A Flow Field Reconstruction Method Using Artificial Neural Network

Jian Yu*
Beihang University, Beijing 100191, China

Jan S Hesthaven †
École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland

Multi-dimensional aerodynamic database development has become more and more important for the design, control and guidance of modern aircrafts. In order to relieve the curse of the dimensionality, we propose a novel flow field reconstruction method based on artificial neural network. The idea is to design a simplified problem which is related to the target problem. Then the map from the simplified problem to the target problem is built using an artificial neural network. Finally, the target problem can be predicted efficiently through solving the simplified problem instead. Examples of the efficiency of this approach include two-dimensional viscous nozzle flows, the inviscid M6 wing flow, and a viscous hypersonic flow of a complex configuration to evaluate the performance of the proposed method. With artificial neural network of moderate complexity, the solution of the target problem can be generated with good accuracy. Among other observations, we find that shocks can be predicted well with sharp resolution.

Nomenclature

ρ	=	density
u, v, w	=	velocity in the x,y,z direction
p	=	pressure
e	=	internal energy
T	=	temperature
R	=	gas constant
μ	=	viscosity coefficient
Pr	=	Prandtl number
$\mathbb{R}$	=	real space

*Assistant Professor, School of Aeronautic Science and Engineering, yuj@buaa.edu.cn.
†Professor, Chair of Computational Mathematics and Simulation Schience, Jan.Hesthaven@epfl.ch.

χ = parameter vector

$\boldsymbol{A}$ = snapshot matrix

N = number of neurons

Superscripts

i = input layer

o = output layer

$h1, h2, h3$ = first, second and third hidden layer respectively

sp = simplified problem

I. Introduction

Nowadays, multi-dimensional aerodynamic databases are extensively used for design optimization, flight control, etc. The term *Multi-dimensional* denotes the multiple parameters such as flight height, Mach number, angle of attack, and etc. However, the amount of data required often grows exponentially with the dimensionality, known as the curse of dimensionality[1]. To reduce the cost, a common method is to generate a limited number of samples using either experimental or numerical approaches, and then extend the high-fidelity samples with an efficient approach. The simplest way for such extension is plain interpolation, which may be inaccurate when the samples are sparse. Bui-Thanh et al.[2, 3] proposed a method combining interpolation with a modified proper orthogonal decomposition(POD) called gappy POD[4], i.e. gappy POD is first conducted on the high-fidelity samples to generate a number of modes, which are then interpolated for the sought parameters. Alonso et al. [5] proposed to compute the interpolated modes using a reduced-order modeling of the governing equation. Mifsud et al. [6] proposed a variable-fidelity aerodynamic model which employs low-fidelity samples for POD to reduce the overall cost. An attractive feature of such methods is that the POD modes contain the global information of the high-fidelity samples, which ensures accuracy. However, when flow compressibility generates shocks(e.g. transonic airfoil flow), the position and/or the structure of the shock can vary significantly with the parameters, which leads to a stair-cased array of jumps rather than a single jump due to the linear combination of the POD modes. Therefore, approaches specially designed for shocks [1, 7] have been developed.

In this work, we propose an alternative approach to deal with the challenge of extending the use of the high-fidelity samples. Instead of tackling the target problem directly, we define a simplified problem, which seeks to mimic the target problem at a certain level. The samples for the simplified problem correspond to those for the target problem in a one-to-one correspondence manner. We then seek to build a map between the simplified and target problems using an artificial neural network(ANN). The concept of artificial neural network originates from biological neural network and is composed of a number of simple and highly interconnected processing elements, known as neurons[8]. An ANN is

generally nonlinear and applicable in various classification and regression problems, and is able to approximate any function with moderate complexity[9]. An ANN is used here as a map function between vectors. For a specific problem, we first train a specified network with samples of the simplified and target problems. For new parameters, the target problem is predicted with the network by solving the corresponding simplified problem, which in general is chosen to be significantly cheaper.

In this paper, we first describe the methods, including the governing equations, POD and the ANN. We subsequently propose our flow reconstruction method based on POD and the ANN, and demonstrate the method for several typical 2D and 3D cases, after which some conclusions are drawn.

II. Method

A. Governing equation

In this section, we only describe the governing equations for the test cases conducted in this paper. Note that the flow field reconstruction does not rely on the governing equation, and is not limited to the problems presented here. The conservation law is generally given by

$$\frac{\partial \boldsymbol{q}}{\partial t} + \nabla \cdot \boldsymbol{f} - \nabla \cdot \boldsymbol{g} = 0, \tag{1}$$

where $\boldsymbol{q}$ is the conserved variables, $\boldsymbol{f}$ is the convective flux, and $\boldsymbol{g}$ is the viscous flux. For the two-dimensional Navier-Stokes system, we have

$$\boldsymbol{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \qquad \boldsymbol{f} = \boldsymbol{f}^x \vec{\boldsymbol{i}} + \boldsymbol{f}^y \vec{\boldsymbol{j}}, \qquad \boldsymbol{g} = \boldsymbol{g}^x \vec{\boldsymbol{i}} + \boldsymbol{g}^y \vec{\boldsymbol{j}}. \tag{2}$$

The convective fluxes are given by

$$\boldsymbol{f}^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho e + p) \end{pmatrix}, \qquad \boldsymbol{f}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho e + p) \end{pmatrix}. \tag{3}$$

For a perfect gas, $p = \rho RT$, and $e = \frac{p}{(\gamma-1)\rho} + \frac{1}{2}(u^2 + v^2)$.

The viscous fluxes are given by

$$\boldsymbol{g}^x = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + b_x \end{pmatrix}, \qquad \boldsymbol{g}^y = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + b_y \end{pmatrix}, \tag{4}$$

where

$$\tau_{xx} = \frac{4}{3}\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\frac{\partial v}{\partial y}, \quad \tau_{yy} = \frac{4}{3}\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\frac{\partial u}{\partial x}, \quad \tau_{xy} = \mu\frac{\partial u}{\partial y} + \mu\frac{\partial v}{\partial x}, \tag{5}$$

and

$$b_x = \kappa\frac{\partial T}{\partial x}, \quad b_y = \kappa\frac{\partial T}{\partial y}, \quad \kappa = \frac{\mu\gamma R}{(\gamma-1)Pr}. \tag{6}$$

The dynamic viscosity μ is determined with

$$\frac{\mu}{\mu_{\text{ref}}} = \left(\frac{T}{T_{\text{ref}}}\right)^{1.5}\left(\frac{T_{\text{ref}} + T_s}{T + T_s}\right), \tag{7}$$

where $T_{\text{ref}} = 273.15\text{K}$, $\mu_{\text{ref}} = 1.716 \times 10^{-5}\text{Pa}\cdot\text{s}$, $T_s = 110.4\text{K}$. The three-dimensional equations are obtained by extension.

Furthermore, to test the proposed method, we will also consider the one-dimensional nozzle flow, which is described by

$$\frac{\partial \tilde{\boldsymbol{q}}}{\partial t} + \frac{\partial \tilde{\boldsymbol{f}}}{\partial x} - \frac{\partial \tilde{\boldsymbol{g}}}{\partial x} = \boldsymbol{J}, \tag{8}$$

where

$$\tilde{\boldsymbol{q}} = \begin{pmatrix} \rho A \\ \rho u A \\ \rho e A \end{pmatrix}, \qquad \tilde{\boldsymbol{f}} = \begin{pmatrix} \rho u A \\ (\rho u^2 + p)A \\ (\rho e + p)uA \end{pmatrix}, \qquad \tilde{\boldsymbol{g}} = \begin{pmatrix} 0 \\ \tau_{xx} A \\ (u\tau_{xx} + b_x)A \end{pmatrix}, \qquad \boldsymbol{J} = \begin{pmatrix} 0 \\ p\frac{\partial A}{\partial x} \\ 0 \end{pmatrix}. \tag{9}$$

Here $A = A(x)$ is the cross sectional area function.

B. Flow field reconstruction

The proposed flow field reconstruction approach is based on the assumption that a simplified problem, corresponding to the target problem, is first defined and then a map between the simplified and target problems is established using an ANN. Therefore, POD and ANNs are first introduced in the following, after which the flow field reconstruction method proposed is presented in more details.

1. Proper orthogonal decomposition

Given a parameter matrix $\boldsymbol{\Xi} = \{\boldsymbol{\chi}^1, \ldots, \boldsymbol{\chi}^{M_s}\}$, we have the corresponding set of high-fidelity solutions, referenced to *snapshots*, denoted as

$$\boldsymbol{A} = [\boldsymbol{S}_1 | \ldots | \boldsymbol{S}_{M_s}] \in \mathbb{R}^{M_h \times M_s}. \tag{10}$$

Here $\boldsymbol{S}_m$ denotes the solution vector of the mth snapshot, M_s is the number of the snapshots, and M_h is the size of each solution vector. Taking the one-dimensional problem as an example, the solution vector is

$$\boldsymbol{S}_m = \begin{bmatrix} \boldsymbol{\rho}_m \\ \boldsymbol{u}_m \\ \boldsymbol{p}_m \\ \boldsymbol{Ma}_m \end{bmatrix}, \tag{11}$$

where $\boldsymbol{\rho}_m$, $\boldsymbol{u}_m$, $\boldsymbol{p}_m$ and $\boldsymbol{Ma}_m$ are the solution vectors for density, velocity, pressure, and Mach number, respectively. Note that for the solution vectors, one is not limited to the above three variables, i.e., one could remove one or more fields and/or add new fields depending on the variables of interest.

Next, we perform the *thin* singular value decomposition (SVD)[10] of the snapshot matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{r} \sigma_i \boldsymbol{\zeta}_i \boldsymbol{\psi}_i^T, \tag{12}$$

where

$$\boldsymbol{U} = [\boldsymbol{\zeta}_1 | \ldots | \boldsymbol{\zeta}_r] \in \mathbb{R}^{M_h \times r}, \quad \boldsymbol{V} = [\boldsymbol{\psi}_1 | \ldots | \boldsymbol{\psi}_r] \in \mathbb{R}^{M_s \times r}, \tag{13}$$

$$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r}, \quad r \leq \min(M_h, M_s). \tag{14}$$

If we rewrite $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}$, $\boldsymbol{\Lambda}$ is the coefficient matrix, which can be obtained with $\boldsymbol{\Lambda} = \boldsymbol{U}^T\boldsymbol{A}$ by the orthonormality of the basis $\boldsymbol{\zeta}_i$.

