

RESEARCH ARTICLE | JANUARY 16 2019

Machine learning methods for turbulence modeling in subsonic flows around airfoils

Linyang Zhu; Weiwei Zhang; Jiaqing Kou  ; Yilang Liu



Physics of Fluids 31, 015105 (2019)

<https://doi.org/10.1063/1.5061693>



View
Online



Export
Citation

CrossMark

AML Machine Learning WEBINAR

Fostering a New Data Culture with *APL Machine Learning*



12:00PM (noon) EST



Thursday, January 18, 2024

Moderated by:
Adnan Mehnec



Machine learning methods for turbulence modeling in subsonic flows around airfoils

Cite as: Phys. Fluids 31, 015105 (2019); doi: 10.1063/1.5061693

Submitted: 22 September 2018 • Accepted: 20 December 2018 •

Published Online: 16 January 2019



View Online



Export Citation



CrossMark

Linyang Zhu, Weiwei Zhang, Jiaqing Kou, and Yilang Liu

AFFILIATIONS

School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

ABSTRACT

In recent years, the data-driven turbulence model has attracted widespread concern in fluid mechanics. The existing approaches modify or supplement the original turbulence model by machine learning based on the experimental/numerical data, in order to augment the capability of the present turbulence models. Different from the previous researches, this paper directly reconstructs a mapping function between the turbulent eddy viscosity and the mean flow variables by neural networks and completely replaces the original partial differential equation model. On the other hand, compared with the machine learning models for the low Reynolds (Re) number flows based on direct numerical simulation data, high Reynolds number flows around airfoils present the apparent scaling effects and strong anisotropy, which induce large challenges in accuracy and generalization capability for the machine learning algorithm. We mainly concentrate on the high Reynolds number turbulent flows around the airfoils and take the results calculated by the computational fluid dynamics solver with the Spallart–Allmaras (SA) model as training data to construct a high-dimensional data-driven network model based on machine learning. The radial basis function neural network and the auxiliary optimization methods are adopted, and the individual models are built separately for the flow fields of the near-wall region, wake region, and far-field region. The training data in this paper is extracted from only three subsonic flow fields of NACA0012 airfoil. The data-driven turbulence model can be applied to various airfoils and flow states, and the predicted eddy viscosity, lift/drag coefficients, and skin friction distributions are all in good agreement with the results of the original SA model. This research demonstrates the promising prospect of machine learning methods in future studies about turbulence modeling.

Published under license by AIP Publishing. <https://doi.org/10.1063/1.5061693>

12 January 2024 23:06:38

I. INTRODUCTION

In 1883, Reynolds discovered the turbulent state in pipe flow, marking the start of turbulent flow research. Based on the contribution of earlier researchers, we now get more and more profound physical insight into turbulent flows. However, the essence of turbulence and how to utilize and control turbulent flows more efficiently are still very challenging for researchers and engineers. At present, numerical simulation and experiments are the main sources to obtain turbulence results for engineering problems. Generally, high Reynolds (Re) number experiments in aeronautics are not only expensive and time-consuming, but also pretty hard to achieve the accurate measurement, especially in the boundary layer. The numerical methods can be further classified as direct numerical simulation (DNS), large eddy simulation (LES), and Reynolds averaged Navier–Stokes simulation (RANS) according to different scales modeled.

In computational fluid dynamics (CFD), with the improvement of computational capability, high-fidelity methods, like DNS and LES, have been increasingly used in turbulence computations and have got some achievement in some practical contexts. However, DNS is still impractical in aeronautical industries due to the extremely high grid resolution, which is exponentially proportional to the Reynolds number. Besides, the aircraft simulation across the full flight envelope by LES needs more than high-performance computing (HPC) advances and improvements in algorithmic technology, which might not be realized until 2030.¹ Hence, RANS models will remain as the critical approaches in the engineering practice during the foreseeable future.²

In traditional turbulent models, the Reynolds stress can be obtained by explicit expression³ or solving partial differential equations (PDEs).⁴ All of these models are extensively adopted in general engineering applications because

of their high efficiency and easy implementation. However, the universality of these models is limited since some prescribed parameters should be determined beforehand, which are derived from some specific experiments and DNS results. Moreover, for some complex separated flows, turbulence modeling is significantly difficult due to the anisotropy.^{5,6} Besides, large discrepancies between turbulence models may occur, which force the users to choose the appropriate turbulence model according to their experience and the specific problem of interest. Although Reynolds stress transport models (RSTM) can obtain higher accuracy, its complicated transport equations and poor convergence lead to less popularity in practical application.

