.

2.3 Thrust-drag bookkeeping

As part of this study, the analysis not only includes the quantification of the PCNN accuracy for predicting primitive flow variables, but other key consideration is the accuracy to predict nacelle drag (D_{nac} in Eq. 1). This integral metric is extracted with a well established thrust-drag bookkeeping extraction method (Figure 3a) [24].

$$D_{nac} = \phi_{pre} + \phi_{nac} + \phi_{post} \tag{1}$$

It is important to note that estimating drag in nacelle

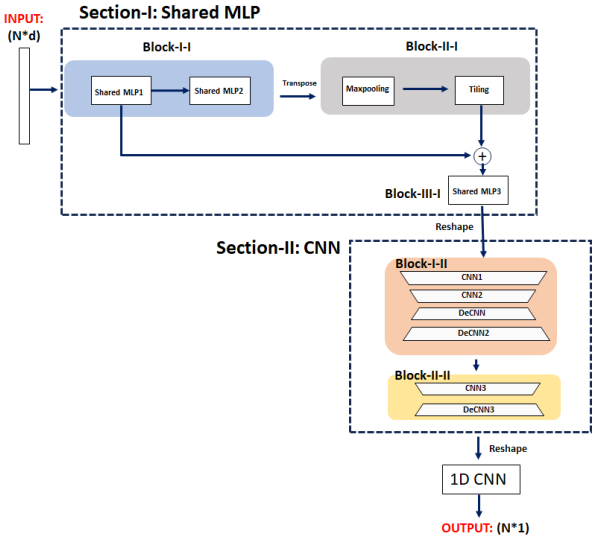


Figure 2: Schematic of the PCNN architecture

applications is more complex than other configurations such as airfoils. This is due to the requirements to accurately predict not only the forces exerted on the surfaces of interest (ϕ_{nac} in Eq. 1) but also in the pre-entry streamtube (ϕ_{pre} in Eq. 1) and post-exist streamtube (ϕ_{post} in Eq. 1) [24].

The calculation of the pre-entry streamtube is challenging because the identification of the stagnation point is needed. For this reason, this work uses a modified near-field method [19], which does not require this extraction. As the sum of forces over a closed surface is 0 in steady flow, the pre-entry streamtube can be expressed as a function of gauge forces (F_G) and the forces (pressure and viscous terms) exerted on the wall of the control volume (Eq. 2).

$$-\mathrm{F_{G0}} - \phi_{\mathrm{pre}} + \mathrm{F_{G2}} + \theta_{\text{intk-fore2}} = 0 \tag{2}$$

where the terms θ and ϕ are defined in Eq. 3, and F_G in Eq. 4.

From Eq. 2 the calculation of $\theta_{intk-fore2}$ requires the extraction of the stagnation point (Figure 3b), which is not trivial as mentioned above.

$$\theta, \phi = \int_S (p_s - p_\infty)\,\hat{n}\overrightarrow{dS} + \int_A \tau \overrightarrow{dA} \tag{3}$$

where p_s is the local static pressure, p_∞ refers to the static pressure at infinity, τ is the wall shear stress, $\hat{n}$ is the normal vector and the terms $\overrightarrow{dS}$ and $\overrightarrow{dA}$ refer to the normal and tangential surfaces.

$$F_G = \int_S (p - p_\infty)\,\hat{n}\overrightarrow{dS} + \int_S \vec{U}\rho|\vec{U}|\hat{n}\overrightarrow{dS} \tag{4}$$

where ρ is the density and $\vec{U}$ is the velocity vector.

The term ϕ_{nac} can be expressed as in Eq. 5, where the term ϕ_{cowl} refers to the pressure and viscous forces acting on the fancowl (Eq. 3) and the term $\phi_{intk-fore1}$ is the force in the intake from the highlight to the stagnation point (Figure 3b).

$$\phi_{nac} = \phi_{cowl} + \phi_{\text{intk-fore1}} \tag{5}$$

The summation of ϕ_{pre} and ϕ_{nac} (Eq. 6) can be performed by combining Eq. 2 and 5. This sum can be expressed in terms of gauges forces and surface forces exerted in geometries that are easy to extract.

$$\begin{aligned}\phi_{\mathrm{pre}} + \phi_{\mathrm{nac}} = -\mathrm{F_{G0}} + \mathrm{F_{G2}} + \theta_{\text{intk-fore2}} + \phi_{\mathrm{cowl}} + \phi_{\text{intk-fore1}} = \\ -\mathrm{F_{G0}} + \mathrm{F_{G2}} + \theta_{\text{intk-fore}} + \phi_{\mathrm{cowl}}\end{aligned} \tag{6}$$