To further reduce the cost, $\boldsymbol{A}$ can be approximated as

$$\widetilde{\boldsymbol{A}} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Lambda}} = \sum_{i=1}^{k} \sigma_i \boldsymbol{\zeta}_i \boldsymbol{\psi}_i^T \in \mathbb{R}^{M_h \times M_s}, \quad k \leq r, \tag{15}$$

where

$$\widetilde{\boldsymbol{U}} = [\boldsymbol{\zeta}_1 | \ldots | \boldsymbol{\zeta}_k] \in \mathbb{R}^{M_h \times k}, \quad \widetilde{\boldsymbol{\Lambda}} = [\sigma_1 \boldsymbol{\psi}_1 | \ldots | \sigma_k \boldsymbol{\psi}_k]^T \in \mathbb{R}^{k \times M_s}. \tag{16}$$

Generally, k is determined by prescribing a tolerance $\varepsilon_{\mathrm{POD}}$[11]

$$\Upsilon(k) = \frac{\sum_{m=1}^{k} \sigma_m^2}{\sum_{m=1}^{r} \sigma_m^2} \geq 1 - \varepsilon_{\text{POD}}^2, \tag{17}$$

where $\Upsilon(k)$ denotes the percentage of the energy preserved by the first k modes. If Eq. (17) is satisfied, we have

$$\frac{\|\widetilde{\boldsymbol{A}} - \boldsymbol{A}\|_F}{\|\boldsymbol{A}\|_F} \leq \varepsilon_{\text{POD}}, \tag{18}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

In practice, we will first construct the reduced basis $\widetilde{\boldsymbol{U}}$, and then compute the coefficient matrix with $\widetilde{\boldsymbol{\Lambda}} = \widetilde{\boldsymbol{U}}^T \boldsymbol{A}$.

2. Artificial neural network

We choose TensorFlow[12] as the framework for the ANN. In the following, we briefly introduce the multilayer feedforward network used in this work and then describe training of the network.

In this paper, the artificial neural network is a multilayer feedforward network, which is among the simplest of the large family of ANN. A typical multilayer feedforward network is plotted in Fig. 1. It is formed by three types of layers, i.e. one input layer, two hidden layers, and one output layer. Each layer consists of one or more neurons. The word *feedforward* indicates that the information flows only from the input to the output. As stated in[13], the feedforward network with two hidden layers can approximate any functions. In this work, we will use at most 3 hidden layers in the ANN.

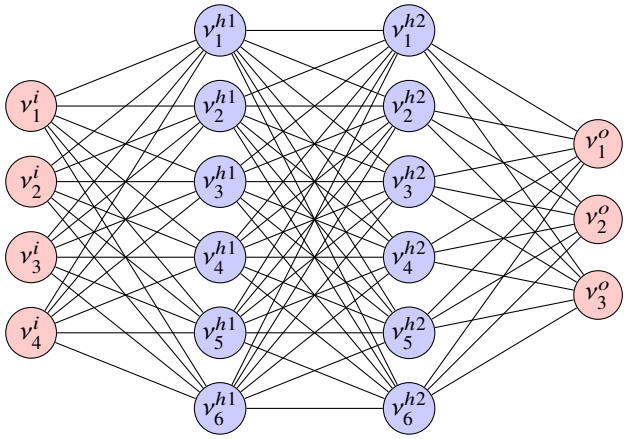


Fig. 1 Neural network

Assume we have two vectors $\boldsymbol{X} \in \mathbb{R}^{N^i}$ and $\boldsymbol{Y} \in \mathbb{R}^{N^o}$ and we seek a map $\boldsymbol{\Phi}$:

$$\widetilde{\boldsymbol{Y}} = \boldsymbol{\Phi}(\boldsymbol{X}), \tag{19}$$

where $\widetilde{\boldsymbol{Y}}$ represents the approximation of $\boldsymbol{Y}$. If we use the network shown in Fig. 1 to represent the map, we have

$$\begin{cases} Y_n^i &= X_n, & n = 1, \ldots, N^i, \\ Y_n^{h1} &= \vartheta^{h1}\left(\sum_{m=1}^{N^i} W_{n,m}^{h1} Y_m^i + B_n^{h1}\right), & n = 1, \ldots, N^{h1}, \\ Y_n^{h2} &= \vartheta^{h2}\left(\sum_{m=1}^{N^{h1}} W_{n,m}^{h2} Y_m^{h1} + B_n^{h2}\right), & n = 1, \ldots, N^{h2}, \\ Y_n^o &= \vartheta^o\left(\sum_{m=1}^{N^{h2}} W_{n,m}^o Y_m^{h2} + B_n^o\right), & n = 1, \ldots, N^o, \\ \widetilde{Y}_n &= Y_n^o, & n = 1, \ldots, N^o, \end{cases} \tag{20}$$

where $N^i = 4, N^{h1} = N^{h2} = 6, N^o = 3$ in Fig. 1. W and B are the weights and biases respectively. $\vartheta^{h1}(a), \vartheta^{h2}(a), \vartheta^o(a)$ are called *activation function*, and are a key element of the nonlinear map capability of the ANN. There are various activation functions proposed in the literature[14], and the specific choice in this work is

$$\vartheta^{h1}(a) = \vartheta^{h2}(a) = \max(a, 0), \quad \vartheta^o(a) = a. \tag{21}$$

The activation function for the two hidden layers is called the ReLU (rectified linear unit) function[15].

After having set up its structure, we train the network using an optimization approach with the following quadratic cost function

$$C = \frac{1}{M^s N^o} \sum_{m=1}^{M^s} \sum_{n=1}^{N^o} \left(\widetilde{Y}_n^m - Y_n^m\right)^2 + P \tag{22}$$

where M^s is the number of snapshots in the training set, P is a regularization term to relieve overfitting and is chosen to be

$$P = \gamma \left(\frac{1}{N^{h1} N^i} \sum_{n=1}^{N^{h1}} \sum_{m=1}^{N^i} \left(W_{n,m}^{h1}\right)^2 + \frac{1}{N^{h2} N^{h1}} \sum_{n=1}^{N^{h2}} \sum_{m=1}^{N^{h1}} \left(W_{n,m}^{h2}\right)^2 + \frac{1}{N^o N^{h2}} \sum_{n=1}^{N^o} \sum_{m=1}^{N^{h2}} \left(W_{n,m}^o\right)^2 \right). \tag{23}$$

γ is a positive coefficient, and depends on the specific problem.

For the optimization approach, we use the so-called Adam (adaptive moment estimation) algorithm[16], given in Algorithm 1. The Adam optimizer computes the exponential moving averages of the gradient($\boldsymbol{D}_k$) and its square, where the moving averages represent the estimates of the first moment and the second raw moment of the gradients. All the parameters in the Adam algorithm are fixed in this paper, i.e. $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda = 0.0001$, where β_1, β_2 are the exponential decay rates of the moving averages, ϵ is to prevent division by zero, and λ is the learning rate. Besides the training set, we need some additional snapshots, referenced to the validation set, to determine when to terminate training. The intersection of the training and validation sets is empty. Here, the nth snapshot and

its corresponding predicted solution in the validation set, consisting of M^{vs} snapshots, are denoted as $\boldsymbol{Z}^n$ and $\widetilde{\boldsymbol{Z}}^n$, respectively. The strategy is that after each training step k, the squared error E_k between the predicted and target values on the validation set is computed. If $E_k > E_{k-M^c}$, this indicates that overfitting starts to develop and the training stops. In all computations, we have fixed M^c to be 500.

Algorithm 1 Adam optimizer (All operations on vectors are element-wise. $\boldsymbol{Q}$ denotes the collection of all weights and biases in the network.)

function AdamOptimizer($\beta_1, \beta_2, \epsilon, \lambda, M^c$)
  $\boldsymbol{M}^0 \leftarrow 0, \boldsymbol{R}^0 \leftarrow 0$
  $\boldsymbol{E} \leftarrow 0$
  $k \leftarrow 1$
  while !StopCriterion($k, \boldsymbol{E}$) **do**
    $\boldsymbol{D}^k \leftarrow \nabla C^k(\boldsymbol{Q}^{k-1})$
    $\boldsymbol{M}^k \leftarrow \beta_1 \boldsymbol{M}^{k-1} + (1-\beta_1)\boldsymbol{D}^k$
    $\boldsymbol{R}^k \leftarrow \beta_2 \boldsymbol{R}^{k-1} + (1-\beta_2)(\boldsymbol{D}^k)^2$
    $\widehat{\boldsymbol{M}}^k \leftarrow \boldsymbol{M}^k/(1-(\beta_1)^k)$
    $\widehat{\boldsymbol{R}}^k \leftarrow \boldsymbol{R}^k/(1-(\beta_2)^k)$
    $\boldsymbol{Q}^k \leftarrow \boldsymbol{Q}^{k-1} - \lambda \frac{\widehat{\boldsymbol{M}}^k}{\sqrt{\widehat{\boldsymbol{R}}^k + \epsilon}}$
    $k \leftarrow k+1$
  end while
  return $\boldsymbol{Q}^{k-1}$
end function

function StopCriterion($k, M^c, \boldsymbol{E}$)
  $E_k \leftarrow$ ErrorOnValidationSet
  if $k < M^c + 1$ or $E_k \leq E_{k-M^c}$ **then**
    return False
  else
    return True
  end if
end function

function ErrorOnValidationSet
  return $\frac{1}{M^{vs}N^o}\sum_{n=1}^{M^{vs}}\sum_{m=1}^{N^o}\left(\widetilde{Z}_m^n - Z_m^n\right)^2$
end function

3. Flow field reconstruction

We are now ready to present our approach for flow field reconstruction. Instead of directly dealing with the expensive target problems, we propose to use a simplified problem which is designed to mimic the target problem. And we use the multilayer feedforward neural network to build a map between the simplified problem and the target problem. If the target problem is needed at a new position in the parameter space, one solves the simplified problem with the same parameters and use the network to obtain the solution for the target problem.