The shortcomings of RANS models mentioned above are difficult to overcome from traditional studies. Recently, some new technologies based on data mining and machine learning have shown their potential in some domains, like speech and visual recognition,^{7,8} signal processing⁹ and the reduced-order framework,¹⁰ etc. In turbulence, Milano and Koumoutsakos¹¹ approximated the high order terms by neural networks. Hocevar *et al.*¹² modeled some turbulent variables in the airfoil wake region by radial basis function neural networks (RBFNN). In fact, the research work that formally adopts the data-driven method to improve or replace the RANS model has been mainly carried out in the past five years.¹³ Duraisamy *et al.*^{14–16} modeled the source terms in the Spallart-Allmaras (SA) turbulence model by neural networks and embedded it into the CFD solver. Jin *et al.*¹⁷ constructed a $C_p - \mathbf{u}$ model to establish the relationship between the pressure field on the cylinder surface and the wake structure by fusion convolutional neural networks. The study above is mainly focused on the modeling of turbulence related variables. The purpose of the other research is to reduce the uncertainties between high fidelity data and results of RANS models. For example, Duraisamy and Singh *et al.*^{18–20} combined the inversion model and machine learning to infer and reconstruct better functional forms in turbulence and transitions modeling; Xiao *et al.*^{21–25} proposed the concept of “physics-informed machine learning (PIML)” to emphasize the importance of including the physical domain knowledge into machine learning. He *et al.*²⁶ proposed the adjoint-based data assimilation (ABDA) model to modify the SA model by a spatially varying coefficient distribution. Different from these studies, other studies are based purely on high fidelity data rather than classic RANS models, which further extended the application of machine learning to turbulence modeling and verified the positive prospect of data-driven methods. For example, Ling *et al.*^{27–30} embedded the invariance property into deep neural networks first and demonstrated the advantage over the architecture without this property. Different from the two kinds of study, Ling’s work was based purely on high fidelity data rather than classic RANS models, which further developed the application of machine learning to turbulence modeling and verified the positive prospect of data-driven methods. Similarly, Gamabara and Hattori³¹ adopted the artificial neural network (ANN) to model the subgrid-scale stress in LES. A detailed description about

turbulence modeling with data-driven techniques is reviewed in Ref. 32.

Although some new studies have been done recently,^{33,34} the development of traditional RANS models has reached a plateau and it is hard to achieve the essential change and improvement. It is perhaps time to make some transformative impacts on modeling turbulent flows with machine learning methods.³⁵ *Different from those studies to improve baseline RANS models or to target Reynolds stress modeling of low/middle Reynolds number flows, this paper underlines the practicality of reconstructing function form of turbulent eddy viscosity for high Reynolds number airfoil flows. The proposed model is an algebraic and purely data-driven black-box model.* It should also be noted that, since *the current approach does not need to solve PDEs, lower computational cost than the original SA model is expected.* The rest of this paper is organized as follows. Section II introduces the methods of model constructing, including the sample selection, model framework, and optimization of the model parameters. Section III characterizes the training and predicting datasets and the numerical results. Finally, conclusions and future outlook are addressed in Sec. IV.

II. METHOD

A. Modeling process

The proposed approach to reconstruct an eddy viscosity function can be divided into two parts: the learning machine and the surrogate machine. The learning machine mainly includes the sample selection, model framework, and parameter optimization. In the surrogate machine part, the proposed model is inserted into the CFD solver, then the eddy viscosity

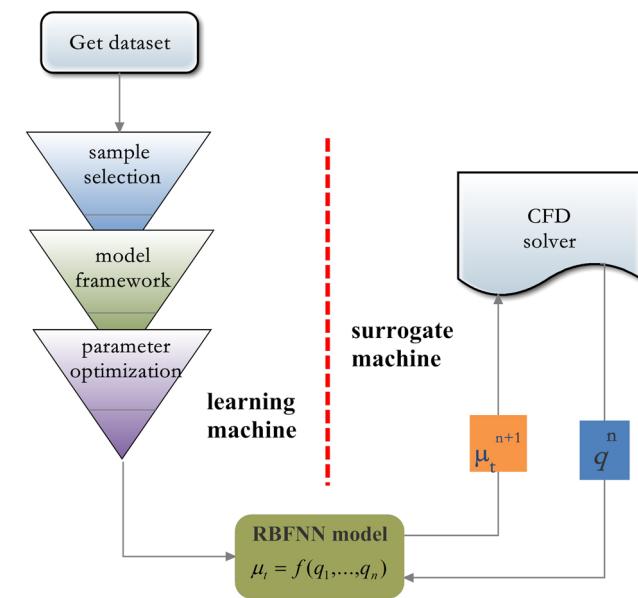


FIG. 1. Flow chart for building the learning machine and surrogate machine where q and μ_t mean the input features and eddy viscosity, respectively.

according to mean flow variables are calculated and passed to CFD solver; see Fig. 1.