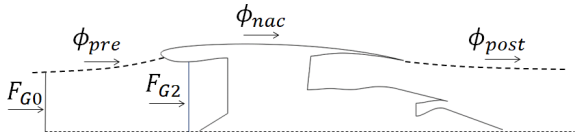
where $\theta_{\text{intk-fore}}$ is defined in Eq. 7.

$$\theta_{\text{intk-fore}} = \theta_{\text{intk-fore2}} + \phi_{\text{intk-fore1}} \tag{7}$$

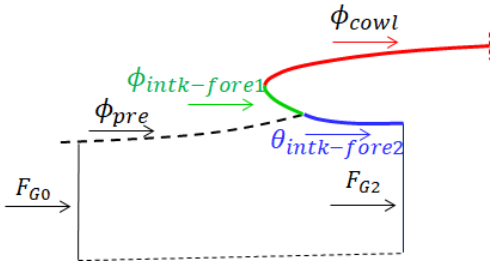
The post-exit streamtube (ϕ_{post} in Eq. 1) is calculated by pressure integration of the streamline from the engine trailing edge that divides the drag and thrust domains.

All the constituent terms that define nacelle drag are expressed in Eq. 8. They can be easily extracted from flow-field data as it does not require the identification of the stagnation point [19]. Table 1 summarizes the flow-field required to calculate each term (Eq. 8). Overall, the extraction of nacelle drag requires the flow properties of density (ρ), axial velocity (u_a), radial velocity (u_r), static pressure (P_s) and x-wall shear stress (τ_x).

$$D_{nac} = -\mathrm{F_{G0}} + \mathrm{F_{G2}} + \theta_{\text{intk-fore}} + \phi_{\mathrm{cowl}} + \phi_{\mathrm{post}} \tag{8}$$



(a) Overview



(b) Zoom in the nacelle leading edge

Figure 3: Thrust drag bookkeeping

Term	PCNN flow-field
F_{G0}	density, axial velocity
F_{G2}	density, axial velocity, static pressure
$\theta_{intk-fore}$	static pressure and shear stress
ϕ_{cowl}	static pressure and x-wall shear stress
ϕ_{post}	axial velocity, radial velocity, static pressure

Table 1: Flow variables required to calculate the different terms that encompass nacelle drag (D_{nac})

3. RESULTS

This work is based on a short nacelle style with a fixed $L_{nac}/r_{hi} = 3.1$. For long-range applications with a flight Mach number of M = 0.85, this is a challenging design space [19, 21] to test the accuracy of the PCNN deep learning method. The Reynolds number based on the highlight radius is $Re_{hi} \approx 12 \times 10^6$. A design space exploration, based on a latin hypercube sampling, was performed to generate nacelle shapes by changing the design variables of f_{max}, r_{max}, r_{if} and β_{nac} (Figure 1). Overall, a total of 800 aero-engine nacelles were computed with the CFD method described in Section 2.1. Across this design space, there are a range of different transonic flow aerodynamics which highlight the challenges of building surrogate models for flow-field prediction. Figure 4 presents three aero-engine nacelles (A1, A2 and A3) with different flow characteristics. Whilst the configuration A1 has a single normal shock, the other two nacelles have a double shock topology [22]. For A2, the flow accelerates around the highlight to terminate in a normal shock located at the forebody, and the flow reaccelerates to induce a second shock at the afterbody. For this configuration, there is shock-induced flow separation which starts at x/L_{nac} = 0.66 and the flow reattaches at x/L_{nac} = 0.82. The design A3 shows a larger initial acceleration around the nacelle lip than A2, which results in a shock-wave close to the leading edge at x/L_{nac} = 0.06. Then the flow reaccelerates to terminate in a strong shock wave at x/L_{nac} = 0.49 that induces boundary layer separation up to the nacelle trailing edge. Conversely, the design A1 does not present flow separation along the nacelle. This variation in the nacelle transonic aerodynamics result in notable changes of the associated nacelle drag response. For example, relative to A1, the nacelle drag for the configurations A2 and A3 increases by a factor of approximately 4 and 10, respectively.