We present the detailed reconstruction method, which is divided into the offline(Algorithm 2) and online(Algorithm 3) stages. The offline stage is expensive, but needs to be conducted only once, while the online stage is very efficient

since only the simplified problem needs to be solved for the test set, which is chosen to test the accuracy of the approach.

Algorithm 2 Flow field reconstruction: Offline stage

1: As defined in Eq. (10), prepare the snapshot matrices $(\boldsymbol{A}^{sp}, \boldsymbol{A})$ for the training set, and $(\boldsymbol{I}^{sp}, \boldsymbol{I})$ for the validation set. The superscript sp denotes the simplified problem, and those without a superscript represent the target problem.
2: Perform POD on $\boldsymbol{A}$ and $\boldsymbol{A}^{sp}$ to obtain the basis matrices $\widetilde{\boldsymbol{U}}, \widetilde{\boldsymbol{U}}^{sp}$.
3: Project $\boldsymbol{A}^{sp}, \boldsymbol{I}^{sp}$ on $\widetilde{\boldsymbol{U}}^{sp}$, as $\widetilde{\boldsymbol{\Lambda}}^{sp} = (\widetilde{\boldsymbol{U}}^{sp})^T \boldsymbol{A}^{sp}, \quad \widetilde{\boldsymbol{\Gamma}}^{sp} = (\widetilde{\boldsymbol{U}}^{sp})^T \boldsymbol{I}^{sp}$. Project $\boldsymbol{A}, \boldsymbol{I}$ on $\widetilde{\boldsymbol{U}}$, as $\widetilde{\boldsymbol{\Lambda}} = \widetilde{\boldsymbol{U}}^T \boldsymbol{A}, \quad \widetilde{\boldsymbol{\Gamma}} = \widetilde{\boldsymbol{U}}^T \boldsymbol{I}$.
4: Train the network $\boldsymbol{\Phi}$ using the coefficient matrices $(\widetilde{\boldsymbol{\Lambda}}^{sp}, \widetilde{\boldsymbol{\Lambda}})$ as the training set, and $(\widetilde{\boldsymbol{\Gamma}}^{sp}, \widetilde{\boldsymbol{\Gamma}})$ as the validation set.

Algorithm 3 Flow field reconstruction: Online stage

1: As defined in Eq. (10), prepare the test set $\boldsymbol{T}^{sp}$ for the simplified problem.
2: Project $\boldsymbol{T}^{sp}$ on $\widetilde{\boldsymbol{U}}^{sp}$, as $\widetilde{\boldsymbol{\Theta}}^{sp} = (\widetilde{\boldsymbol{U}}^{sp})^T \boldsymbol{T}^{sp}$.
3: Use the network to predict the corresponding coefficients $\widetilde{\boldsymbol{\Theta}}$ for the target problem.
4: Recover the solution for the target problem with $\widetilde{\boldsymbol{T}} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Theta}}$.

III. Results

In this section, we demonstrate the flow reconstruction method on typical cases including the two-dimensional viscous nozzle flow, the inviscid M6 wing flow, and the HYFLEX hypersonic flow. The general strategy for constructing ANN is as following unless specified otherwise

1) Set the number of neurons in the hidden layers to be bounded between the number of neurons in the input and output layers. Note that there is no rigorous rule for choosing the number of neurons in the hidden layers, and other choices are also possible.
2) Set the number of hidden layers for the ANN. As discussed above, no more than three hidden layers are sufficient in this work. In this work, for each case, we will compare different number of hidden layers, and choose the one delivering the smallest error.
3) For each candidate ANN, training is conducted ten times to address the random effect from the initialization of weights and biases. The network with the smallest error on the test set will be adopted as the final model.

A. Two-dimensional viscous nozzle flow

1. Parallel inflow

We start with the two-dimensional viscous nozzle flow, and the term *parallel* indicates that the inflow is parallel to the central line of the nozzle. The geometry of the nozzle is given by

$$A(x) = \begin{cases} \dfrac{h_i + h_t}{2} - \dfrac{h_i - h_t}{2}\cos(\pi(2x-1)), & x \in [0, 0.5], \\ \dfrac{h_e + h_t}{2} - \dfrac{h_e - h_t}{2}\cos(\pi(2x-1)), & x \in [0.5, 1], \end{cases} \tag{24}$$

where $A(x)$ denotes the cross-section area of the nozzle at x, and h_i, h_t, h_e are the corresponding values of $A(x)$ at the positions of inlet, throat and exit, respectively. In this work, we fix h_i to be 0.5 and h_e to be 0.3, while h_t is varied around 0.2. The parameters forming the parameter space include the Reyholds number Re, the area of the nozzle at the throat h_t, the ratio of the static pressure at the exit and the stagnation pressure at the inlet, i.e.p_e/p_0. The parameter ranges for this case as well as the distribution of the samples in each parameter are given in Table 1. For each entry of the data set in Table 1, the number denotes the number of samples for the parameters while the term *uniform* or *random* means the distribution of samples. For a uniform distribution, we compute the parameter of the samples as

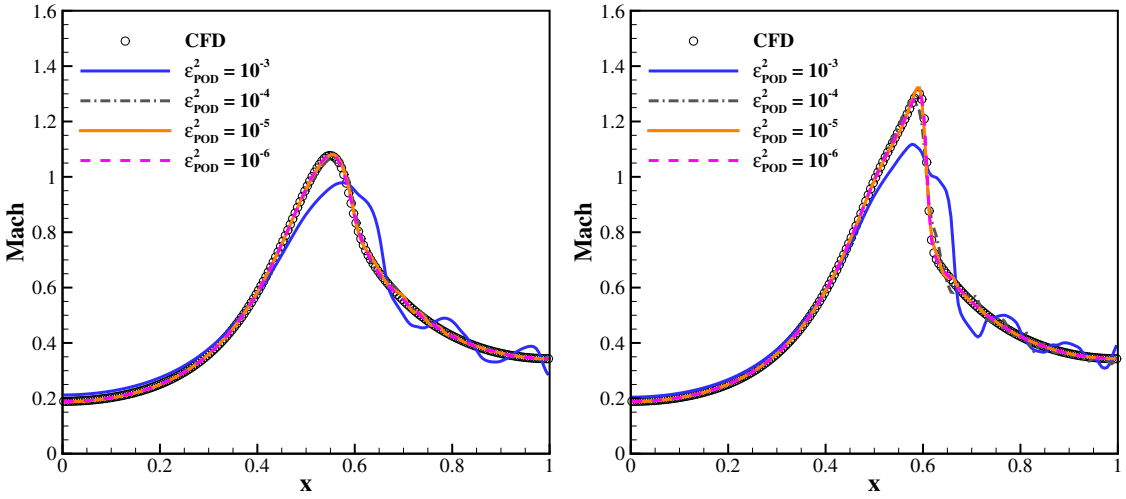
$$\begin{aligned} &\text{Training set:} \quad \chi_m = \chi_{\min} + \frac{m}{M_{\text{tr}} - 1}(\chi_{\max} - \chi_{\min}), \quad m = 1, \ldots, M_{\text{tr}}, \\ &\text{Test set:} \quad \chi_m = \chi_{\min} + \frac{m + 0.5}{M_{\text{te}}}(\chi_{\max} - \chi_{\min}), \quad m = 1, \ldots, M_{\text{te}}, \end{aligned} \tag{25}$$

where M_{tr} and M_{te} are the number of samples, and the subscripts 'max' and 'min' denotes the specified range of each parameter. The final parameter space is obtained by the tensor-product of samples from all the parameters. For this case, there are 960 samples in the training set, and 60 samples in both the test and validation set. The same strategy of constructing the data set is used in all the following cases.

Table 1 Parameter space for the two-dimensional parallel nozzle flow

	Re	h_t	p_e/p_0
Range	[500, 2000]	[0.16, 0.26]	[0.25, 0.89]
Training set	8, uniform	10, uniform	12, uniform
Test set	3, uniform	4, uniform	5, uniform
Validation set	3, random	4, random	5, random

A key factor for the proposed method is the definition of the simplified problem. This choice is generally not unique. For this case, the one-dimensional viscous nozzle flow governed by Eq. (9) is a suitable choice. Also, we consider a second simplified problem, i.e. the one-dimensional inviscid nozzle flow with Re added to account for the viscous effect. Furthermore, an extreme case, using only the flow parameters as the simplified problem, is also considered. For the 1D nozzle flow, 200 uniform elements are used in the domain. For the two-dimensional nozzle flow, we use a structured mesh of 85×30 elements for half the domain, with grid points clustered near the wall. POD is conducted for both 1D and 2D nozzle flows and the bases are obtained. Note that the POD is not needed for flow parameters we used. To determine how many modes are needed, we choose typical cases, project the corresponding solutions onto the bases of different energy loss, and use Eq. (15) to recover the approximated solution, shown in Figs. 2 and 3. As expected, with decreasing energy loss, the approximation to the CFD solution improves and $\varepsilon^2_{\text{POD}} = 10^{-5}$ is seen to generate a good approximation. The final choices for $\varepsilon^2_{\text{POD}}$ as well as the corresponding number of the required modes are listed in Table 2. For simplicity, M_{mo} denotes the minimum value for k that statifies Eq. (17) unless specified otherwise.