B. Modeling strategy

The selection of modeling strategies depends on the specific problem. A typical strategy is the **local model** based on the grid topology, which approximates the output by the information of current and neighboring nodes. This strategy can only be applied to fixed grid topology and needs a large number of local models because the local model is ad hoc for each grid node. Another strategy computes the output according to the free stream condition like the Mach number and angle of attack, etc. This strategy constructs a mapping to modal coefficients by extracting some dominant modes. In essence, this **projection-based model** is a database between the free stream condition and output, which is hard to generalize to different geometries for lack of information of the flow field. Different from the above two strategies, the model construction process in this paper combines the free stream conditions and local mean variables as well as some location information together. The proposed model in our work is ad hoc for each zone. We divided the modeled flow field into four zones, and thus four individual models are constructed. In this way, compared with the local model, the amount of the proposed model is cut down dramatically but the dimension is increased to some degree. We tried to build a model with the acceptable dimension and the desirable generalization ability.

A high Reynolds number means a thin boundary layer. Thus, the gradient and data range of the eddy viscosity is very large along the normal direction of wall. As a consequence, modeling the eddy viscosity directly may lead to many outliers. To avoid this problem, we referred to the idea of Wang et al.²³ and Gamahara and Hattori³¹ and divided the whole flow field into different zones according to the normal distance from the wall. Furthermore, although the partition modeling method takes effect to a certain extent, it does not reflect the truth that the small errors of the eddy viscosity in the near wall region may lead to large discrepancies in the skin friction. In order to highlight the weight of the near wall region, an exponential function $F_s = e^{\sqrt{d/d_{min}}} - 2$ was introduced in the

TABLE I. The flow features used as a regression input where \mathbf{S} is the strain rate, $\boldsymbol{\Omega}$ the vorticity, α the angle of attack, and $\|\cdot\|$ the matrix norm. For convenience, the entropy was redefined as $S' = p/\rho^\gamma - 1$, ($\gamma = 1.4$) in this paper.

Feature	Description	Sign
1	Horizontal velocity	u
2	Density	ρ
3	Normal wall distance	d
4	Normal wall distance squared times the vorticity	$d^2 \Omega$
5	Exponential function	F_s
6	Projection of free stream to normal direction of streamline	$\text{sgn}(y)[-v + u \tan(\alpha)]$
7	Velocity direction	$\arctan[v/u]$
8	Entropy	S'
9	Normalized strain rate	$\ \mathbf{S}\ / (\ \mathbf{S}\ + \ \boldsymbol{\Omega}\)$

modeling process, where d_{min} is the minimum value of normal distance from the wall in a specific zone. Another method we tried but failed to handle the large data range is using logarithmic transformation before modeling and then making inverse transformation. But this will also make the error between the model output and true value magnified exponentially. For example, an output value of 100 is logarithmically transformed to 2. If the proposed model shows an error of 0.1 larger than 2, then the error is actually 25.9 after exponential transform. Our practice indicates that it is almost impossible to achieve very high accuracy for every sample. There are always some outliers emerged from where the model performance is poor, which reduce the smoothness of output space. Another advantage of partition modeling is that the features and model parameters can be tailored for different zones. Compared with the far field and wake region, the flow in the boundary layer is more intense and critical. Hence, more neurons in the hidden layer are necessary to capture the sharp changes in the turbulent boundary layer. In the wake region, the entropy can be used to reflect the distribution of eddy viscosity to some extent, so it can be selected as one of the features.³⁶

C. The artificial neural networks

Machine learning can be used to build models from datasets by some algorithms, which have the ability of judgment and prediction. As one typical algorithm of machine learning, neural networks are built according to the mutual connection among brain neurons. This technique has already been applied to the reduced-order models of nonlinear aerodynamic systems.^{37,38} Generally, compared with the compact