The database of 800 aero-engine nacelles was split with a ratio of 80% and 20% for training and validation of the PCNN models, respectively. Independent architectures for the flow variables of axial velocity, radial velocity, static pressure and density were generated. These variables are required as part of the nacelle drag extraction method described in Section 2.3 (Table 1). The

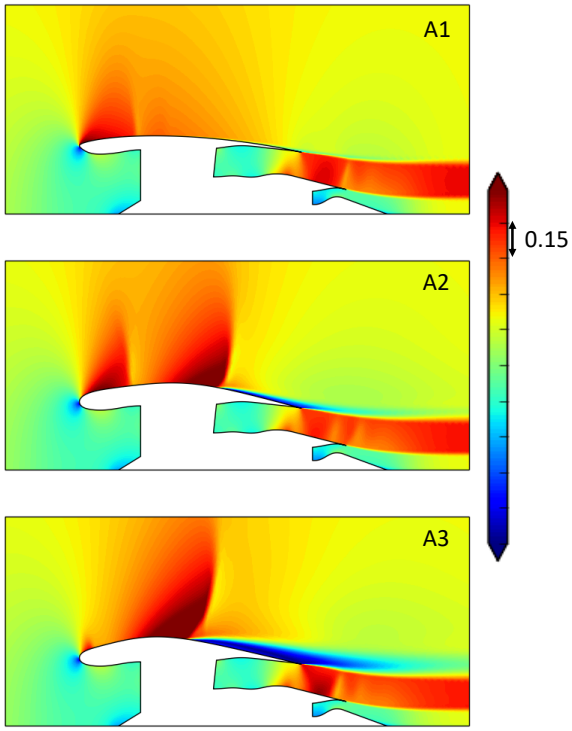


Figure 4: Mach number distribution for three nacelles (A1, A2, A3) across the design space

training process was based on minimising the root mean square error (σ_f in Eq. 9) across the domain.

$$\sigma_f = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f^{CFD} - f^{ANN})^2} \tag{9}$$

where f refers to the flow variable that the PCNN model is trained with, i.e. density (ρ), axial velocity (u_a), radial velocity (u_r), static pressure (P_s) and x-wall shear stress (τ_x).

This work assesses the viability of deep learning architectures for flow-field prediction and their accuracy to extract integral values. It is important to note that the proposed approach, based on training independent models, does not guarantee that the primitive variables fulfill the underlying Navier-Stokes equations. This requires to train the models simultaneously using a physics-informed loss-function [25]. However, this is not within the scope of this work as there is an initial need to quantify the potential accuracy of the PCNN deep learning architecture for complex transonic applications.

All the PCNN models had the architecture described in Section 2.2. Overall, the flow-fields of approximately 640 nacelles were used during the training process and the validation was performed with the remaining 160 nacelles of the database. The models to predict the properties of u_a, u_r, P_s and ρ used the 30,000 cells points of the