(a) Viscous case($Re = 500$, $h_t = 0.16$, $p_e/p_0 = 0.89$) (b) Inviscid case($h_t = 0.16$, $p_e/p_0 = 0.89$)

Fig. 2 POD approximations for the one-dimensional nozzle flow

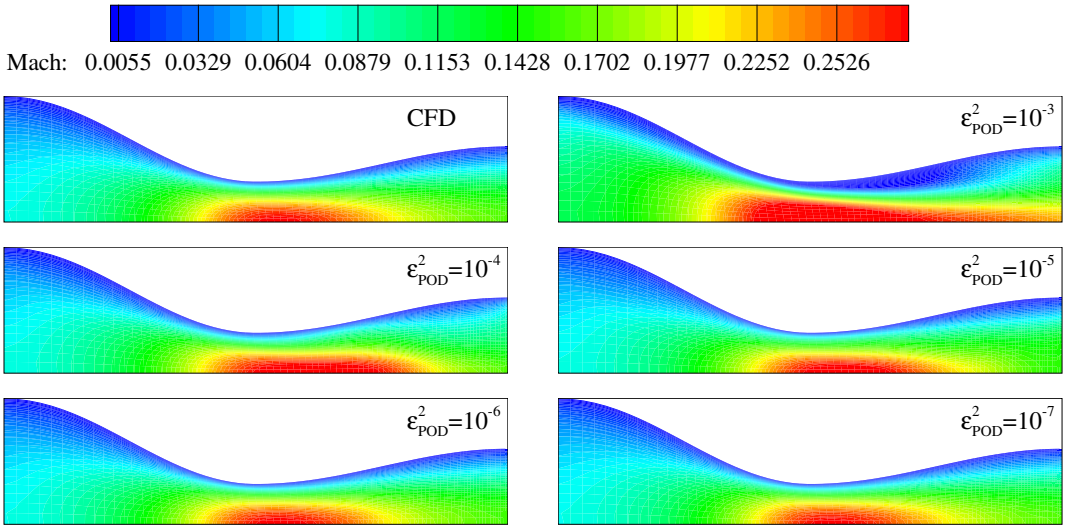


Fig. 3 POD approximations for the two-dimensional parallel nozzle flow($Re = 500$, $h_t = 0.08$, $p_e/p_0 = 0.89$)

Table 2 POD approximations chosen for the parallel nozzle flow case

	1D viscous nozzle flow	1D inviscid nozzle flow	2D viscous nozzle flow
$\varepsilon^2_{\text{POD}}$	10^{-5}	10^{-5}	10^{-5}
M_{mo}	28	29	14

The ANN parameters for the three cases of the simplified problems are listed in Table 3. Note that for the case using $\left(\boldsymbol{A}_{\text{inv}}^{\text{sp}}, Re\right)$ (i.e. the 1D inviscid nozzle flow as well as the Reynolds number) as input only one-hidden layer is employed, while two hidden layers are used for the other two. Also, for comparison purposes, for the case $(Re, h_t, p_e/p_0)$ we take the same hidden layers as that for $\left(\boldsymbol{A}_{\text{vis}}^{\text{sp}}\right)$. The solution vector for this case is chosen to be $[\boldsymbol{\rho}^T \quad \boldsymbol{u}^T \quad \boldsymbol{p}^T \quad \boldsymbol{Ma}^T]^T$. For comparison, we also conduct linear regression(i.e. no hidden layer) for all cases. The relative L^2 error for the Mach number on the test set are presented in Figs. 4 and 5. As is clear, the linear regression generates much larger error than the nonlinear ANN. For this case, the simplest parametric 1D case generates the most accurate solution. We plot the solution of typical cases predicted by the ANN along the central line in Fig. 6, observing that the comparison agrees well with the error given in Fig. 5.

Table 3 ANN parameters for the parallel nozzle flow case

	Input: $\left(\boldsymbol{A}_{\text{vis}}^{\text{sp}}\right)$	Input: $\left(\boldsymbol{A}_{\text{inv}}^{\text{sp}}, Re\right)$	Input: $(Re, h_t, p_e/p_0)$
N^i	28	30	3
N^{h1}	28	30	28
N^{h2}	21	—	21
N^o	14	14	14

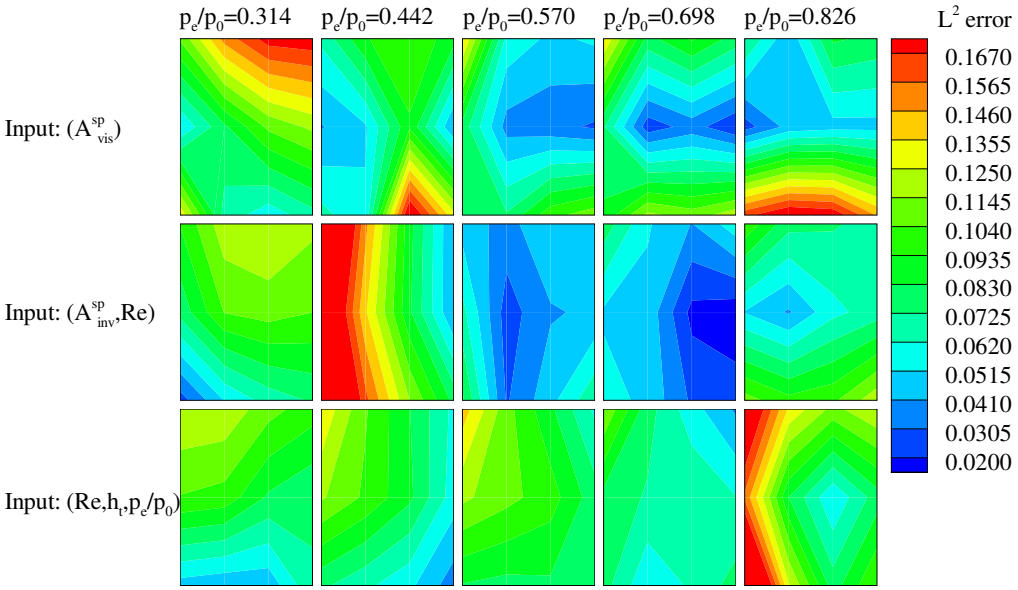


Fig. 4 Relative L^2 error for the Mach number of the parallel nozzle flow on the test set. Linear regression.

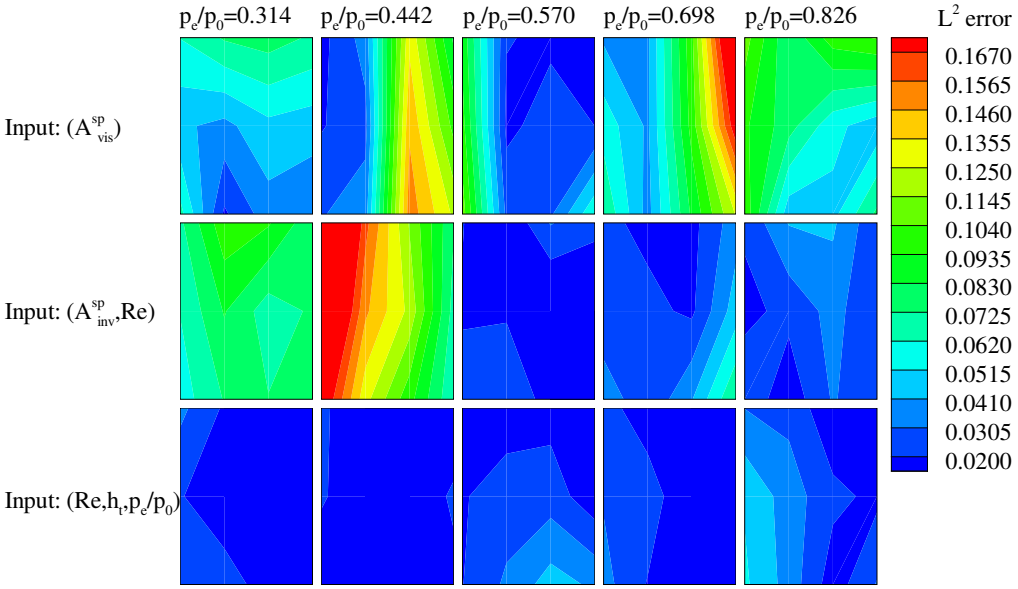
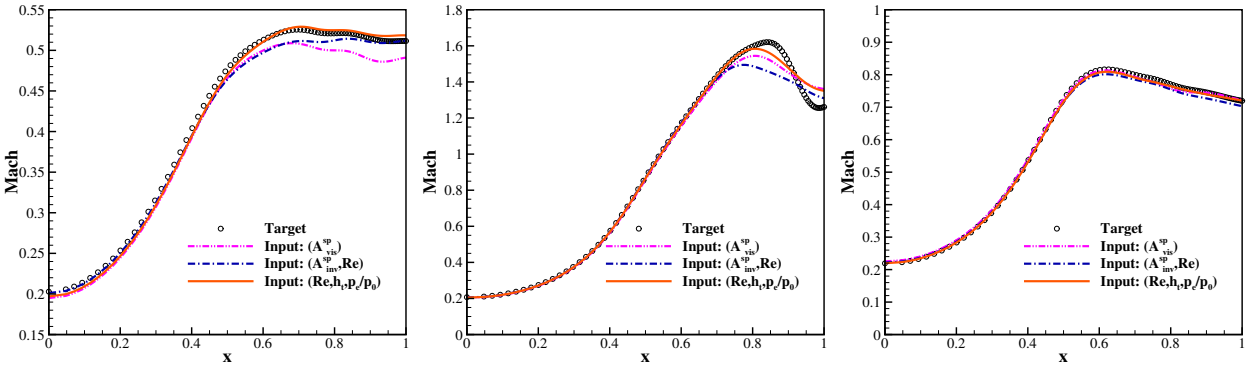


Fig. 5 Relative L^2 error for the Mach number of the parallel nozzle flow on the test set. ANN.



(a) $Re = 1250, h_t = 0.124, p_e/p_0 = 0.826$ (b) $Re = 1750, h_t = 0.0863, p_e/p_0 = 0.314$ (c) $Re = 1750, h_t = 0.0988, p_e/p_0 = 0.698$

Fig. 6 Mach number profile of the parallel nozzle flow along the central line

2. Non-parallel inflow

In this case, we fix h_t to be 0.2, and incorporate the inflow angle as another parameter. This case is particularly interesting since the one-dimensional nozzle flow can not account for the non-parallel inflow, and such a situation is likely to occur for practical problems. Our solution for this is to add meta-parameters such as the inflow angle explicitly to the input end of the ANN, in addition to the one-dimensional nozzle flow. The parameters range is given in Table 4, and the chosen POD approximation is given in Table 5. The ANN for this case includes one hidden layer of 13 neurons. The solution vector for this case is the same as that in the parallel case.

In Fig. 7, the relative L^2 error for the Mach number on the test set is shown. In Figs. 8 and 9, two typical cases are chosen to compare the predicted and target solutions. As is clear, explicitly adding the inflow angle performs well, demonstrating the feasibility of the proposed method for a wide range of problems.