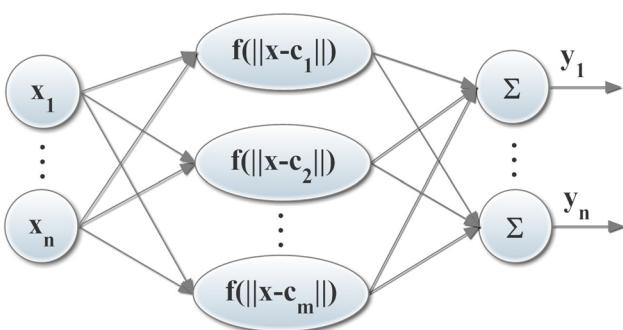
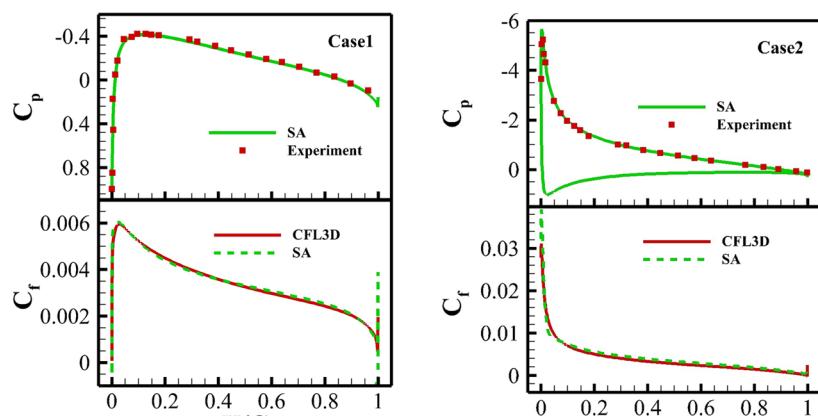


FIG. 2. Framework of one hidden layer neural network. The $x_1 \dots x_n$ and $y_1 \dots y_n$ mean the input layer and output layer, respectively; The $c_1 \dots c_m$ are the centers.

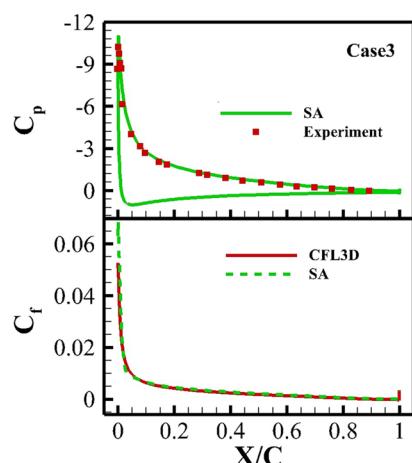
TABLE II. The airfoil flow conditions for validation.

Airfoil	Case	$\alpha(^{\circ})$	Re	Ma
NACA0012	Case 1	0	3×10^6	0.15
	Case 2	10	3×10^6	0.15
	Case 3	15	3×10^6	0.15
RAE2822	Case 9	2.8	6.5×10^6	0.73
	Case 10	2.8	6.2×10^6	0.75



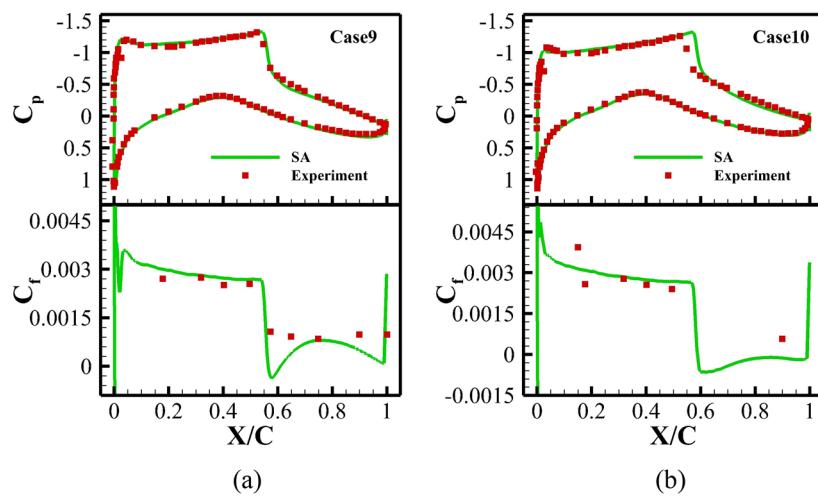
(a)

(b)



(c)

FIG. 3. The comparison of surface pressure (upper) and skin friction coefficient (lower) in different cases of NACA0012 airfoil.