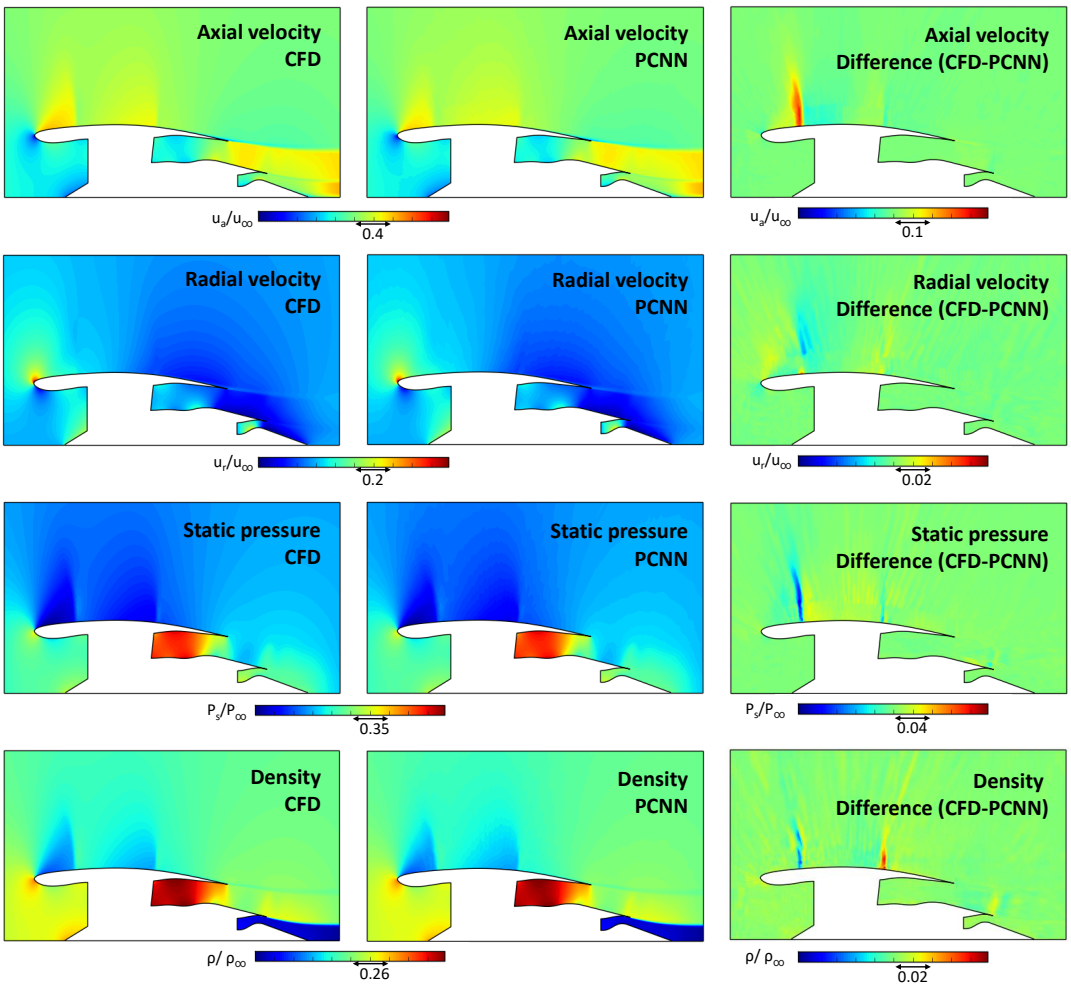


Figure 5: Comparison between CFD and PCNN of flow properties for Design B1

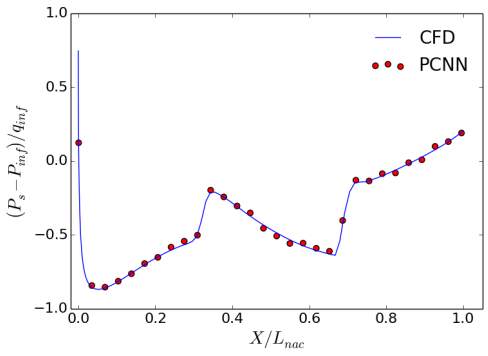
computational domain from each nacelle. As such, each PCNN model is trained with 19.2 million inputs (30,000 $\times$ 640). However, the flow variable τ_{wall} was only trained with the 190 cells that discretise the nacelle wall. Therefore, this models is composed by approximately 120,000 inputs (190 $\times$ 640). Table 2 summarizes the accuracy of each independent PCNN architecture.

Flow property	Training loss	Validation loss
u_a	4.4×10^{-6}	6.5×10^{-5}
u_r	2.4×10^{-6}	4.4×10^{-6}
P_s	9.3×10^{-7}	7.6×10^{-6}
ρ	5.5×10^{-6}	6.1×10^{-5}
τ_{wall}	1.2×10^{-6}	4.5×10^{-6}

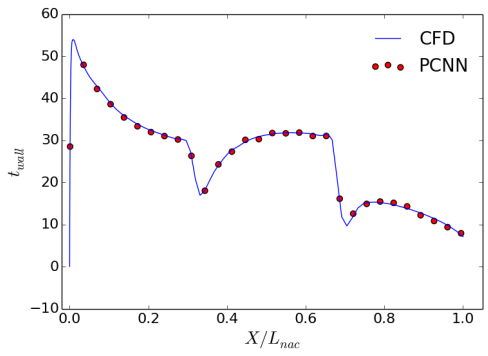
Table 2: Normalised mean square error of the independent PCNN models required to calculate nacelle drag

Figure 5 provides an insight of the PCNN capabilities to predict u_a, u_r, P_s and ρ across the domain. It shows an aero-engine nacelle (design B1) from the validation dataset. This is a challenging configuration due to the formation of two supersonic regions along the nacelle as well as the complex flow aerodynamics in the exhaust system with compression and expansion waves [26]. The axial velocity is predicted with a root mean square error relative to the freestream velocity of 1.3%. The largest difference appears in the region of the first shock-wave and this is due to a small change in the predicted shock location (Figure 5). Due to the large flow gradients in that region, small difference in shock location result in large differences between the CFD and PCNN results. It is worth highlighting that the agreement between both methods (CFD and PCNN) in the axial velocity of the exhaust flow is excellent. As the exhaust shape was not modified during the generation of the database, the changes in the associated transonic aerodynamics across the design space are more benign that the nacelle aerodynamics. For this case, the predictions of the radial velocity is very good with a root mean square

error relative to freestream velocity of 0.41%. A similar level of agreement was achieved for the static pressure and density fields with root mean square errors relative to freestream conditions of 0.60% and 0.43%, respectively. For these two flow variables, i.e. P_s and ρ, the largest difference is also at the shock locations due to small differences in the prediction of their positions (Figure 5). Whilst these 4 models (PCNN architectures for u_a, u_r, P_s and ρ) predict the flow in the full domain, the properties along the surfaces of the housing components can be extracted. Specifically, this is required for the P_s model to extract the pressure terms of $\theta_{\text{intk-fore}}$ and ϕ_{cowl} (Eq. 6) as part of the nacelle drag extraction method (Section 2.3). The model for the τ_x directly provides the x-wall shear stress along the wall of the nacelle. Figure 6 compares the normalised pressure distribution (P_s) with the dynamic head (q_{inf}) and the x-wall shear stress (τ_x) along B1 between PCNN and CFD. For both flow properties, i.e. P_s (Figure 6a) and τ_x (Figure 6b), there is a good agreement between the methods.