Table 4 Parameter space for the non-parallel nozzle flow case

	Re	α	p_e/p_0
Range	[500, 2000]	[0°, 45°]	[0.25, 0.89]
Training set	8, uniform	10, uniform	12, uniform
Test set	3, uniform	4, uniform	5, uniform
Validation set	3, random	4, random	5, random

Table 5 POD approximations for the non-parallel nozzle flow case

	1D viscous nozzle flow	2D viscous nozzle flow
$\varepsilon^2_{\text{POD}}$	10^{-6}	10^{-5}
M_{mo}	17	13

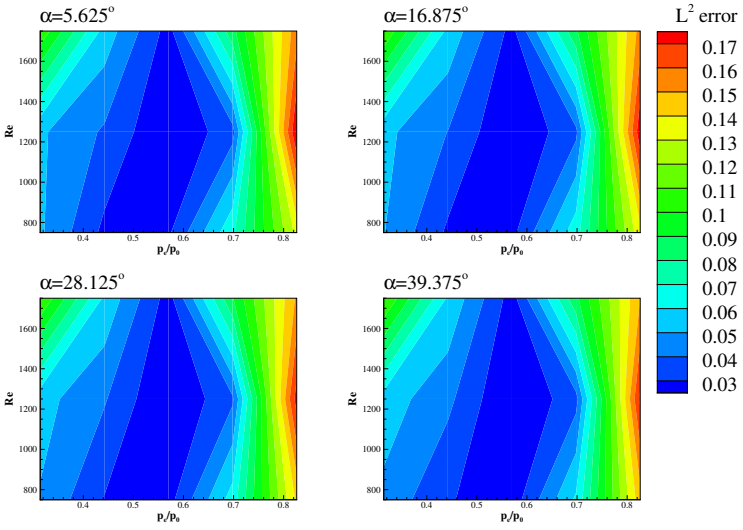


Fig. 7 Relative L^2 error for the Mach number of the non-parallel nozzle flow on the test set. ANN.

B. Inviscid M6 wing flow

In this section, we turn to the Onera M6 wing, which is intended to test the performance of the proposed method for three-dimensional cases. The parameters are chosen to be Mach number and the angle of attack. Similar to the parallel nozzle case, we consider three versions of simplified problems: 1) Inviscid NACA0012 flow at different Ma and α; 2) Inviscid NACA0012 flow at varying Ma and fixed α ($\alpha = 0°$) as well as the parameter α; 3) Parameters of varying Ma and α.

The parameter space for this case is given in Table 6. The mesh for the NACA0012 airfoil and the M6 wing are obtained from the test case suite of SU2[17]. The NACA0012 mesh consists of 10216 triangular elements, with 200 points along the wall. The M6 wing consists of 582752 tetrahedral elements, and the wing surface has 36454 triangular elements. And the final POD approximation corresponding to the parameter space is listed in Table 7. In Table 8, we present the details of the neural networks, which consist of three-hidden layers. For the purpose of comparison, the hidden layers are the same for the three cases. The solution vector for this case is chosen to be $[\boldsymbol{\rho}^T \quad \boldsymbol{u}^T \quad \boldsymbol{v}^T \quad \boldsymbol{p}^T \quad \boldsymbol{Ma}^T]^T$.

The relative L^2 error for the Mach number is plotted in Fig. 12. We observe that using airfoil flow (i.e. $\boldsymbol{A}_{\alpha}^{\mathrm{sp}}$) as input generates the smallest error for most samples in the test set, while the approach using $\left(\boldsymbol{A}_{\alpha=0°}^{\mathrm{sp}}, \alpha\right)$ is seen to be the least accurate from a global view. To better compare the prediction accuracy, we plot the pressure distribution at different sections along the wing span, including a subsonic case (Fig. 13) and a transonic case (Fig. 14). The approach

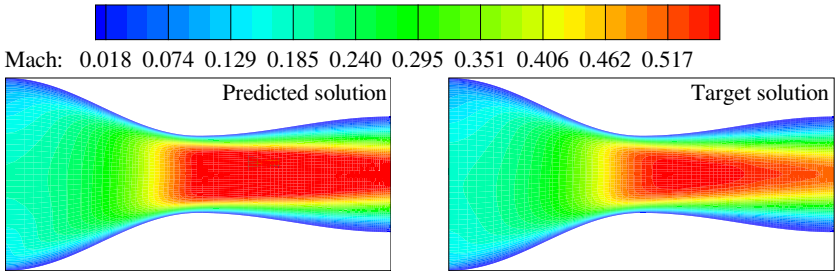


Fig. 8 Mach number contours for the non-parallel nozzle flow($Re = 1750, \alpha = 5.625°, p_e/p_0 = 0.826$)**.**

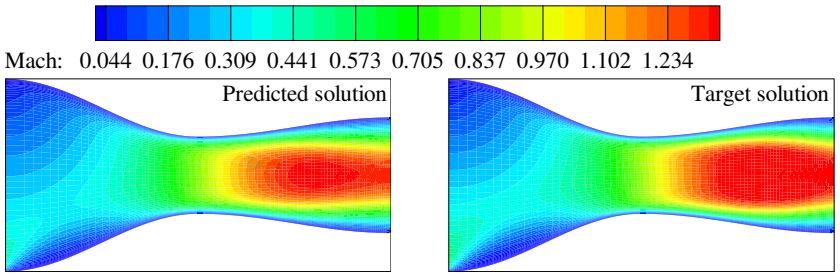


Fig. 9 Mach number contours for the non-parallel nozzle flow($Re = 1750, \alpha = 39.375°, p_e/p_0 = 0.314$)**.**

using $\left(\boldsymbol{A}_{\alpha}^{\mathrm{sp}}\right)$ generates the best agreement, while significant error occurs for the one using (Ma, α). Note that the slight inconsistency of the comparison in Fig. 12-Fig. 14 is reasonable considering that the integral norm is used for the L^2 error, while the error for pressure is point-wise.

Table 6 Parameter space for the M6 wing case

	Ma	α
Range	$[0.2, 0.85]$	$[0°, 3.06°]$
Training set	13, uniform	13, uniform
Test set	7, uniform	7, uniform
Validation set	7, random	7, random

Table 7 POD approximations for the M6 wing case

	NACA0012 flow at varying Ma and α	NACA0012 flow at varying Ma and fixed $\alpha(\alpha = 0°)$	M6 wing flow
$\varepsilon^2_{\mathrm{POD}}$	10^{-6}	10^{-7}	10^{-7}
M_{mo}	40	5	20

Table 8 ANN parameters for the M6 wing case

	Input: $\left(\boldsymbol{A}_{\alpha}^{\mathrm{sp}}\right)$	Input: $\left(\boldsymbol{A}_{\alpha=0°}^{\mathrm{sp}}, \alpha\right)$	Input: (Ma, α)
N^i	40	6	2
N^{h1}	40	40	40
N^{h2}	30	30	30
N^{h3}	25	25	25
N^o	20	20	20

C. Hypersonic flight experiment (HYFLEX) case

HYFLEX is a typical hypersonic vehicle (see Fig. 15) for entry flight and is one of a series of experiments within the Japanese H-II orbiting plane (HOPE) program[18], that has been widely used as a test case of numerical methods for hypersonic flows. Hence, it is interesting to test the proposed method on this complex hypersonic case. We use the 2D flow around the symmetry plane of HYFLEX as the simplified problem. Using only the parameters as input is also considered.

The mesh consists of 53,448 quadrilateral elements and 6,051,792 hexahedral elements for 2D and 3D configurations, respectively, with grid points clustered toward the solid surface. In order to reduce the computational cost, we consider varying only one parameter (i.e. Mach number) for the parameter space. The Mach number ranges from 2 to 6 with the angle of attack fixed to be 30°, and the sizes for the training, test and validation sets are 101, 22, and 22, respectively. Similar to the previous examples, uniform sampling is adopted for the training and test sets while random sampling is used for the validation set. We notice that, although we are working with non-dimensional variables, the values for

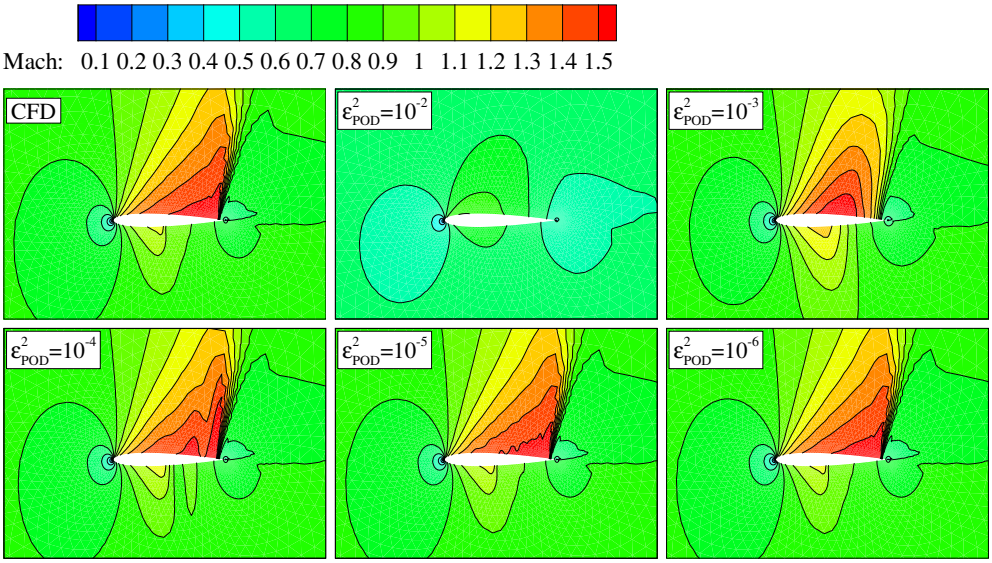


Fig. 10 POD approximations for the inviscid NACA0012 flow($Ma = 0.85, \alpha = 3.06°$)