(a)

(b)

FIG. 4. The comparison of surface pressure (a) and skin friction coefficient on upper surface (b) in different cases of RAE2822 airfoil.

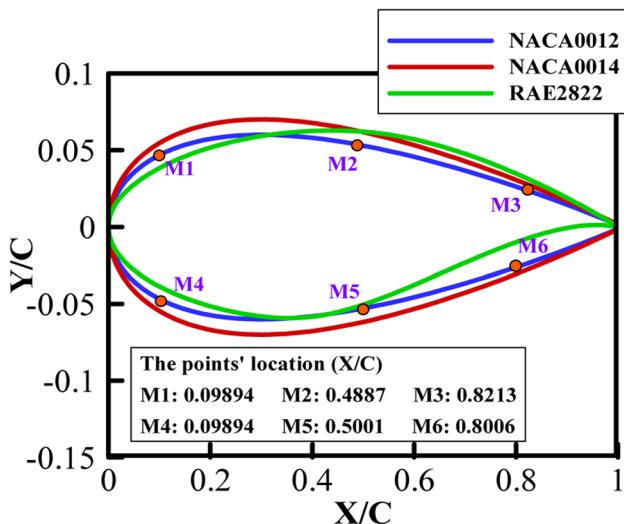


FIG. 5. The adopted airfoils for training and predicting the RBFNN model.

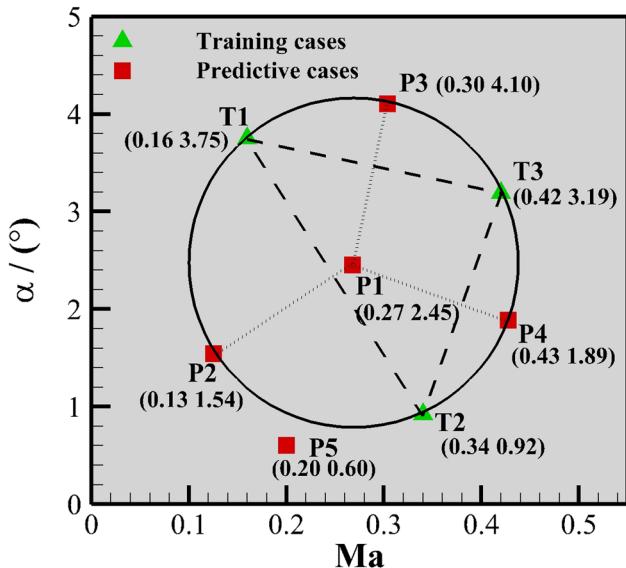


FIG. 6. The adopted training and predictive cases.

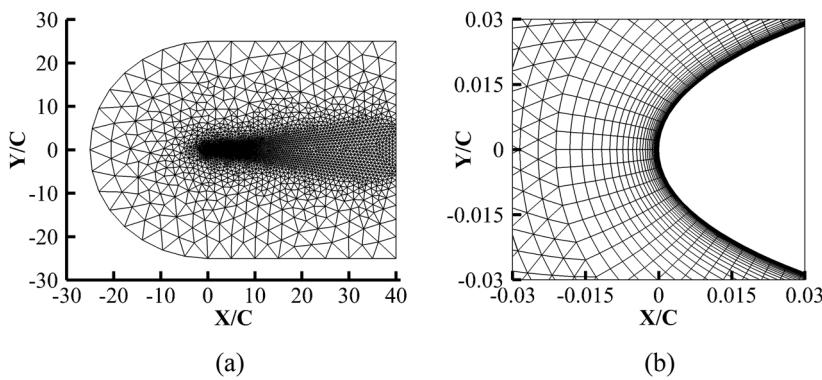


FIG. 7. The adopted mesh for NACA0012 airfoil.