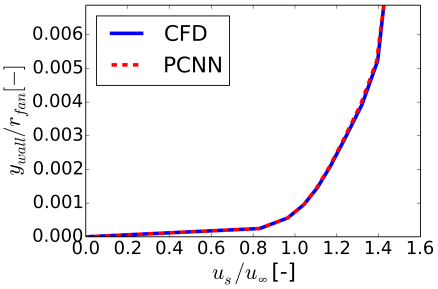
(a) Pressure coefficient



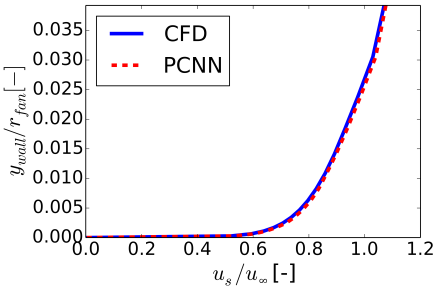
(b) X-wall shear stress

Figure 6: PCNN and CFD comparison along the nacelle surface for design B1

An advantage of this PCNN approach relative to state-of-the-art CNN architectures is its capability to retain the spatial resolution of the underlying data. This is important for transonic wall-bounded flows where may be required accurate predictions of the boundary layer characteristics. To provide an insight of this capability, the boundary layer for the design B1 at two axial locations (X/L_{nac} =0.12 and 0.89) is presented in Figure 7. For this nacelle, the boundary layer is fully attached along the fancowl (Figure 6b). The agreement between PCNN and CFD is excellent. The boundary layer characteristics in terms of the integral parameters of incompressible displacement thickness (δ_i^*), momentum thickness (θ_i^*) and shape factor (H_i^*) [27] were extracted. For the first axial location at X/L_{nac} = 0.12 (Figure 7a), the PCNN method predicted the δ_i^*, θ_i^* and H_i^* with a difference of 1.4%, 1.7% and 0.28% relative to CFD, respectively. This difference slightly increased to 2.0%, 1.8% and 0.3% at the second location with X/L_{nac} = 0.89 (Figure 7b). This is an excellent agreement as it is within the uncertainty of measurements [27].



(a) X/L_{nac} = 0.12



(b) X/L_{nac} = 0.89

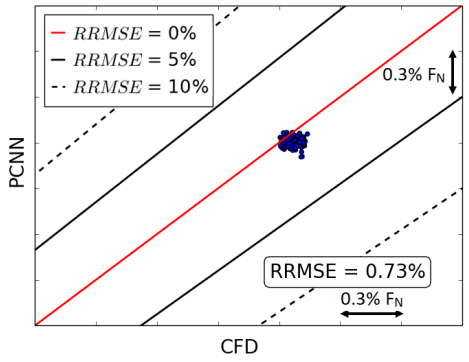
Figure 7: PCNN and CFD comparison of boundary layer profiles for design B1, where u_s is the streamwise velocity and u_∞ is the free-stream velocity

A key aspect of this work is to quantify the accuracy of the nacelle drag prediction through the integration of flow properties from the PCNN models. This is important for a preliminary design environment in which different considerations within a multidisciplinary process are considered. It enables the use of self-consistent models and