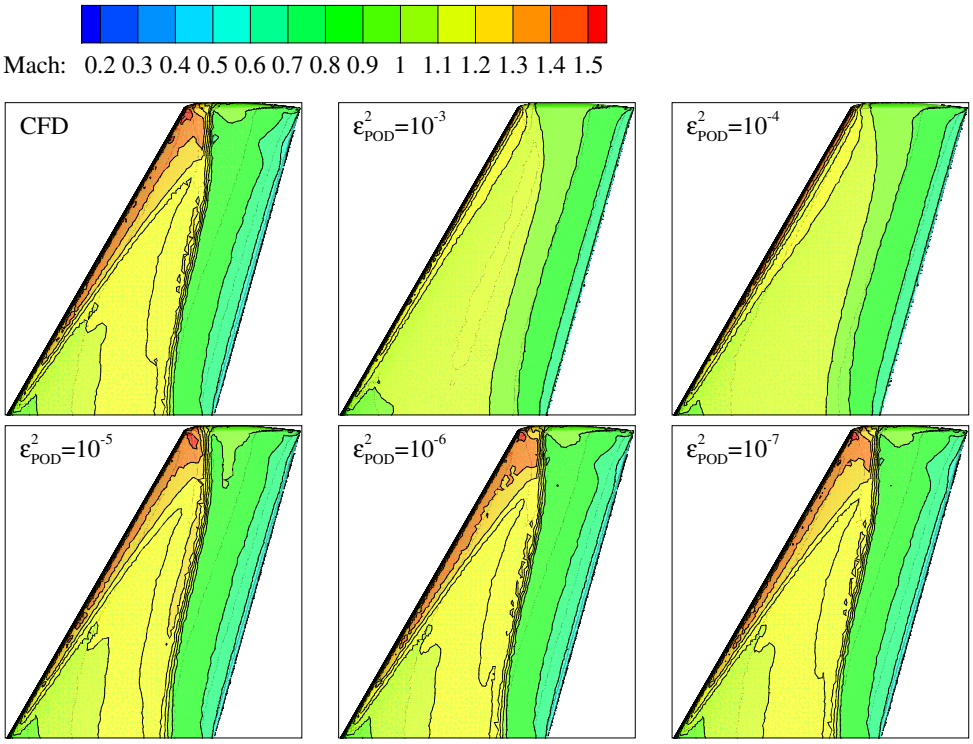


Fig. 11 POD approximations for the inviscid M6 wing flow($Ma = 0.85, \alpha = 3.06°$)

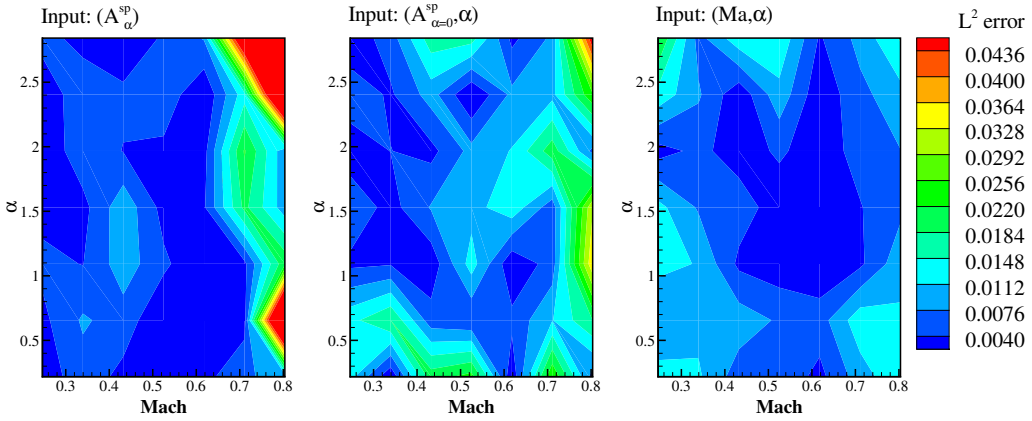
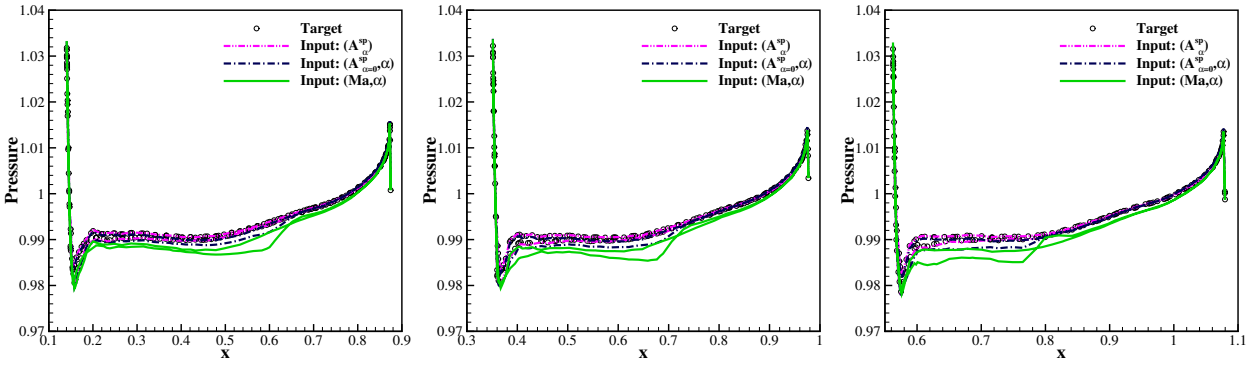
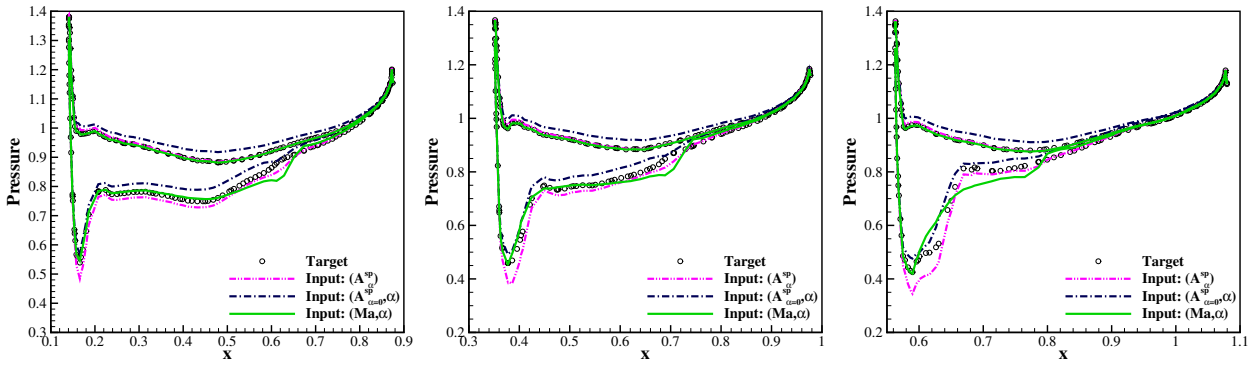


Fig. 12 Relative L^2 error for the Mach number of the M6 wing case on the test set. ANN.



(a) Section 1 (b) Section 2 (c) Section 3

Fig. 13 Comparison of the pressure on the wall at different sections(Section 1: 20%, Section 2: 50%, Section 3: 80%. The sections are specified with respect to the wing span, originating from the symmetry plane). $Ma = 0.246, \alpha = 0.219°$.



(a) Section 1 (b) Section 2 (c) Section 3

Fig. 14 Comparison of the pressure on the wall at different sections(Section 1: 20%, Section 2: 50%, Section 3: 80%. The sections are specified with respect to the wing span, originating from the symmetry plane). $Ma = 0.804, \alpha = 2.84°$**.**

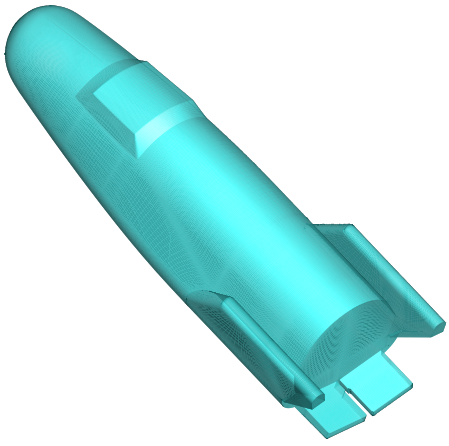


Fig. 15 Illustration of the geometry for HYFLEX

different fields show significant variances for this hypersonic case. Consequently, for a similar energy loss, the number of the POD modes is large, making the training of ANN difficult. Therefore, we adopt a slightly different strategy and treat each flow component of interest (pressure and Mach number in this section) separately in terms of both POD and training. In Figs. 16 and 17, we show the approximation accuracy with different POD modes preserved for pressure, and the final choices for the POD approximations are given in Table 9. Note that the values of $\varepsilon^2_{\text{POD}}$ in Table 9 (and their corresponding values in Figs. 16 and 17) are in terms of M_{mo} (Recall that for the other cases, M_{mo} is the smallest number satisfying $\varepsilon^2_{\text{POD}}$). It should be noted that the size of the output variables for the Mach component is larger than for the pressure component, while the energy loss remains comparable. To gain a clear insight on this, we plot the singular values for the 3D case in Fig. 18, from which it can be seen that the value decreases more rapidly for pressure, causing the first modes to be more dominant. This is likely due to the fact that for the pressure component, only the region between the bow shock and the solid surface changes with different flow conditions, and most part of the solution vector remains unchanged across different samples. The contrary is true for the Mach component. This indicates that some fields (pressure) are better than others (Mach) and that there can be benefits to identify those best suited for compression. However, this remains an open question and requires further investigation.