model, more hidden layers and neuron units are beneficial to improve modeling results, but the risk of over-fitting and low generalization is also increasing. Considering the balance of accuracy and generalization, one hidden layer neural networks is adopted in this paper. The radial basis function was proved to be a great approximator³⁹ and has been applied to the PDEs solution,⁴⁰ flow field reconstruction,⁴¹ and model of nonlinear unsteady aerodynamics.⁴²⁻⁴⁴ A typical radial basis function neural networks consisting of one input layer, one hidden layer, and one output layer is shown in Fig. 2. The input layer consists of the set of input vectors $\{x_1 \dots x_n\}$. Each component of the input vector is called feature which represents an individual property. The hidden layer performs a non-linear transformation to map the input space to a new space. The non-linear transformation can be achieved by multiquadric function or Gaussian function or other functions.⁴⁵ Then the output layer combines this new space linearly with the parameters called weight. The i th output can be described by

$$y_i = \sum_{j=1}^m w_j \phi(\|x_i - c_j\|), \quad (2.1)$$

where w_j is the j th weight, ϕ is the non-linear function, c_j is the j th center, $\|\cdot\|$ denotes the Euclidean norm, and m is the number of centers. The Gaussian function was adopted as the non-linear function, which is described as follows:

$$\phi(x) = \exp(-\sigma \|x - c\|^2), \quad (2.2)$$

where σ is the width. From Eqs. (2.1) and (2.2), the model performance is determined by the neuron number in the hidden layer, the centers, the width, and the weight.⁴⁶ Generally, more neurons are helpful to improve model accuracy, but it is at the risk of overfitting and low generalizability.

In this paper, the input layer is formed by sample features which include free stream conditions, mean flow variables and derivatives, and so on; see Table I. Additionally, the effect of the neurons' number was also studied in our research by experiments with many trials and trade-offs between loss function and computation cost. The results indicate that 80 neurons are enough for reaching the optimal solution.⁴⁷ The output layer is the eddy viscosity of each sample. To get compact datasets, the whole data points were scaled linearly to $[-1, 1]$ before the training procedure.

For a dense grid, the adjacent cells might have very similar flow information. Thus, if each cell is taken as one sample, there will be many redundant samples for similar flow cases, which can increase the training time. In addition, the distribution of sample space is imbalanced due to different grid densities in the whole domain. Consequently, the model performance is inclined to those denser regions. Facing these two problems, this paper performed sample selection to decrease unnecessary samples, aiming at approximating the original sample space by less but more representative samples. Specifically, the algorithm is shown as follows:

1. Define the expected sample number K , the constant λ ($\lambda > 1$), and the initial relative distance δ . The sample group is denoted by S . Then choose a data point from the whole dataset T randomly as the first one of sample group, $k = 1$.
2. Compute the minimum relative distance between data point T_l and every sample S_m , i.e., $\min\left(\sum_{n=1}^N \left| \frac{T_{l,n} - S_{m,n}}{\max(T_{l,n}, S_{m,n})} \right|, m = 1, 2, \dots, K\right)$, where N is the feature number. If the minimum relative distance is larger than δ , then T_l is chosen as a new sample, $k = k + 1$; else discarded.
3. Recycle the last step for the whole data points in T .
4. If $k = K$, then stop; else $\delta = \delta / \lambda$, repeat from step 2.

D. Parameter optimization

Once the sample group is determined, the parameters of the center and width in each neuron can be chosen from the input vectors or obtained by various algorithms like gradient descent (GD), orthogonal least square, recurrent least square, and so on.^{48–50} The equations of gradient descent optimization algorithm are shown as follows:

$$\mathbf{c}_j^{n+1} = \mathbf{c}_j^n - \eta_c \cdot \nabla \mathbf{c}_j, \quad (2.3)$$

$$\sigma_j^{n+1} = \sigma_j^n - \eta_\sigma \cdot \nabla \sigma_j. \quad (2.4)$$

The learning ratios η_c and η_σ were both set to be 0.01 in this paper, and the gradients in above equations were calculated by

$$\nabla \mathbf{c}_j = -2w_j \sigma_j \sum_{i=1}^K e_i \Phi_{ij} (\mathbf{x}_i - \mathbf{c}_j), \quad (2.5)$$

$$\nabla \sigma_j = w_j \sum_{i=1}^K e_i \Phi_{ij} \|\mathbf{x}_i - \mathbf{c}_j\|^2, \quad (2.6)$$

where $\Phi_{ij} = \exp(-\sigma_j \|\mathbf{x}_i - \mathbf{c}_j\|^2)$.

To avoid unreasonable values of model parameters,⁵¹ the centers were limited in the range of sample space during the optimization process, and the width is assigned as 0.01 if negative. The optimal weight can be obtained by GD or pseudo-inverse as following:

$$\mathbf{w}^{n+1} = (\Phi^T \Phi + \lambda (|\mathbf{w}|)^\dagger)^{-1} \Phi^T \mathbf{y}, \quad (2.7)$$

where $|\mathbf{w}|$ is a diagonal matrix with diagonal element $|w_1|, |w_2|, \dots, |w_M|$ and \dagger is the generalized inverse.