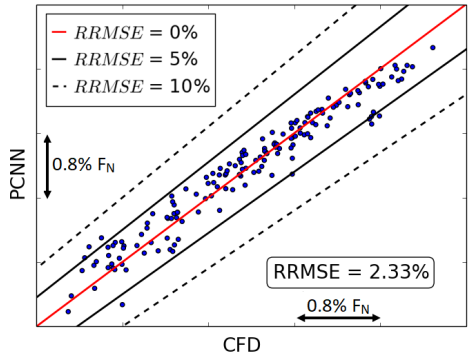
avoids conflicting conclusions due to the deployment of different low order models. For example, this PCNN approach could be used by the aerodynamic team for aerodynamic, acoustic or aeroloads assessments. A comparison between CFD and PCNN for the different constituent terms of the nacelle drag (Eq. 8) is presented in Figure 8. The difference in gauge force between the plane-2 (Figure 3a) and the upstream infinity planes, i.e. $F_{G2} - F_{G0}$, is predicted by the PCNN model with a relative root mean square error (RRMSE in Eq. 10) of 0.73% relative to CFD. It is important to highlight that the changes across the validation dataset are very small with a variation of approximately 0.1% of the aero-engine net thrust (F_N). The gauge force at the upstream infinity plane (F_{G0}) does not change across the design space because the massflow through the fan was kept constant in the design space exploration. Therefore, the observed variation of 0.1% in F_N is caused by small changes on F_{G2}. More differences between the PCNN and CFD arise for the post-exist streamtube (ϕ_{post}) with a RRMSE of 2.33% (Figure 8b). Across the space, this metric varies by approximately 3.3% of F_N. As the nacelle shapes were modified while compiling the database, there are different pressure fields on the nacelle trailing edge which result in changes in the post-exist streamtube shape and its static pressure along it. There is also a very good agreement in the prediction of $\theta_{intk-fore}$ with a relative root mean square error of 0.82% with respect to numerical simulations (Figure 8c). The changes in this term across the design space are modest with a variation of approximately 1.8% of the net thrust. This is caused by the modifications to the intake geometry throughout the design space exploration to ensure continuity in the initial radius of curvature between the intake and nacelle (r_{if} in Figure 1). The nacelle cowl force (ϕ_{cowl}) has a RRMSE = 1.91%. This is the most challenging term due to the range of transonic aerodynamics that are encountered in the database (Figure 5). Across the design space, ϕ_{cowl} changes by about 30% of the net thrust.

$$RRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{f^{CFD} - f^{ANN}}{f^{CFD}}\right)^2} \qquad (10)$$

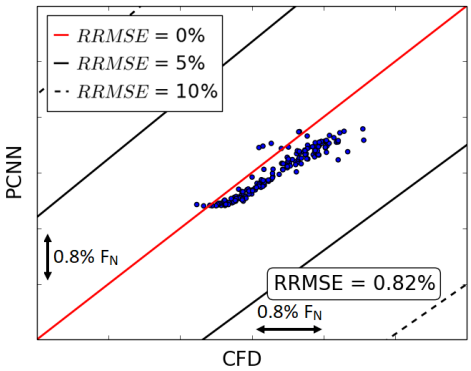
All the terms in Figure 8 were combined to calculate the aero-engine nacelle drag (D_{nac}) as defined in Eq. 8. Overall, the root mean square error of D_{nac} extracted from the PCNN models is 6.8% relative to CFD. Within the context of preliminary nacelle design, ESDU introduced an approach to predict nacelle drag for 2D axisymmetric configurations at subsonic free-stream Mach numbers [28]. This method exhibited a nacelle drag prediction error of 20% for wave drag transonic conditions. As such, the current PCNN capability outperforms ESDU for preliminary design. However, it is important to note that recent studies with artificial neural networks have demonstrated that a single regression model to predict the na-



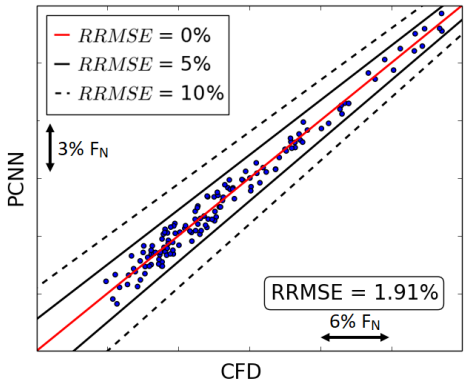
(a) $F_{G2} - F_{G0}$



(b) ϕ_{post}



(c) $\theta_{intk-fore}$



(d) ϕ_{cowl}

Figure 8: Cross-validation of the PCNN and CFD models for different terms that compose nacelle drag

celle drag can achieve an accuracy up to 2.4% [18]. As such, to benchmark the capabilities of the PCNN model, an ANN was built following the guidelines and recommendations from Tejero *et al.* [18]. For this purpose, fully connected layers were used in which the input comprises 4 neurons (f_{max}, r_{max}, r_{if} and β_{nac}) and the output has one neuron (D_{nac}). The architecture was composed of 2 hidden layers with 128 neurons per hidden layers, and the ReLu activation function. The gradient descent ADAM optimised was also used in the training of ANN network, where the function loss was based on the relative root mean square error of the nacelle drag. For this complex design space, the ANN had an uncertainty of 2.5% (Figure 9). As expected, the ANN method outperforms the PCNN approach. Nevertheless, the PCNN deep learning architecture also includes the flow-field in the domain which enable future multidisciplinary trade-off studies. This is a key advantage for preliminary design work.