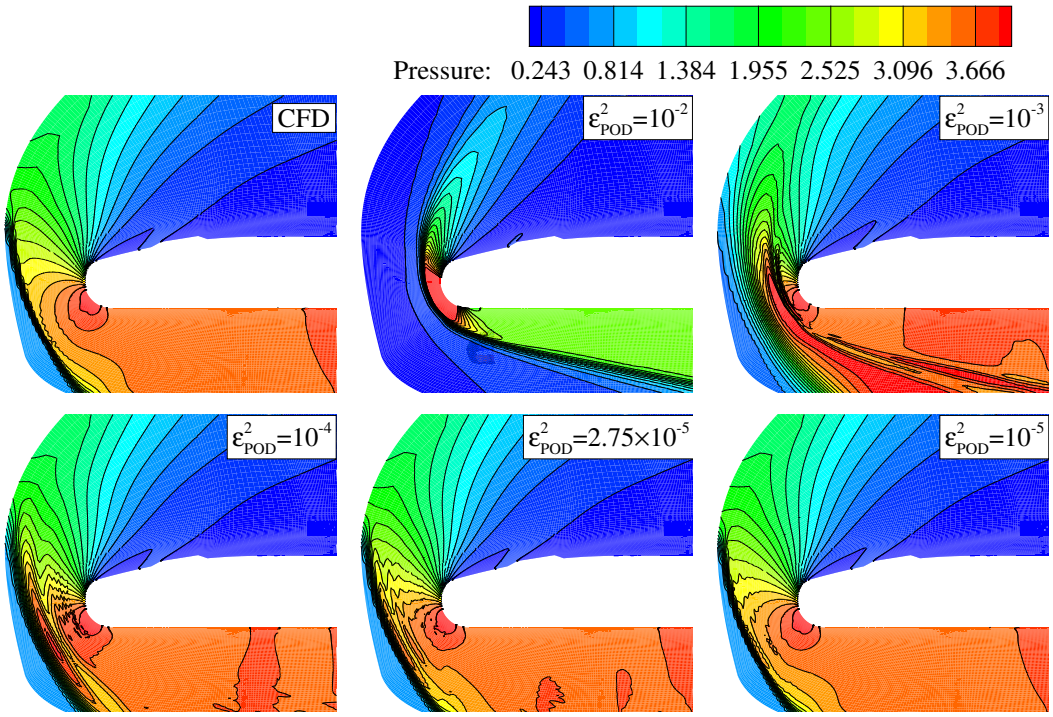


Fig. 16 POD approximations for the 2D flow around the symmetry plane of HYFLEX($Ma = 2, \alpha = 30°$)

The chosen ANN parameters are given in Table 10. Note that we keep the hidden layers unchanged for each component to make a fair comparison. The strategy of using directly the parameter as input is also considered. The

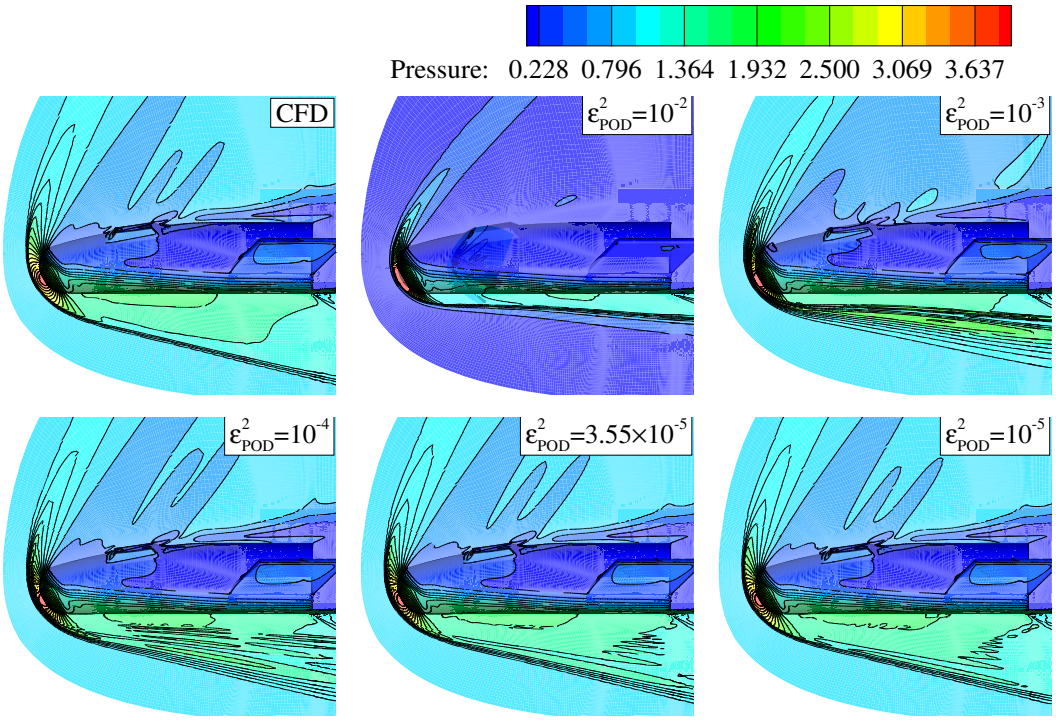


Fig. 17 POD approximations for the HYFLEX flow($Ma = 2, \alpha = 30°$)

Table 9 POD approximations for the HYFLEX case

	Pressure		Mach	
	2D	3D	2D	3D
$\varepsilon^2_{\text{POD}}$	2.75×10^{-5}	3.55×10^{-5}	3.90×10^{-5}	1.56×10^{-5}
M_{mo}	30	15	35	85

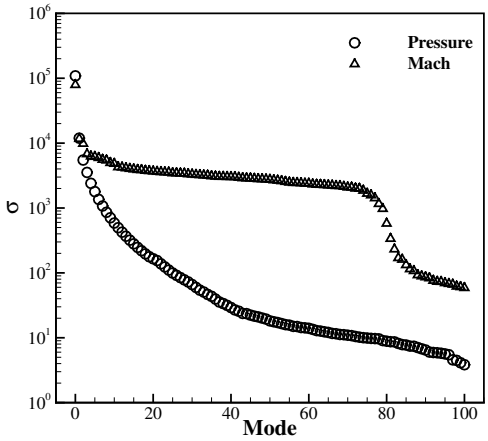
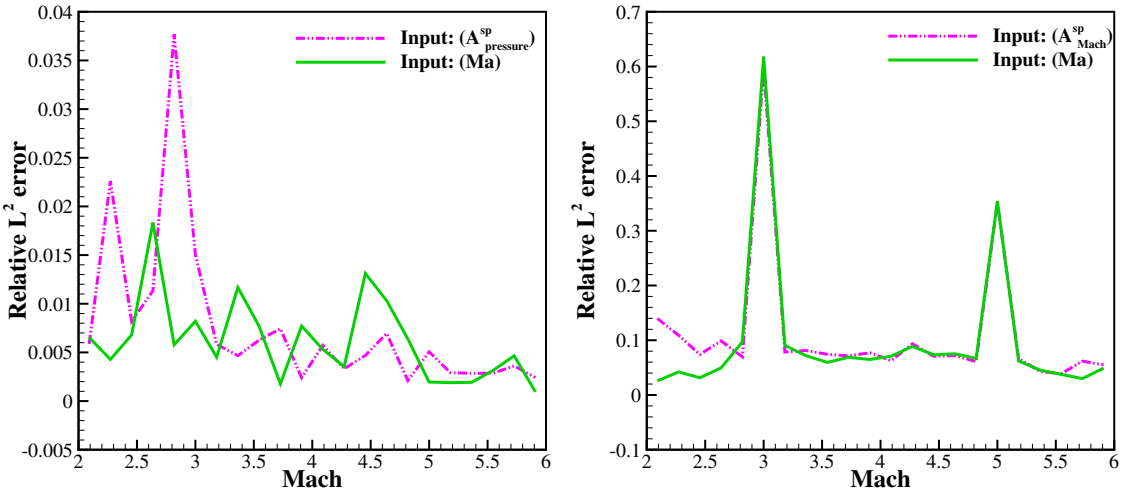


Fig. 18 Singular values for the 3D HYFLEX flow

relative L^2 error for pressure and Mach components is plotted in Fig. 19. As can be seen, the error for Mach is much larger than that for pressure, due to the large number of unknowns (i.e. weights and biases), compared to the size of the corresponding training set. In Figs. 20 to 23, we compare the predicted results by ANN and the target solution for typical cases. For the pressure, all the results agree well with each other, indicating that the number of the training samples could be further reduced. On the contrary, the comparison for Mach number shows that the error is large, possibly due to insufficient training samples.

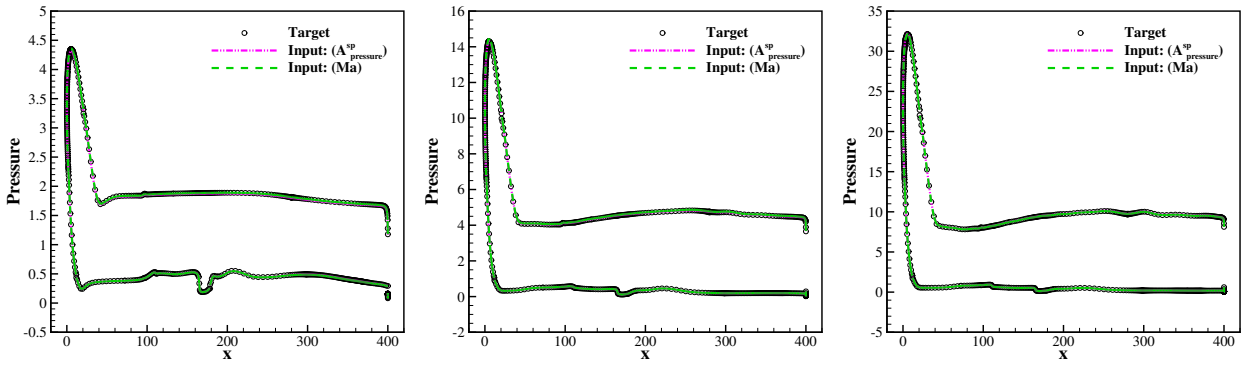
Table 10 ANN parameters for the HYFLEX case

	Pressure		Mach	
	Input: $\left(A^{sp}_{pressure}\right)$	Input: (Ma)	Input: $\left(A^{sp}_{Mach}\right)$	Input: (Ma)
N^i	30	1	35	1
N^{h1}	25	25	52	52
N^{h2}	20	20	69	69
N^o	15	15	85	85



(a) Pressure (b) Mach

Fig. 19 Relative L^2 error of the HYFLEX case on the test set. ANN.



(a) $Ma = 2.09$ (b) $Ma = 3.91$ (c) $Ma = 5.91$

Fig. 20 Comparison of the pressure on the wall of the symmetry plane of HYFLEX.