For the multi-extremum problem, the optimum of the GD method is influenced by the initial value and falls into the local minimum easily. In order to contain more information, the clustering result is taken as the initial value. Clustering is a method used to divide the dataset into several disjoint subsets. There are various clustering methods, and

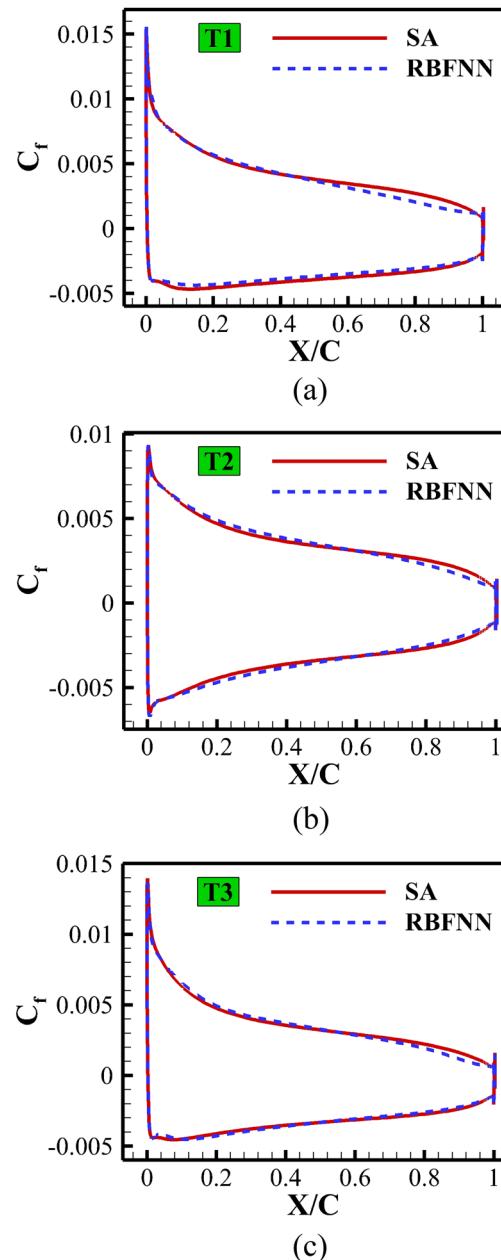


FIG. 8. Comparison of the skin friction coefficient.

the distance calculating formula is dependent on the specific problem.^{52–55} For less similarity between the model centers, the K-means clustering in Refs. 56 and 57 was used in this paper.

The loss function is the objective function for parameter optimization, like L1 (one norm) loss function, L2 (square norm) loss function, etc. It is found that although the L2 loss function can achieve higher training accuracy, the convergence of the Navier-Stokes equation is difficult to be guaranteed after the model is embedded in the CFD solver.

Furthermore, to put less weight on outlier points,⁵⁸ the L1 constraint is adopted. The final objective function is as follows;

$$L = \frac{1}{2} \sum_{i=1}^K (\bar{y}_i - y_i)^2 + \lambda \sum_{j=1}^N |w_j|, \quad (2.8)$$

where \bar{y}_i and y_i are the true value and model value of the i th sample, respectively; λ is the regulation constant set to be 0.01–0.05 in this paper.