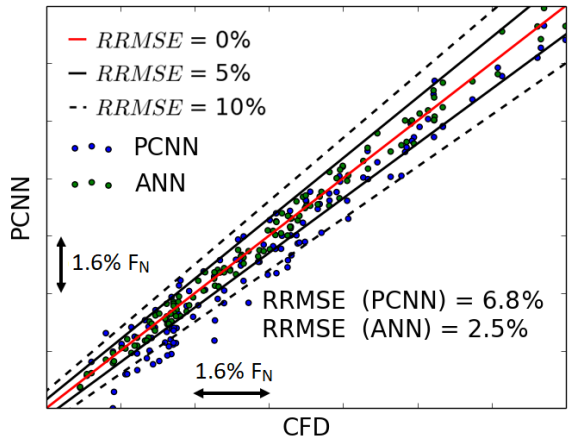


Figure 9: Cross-validation of low order models and CFD for nacelle drag

4. CONCLUSIONS

The capabilities of the Point-Enhanced Convolutional Neural Network (PCNN) for flow-field prediction of transonic aero-engine nacelles has been investigated. The method has been successfully demonstrated for this complex design space, where models to predict axial velocity, radial velocity, density, static pressure and X-wall shear stress have been trained. Overall, the accuracy to predict these flow properties was within a root mean square error of 6×10^{-5}. The models were combined to extract the nacelle drag by using a well-established thrust-drag bookkeeping method. The contribution to knowledge of this work has been in quantifying the accuracy of nacelle drag prediction, and its comparison with current state-of-the-art surrogate models. It is demonstrated that the nacelle drag can be predicted with an uncertainty of 6.8% relative to CFD across a relatively large design space. As expected, this is larger than the error from a single artificial neural network to predict the nacelle drag with an error of 2.5%. However, the PCNN architecture also contains the associated flow-fields, making this approach a preliminary method with self-consistent data that can be deployed within a multidisciplinary framework. The overall PCNN method is an enabling and fast capability for preliminary trade-off studies.

ACKNOWLEDGMENTS

This work was partially funded by the Innovate UK Fanfare project. This included the generation of the CFD database for 2D aero-engine nacelles. This work has also received funding from the Clean Sky-2 Joint Undertaking (JU) under grant agreement number 101007598 – ODIN Project. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union. This finding included the development of the PCNN methodology and the analysis of the data.

DATA AVAILABILITY STATEMENT

Due to commercial confidentiality agreements the supporting data are not available

REFERENCES

[1] L. Martinez and T. Nguyen. Challenges and Opportunities of Simulation-Driven Industrial Design. *Engineering Design Journal*, 22(4):289–306, 2017.

[2] J. Li, X. Du, and J.R.R.A. Martins. Machine Learning in Aerodynamic Shape Optimization. *Progress in Aerospace Sciences*, 134(100849), 2022.

[3] G.C. Tao, W. Wang, Z.T. Ye, Y.N. Wang, J.Q. Luo, and J.H. Cui. Multi-Fidelity and Multi-Objective Aerodynamic Short Nacelle Shape Optimisation Under Different Flight Conditions. *The Aeronautical Journal*, page 1–30, 2023.

[4] F. Sanchez, D. MacManus, J. Hueso, F. Tejero, and C. Sheaf. Optimization of Installed Compact and Robust Nacelles Using Surrogate Models. In *33rd Congress of the International Council of the Aeronautical Sciences*, 2022.

[5] P. H. Niknam, B. Mokhtarani, and H. R. Mortaheb. Prediction of shockwave location in supersonic nozzle separation using self-organizing map classification and artificial neural network modeling. *Journal of Natural Gas Science and Engineering*, 34:917–924, 2016.

[6] C. Wang, S. Wang, L. Wang, C. Cao, G. Sun, C. Li, and Y. Yang. Framework of Nacelle Inverse Design Method Based on Improved Generative Adversarial Networks. *Aerospace Science and Technology*, 121:107365, 2022.

[7] M. Shams and A.H. Elsheikh. Gym-preCICE: Reinforcement Learning Environments for Active Flow Control. *SoftwareX*, 23:101446, 2023.

[8] J. Li, M. Zhang, J. R. R. A. Martins, and C. Shu. Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering. *AIAA Journal*, 58(10):4243–4259, 2020.

[9] R.D. Sanhueza, S.H.H.J. Smit, J.W.R. Peeters, and R. Pecnik. Machine Learning for RANS Turbulence Modeling of Variable Property Flows. *Computers AMD Fluids*, 255:105835, 2023.

[10] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A.Y.C. Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58:3–21, 2021.

[11] C. Sabater, P. Sturmer, and P. Bekemeyer. Fast Predictions of Aircraft Aerodynamics Using Deep-Learning Techniques. *AIAA Journal*, 60(9):5249–5261, 2022.