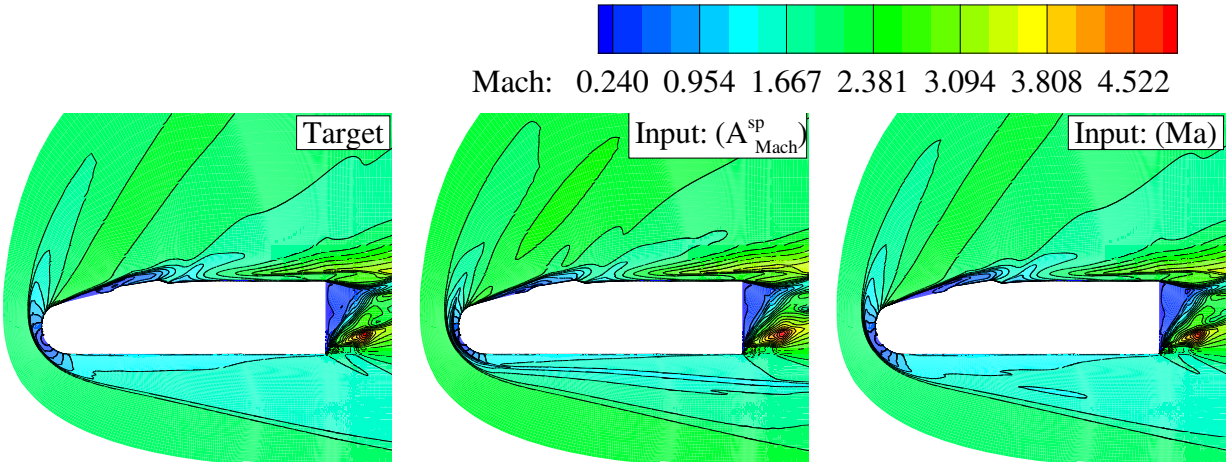


Fig. 21 Comparison of the Mach number on the symmetry plane($Ma = 2.09$).

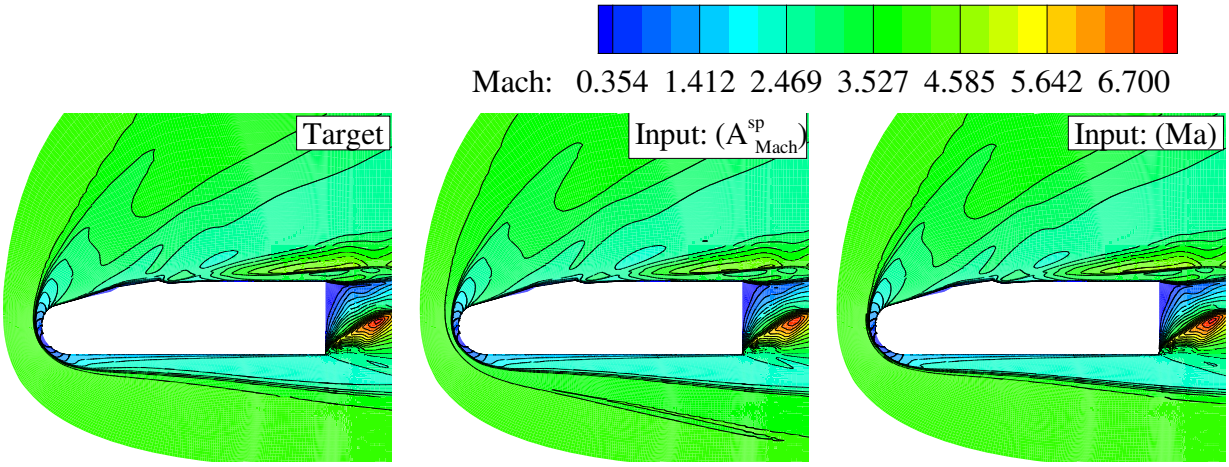


Fig. 22 Comparison of the Mach number on the symmetry plane($Ma = 3.91$).

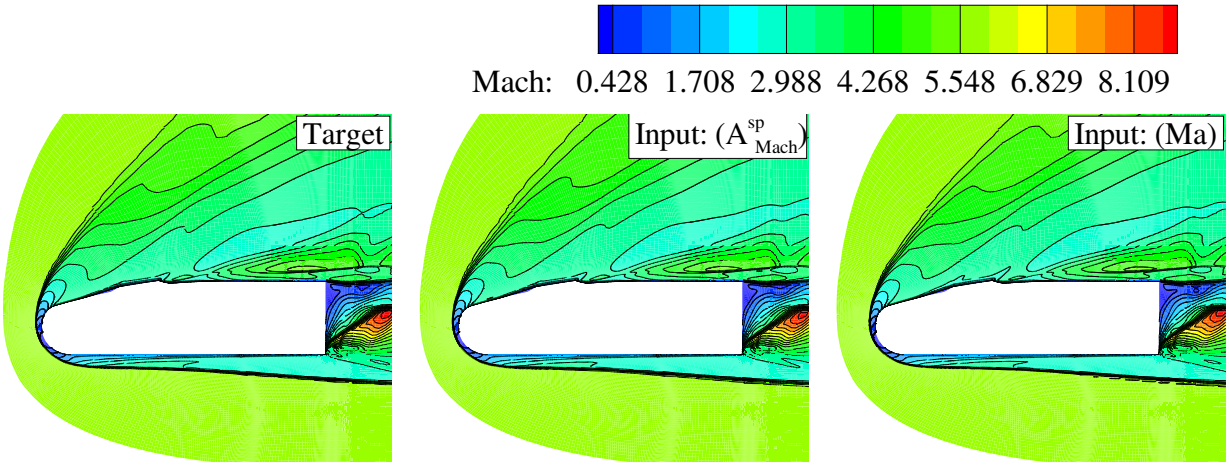


Fig. 23 Comparison of the Mach number on the symmetry plane(Ma = 5.91).

IV. Conclusion

In this work, we propose a flow reconstruction method based on the use of artificial neural network. Given a target problem (which is usually expensive to solve), we first define a simplified problem, and solve both problems to obtain the training, test, and validation samples. The simplified problem may be related to the target problem at different levels, ranging from dimensional reduction to direct use of parameters. Then, with a POD, the training samples for both the simplified problem and the target problem are further decomposed into bases and coefficients. The bases are fixed for a certain problem, while the coefficients vary to generate different solutions. The relation of the coefficients between the simplified problem and the target one is constructed with an ANN. To obtain the solution of the target problem at new parameter values, one only needs to solve the simplified problem, and then use the ANN to recover the coefficients and efficiently predict the solution of the original problem.

Several typical test cases, including the two-dimensional viscous nozzle flow, the inviscid M6 wing flow, and the viscous HYFLEX hypersonic flow, are conducted to test the proposed flow reconstruction method. Overall, the proposed method is shown to work well for this wide range of problems with an ANN of no more than three hidden layers. Shocks can also be captured well without obvious stair-cased arrays of jumps observed. Since the ANN is nonlinear, the solution vector for each sample can be the combination of multiple field components, or contain only one single component. Regarding the choice of the simplified problem, it is observed that using a simplified problem that can fully account for the whole parameter space is more robust in terms of accuracy.

The POD-based approach proposed in this paper tends to require a large number of samples since the preserved POD modes are generally much fewer than the samples. The generation of the samples, especially for the target problems, may still be expensive. A possible solution for this issue is to employ a greedy algorithm, which is part of our future work.

Funding Sources

This research was supported by the China Scholarship Council (Grant No. 201606025011).

Acknowledgments

The authors thank Deep Ray for helpful discussions on the implementations of artificial neural network. All the simulations are conducted on the high performance computing clusters of SCITAS at EPFL.

References

[1] Lorente, L., Vega, J., and Velazquez, A., "Generation of aerodynamic databases using high-order singular value decomposition," *Journal of Aircraft*, Vol. 45, No. 5, 2008, p. 1779.

[2] Tan, B. T., "Proper orthogonal decomposition extensions and their applications in steady aerodynamics," *Master of Engineering in High Performance Computation for Engineered Systems (HPCES), Singapore-MIT Alliance*, 2003.

[3] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition," *AIAA Journal*, Vol. 42, No. 8, 2004, pp. 1505–1516.

[4] Everson, R., and Sirovich, L., "Karhunen–Loeve procedure for gappy data," *JOSA A*, Vol. 12, No. 8, 1995, pp. 1657–1664.

[5] Alonso, D., Vega, J. M., Velázquez, Á., and de Pablo, V., "Reduced-order modeling of three-dimensional external aerodynamic flows," *Journal of Aerospace Engineering*, Vol. 25, No. 4, 2011, pp. 588–599.

[6] Mifsud, M., MacManus, D. G., and Shaw, S., "A variable-fidelity aerodynamic model using proper orthogonal decomposition," *International Journal for Numerical Methods in Fluids*, Vol. 82, No. 10, 2016, pp. 646–663.

[7] Alonso, D., Vega, J., and Velazquez, A., "Reduced-order model for viscous aerodynamic flow past an airfoil," *AIAA journal*, Vol. 48, No. 9, 2010, pp. 1946–1958.

[8] Tosun, E., Aydin, K., and Bilgili, M., "Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures," *Alexandria Engineering Journal*, Vol. 55, No. 4, 2016, pp. 3081–3089.

[9] Cybenko, G., "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, Vol. 2, No. 4, 1989, pp. 303–314.

[10] Quarteroni, A., Manzoni, A., and Negri, F., *Reduced Basis Methods for Partial Differential Equations: An Introduction*, 1st ed., Springer, Switzerland, 2015, Vol. 92, Chap. 6.

[11] Hesthaven, J. S., Rozza, G., Stamm, B., et al., *Certified reduced basis methods for parametrized partial differential equations*, Springer, 2016.

[12] Abadi, M., Agarwal, A., Barham, P., and et al, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," , 2015. URL `:https://www.tensorflow.org/`.

[13] Cybenko, G., "Continuous Valued Neural Networks with Two Hidden Layers are Sufficient," Tech. rep., 1988.

[14] Karlik, B., and Olgac, A. V., "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, Vol. 1, No. 4, 2011, pp. 111–122.

[15] Glorot, X., Bordes, A., and Bengio, Y., "Deep sparse rectifier neural networks," *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

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

[17] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., "SU2: an open-source suite for multiphysics simulation and design," *AIAA Journal*, 2015.

[18] Watanabe, S., Ishimoto, S., and Yamamoto, Y., "Aerodynamic Characteristics Evaluation of Hypersonic Flight Experiment Vehicle Based on Flight Data," *Journal of Spacecraft Rockets*, Vol. 34, No. 4, 1997, pp. 464–470.