[12] G. Immordino, A. Da Ronch, and M. Righi. Deep–Learning Framework for Aircraft Aerodynamics Prediction. In *AIAA AVIATION 2023 Forum, AIAA Paper 2023-3846*, 2023.

[13] F. Tejero, D. MacManus, J. Matesanz-Garcia, L. Boscagli, J. Hueso-Rebassa, and C. Sheaf. Deep-Learning for Flow-Field Prediction of 3D Non-Axisymmetric Aero-Engine Nacelles. In *AIAA AVIATION 2023 Forum, AIAA paper 2023-3307*, 2023.

[14] C. Duru, H. Alemdar, and O. U. Baran. A Deep Learning Approach for the Transonic Flow Field Predictions Around Airfoils. *Computers and Fluids*, 236:105312, 2022.

[15] V. Sekar, Q. Jiang, C. Shu, and B. C. Khoo. Fast Flow Field Prediction Over Airfoils Using Deep Learning Approach. *Physics of Fluids*, 31(5):057103, 2019.

[16] A. Kashefi, D. Rempe, and L. J. Guibas. A Point-Cloud Deep Learning Framework for Prediction of Fluid Flow Fields on Irregular Geometries. *Physics of Fluids*, 33(2):027104, 2021.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[18] F. Tejero, D. MacManus, F. Sanchez-Moreno, and C. Sheaf. Neural Network-Based Multi-Point, Multi-Objective Optimisation for Transonic Applications. *Aerospace Science and Technology*, 136:108208, 2023.

[19] F. Tejero, M. Robinson, D. MacManus, and C. Sheaf. Multi-Objective Optimization of Short Nacelles for High Bypass Ratio Engines. *Aerospace Science and Technology*, 91:410–421, 2019.

[20] F. Tejero, D. MacManus, A. Heidebrecht, and C. Sheaf. Artificial Neural Network for Preliminary Design and Optimisation of Civil Aero-Engine Nacelles . *The Aeronautical Journal*, 128:2261–2280, 2024.

[21] F. Tejero, D. MacManus, I. Goulos, and C. Sheaf. Propulsion Integration Study of Civil Aero-Engine Nacelles. *The Aeronautical Journal*, page 1–15, 2023.

[22] F. Tejero, D. MacManus, J. Matesanz-Garcia, and C. Sheaf. Towards the Design and Optimisation of Future Compact Aero-Engines: Intake/Fancowl Trade-Off Investigation. *International Journal of Numerical Methods for Heat and Fluid Flow*, 33:1319–1335, 2023.

[23] L. Boscagli, D. MacManus, F. Tejero, K. Sabnis, H. Babinsky, and C. Sheaf. Characteristics of Shock-Induced Boundary Layer Separation on Nacelles Under Windmilling Diversion Condition. *AIAA Journal*, In press, 2023.

[24] AGARDograph AG-237. Guide to In-Flight Thrust Measurement of Turbojets and Fan Engines. Technical report, AGARD Report, 1979.

[25] L.H. Queiroz, F.P. Santos, J.P. Oliveira, and M.B. Souza. Physics-Informed Deep Learning to Predict Flow Fields in Cyclone Separators. *Digital Chemical Engineering*, 1:100002, 2021.

[26] I. Goulos, T. Stankowski, J. Otter, D. MacManus, N. Grech, and C. Sheaf. Aerodynamic Design of Separate-Jet Exhausts for Future Civil Aeroengines—Part I: Parametric Geometry Definition and Computational Fluid Dynamics Approach. *J. Eng. Gas Turbines Power*, 138:081201, 2016.

[27] A. Coschignano, N. Atkins, H. Babinsky, and J. Serna. Effect of Reynolds Number on a Normal Shock Wave-Transitional Boundary-Layer Interaction over a Curved Surface. *Experiments in Fluids*, 60:1–12, 2019.

[28] ESDU. Drag of Axisymmetric Cowls at Zero Incidence for Subsonic Mach Numbers. Technical Report 81024, ESDU, December 1994.

https://dspace.org/

Towards real-time CFD: novel deep learning architecture for transonic wall-bounded flows

Tejero, Fernando; Sureshbabu, Sanjeeth; Boscagli, Luca; MacManus, David G.

2024-03-27

https://dspace.lib.cranfield.ac.uk/handle/1826/23268

Tejero F, Sureshbabu S, Boscagli L, MacManus D. (2024) Towards real-time CFD: novel deep learning architecture for transonic wall-bounded flows. In: 58th 3AF International Conference on Applied Aerodynamics, 27-29 Mar 2024, Orleans, France

Downloaded from a DSpace repository. https://dspace.org/