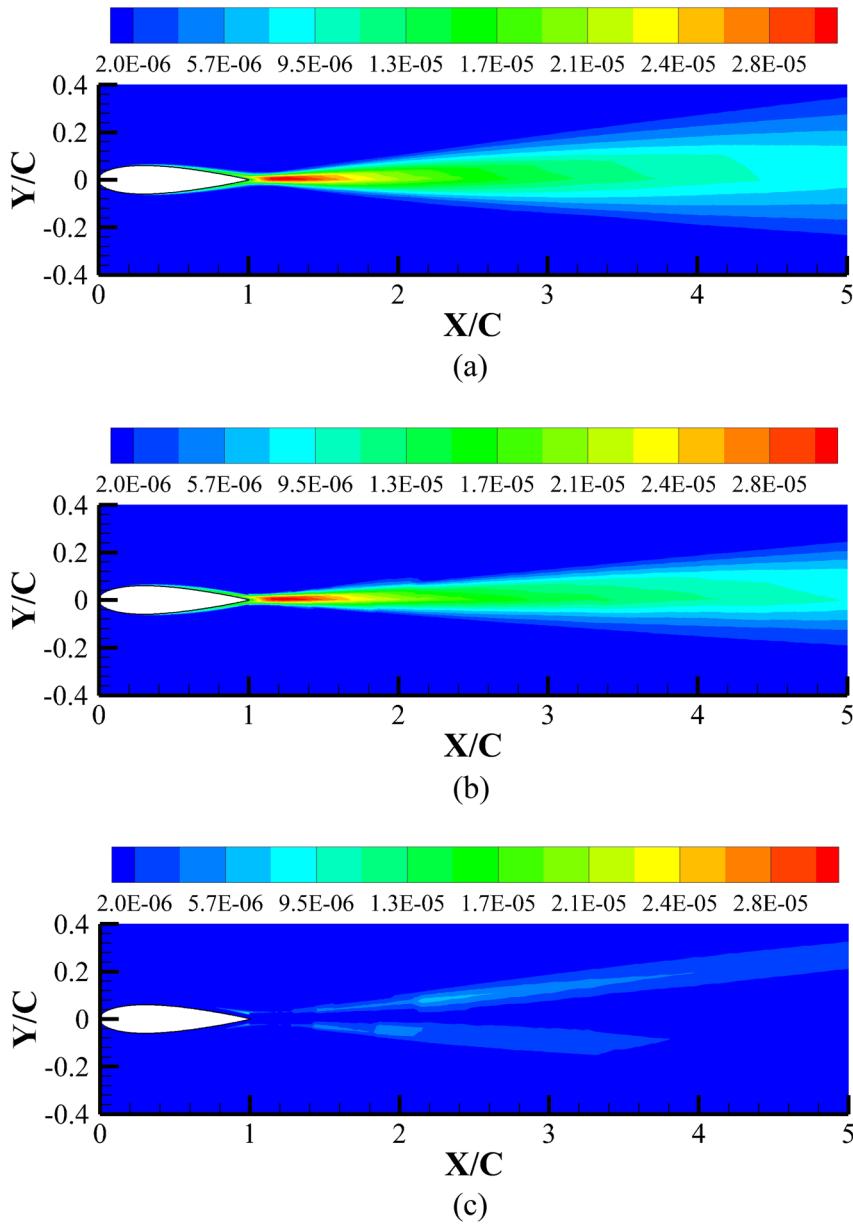


FIG. 9. The contour of eddy viscosity at T2 case calculated by (a) SA model (b) RBFNN model, and (c) the error contour.

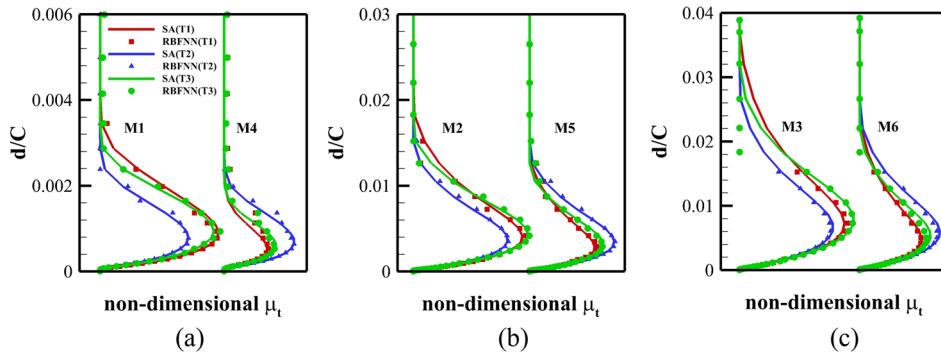


FIG. 10. The eddy viscosity profile of training cases at monitoring points along the normal direction of wall; the panels (b) and (c) share the same legend as (a).

III. EXAMPLES AND ANALYSIS

A. Code validation

The validation of the SA model is performed for subsonic flow around NACA0012 airfoil⁵⁹ and transonic flow around RAE2822 airfoil.⁶⁰ The specific free stream conditions are shown in Table II. The hybrid grid is adopted with the height of first grid $d_{y^+<1} = 4.0 \times 10^{-6}$.

CFL3D⁶¹ is a structure-grid CFD code including various turbulence models, which has been used by many academic researchers. The flow solver with a cell-centered finite volume scheme used in our work is also validated by CFL3D besides the experimental results. The pressure coefficient C_p and skin friction coefficient C_f around NACA0012 airfoil at three angles of attack agree well with the experimental results and CFL3D results, respectively; see Fig. 3. Figure 4 shows the

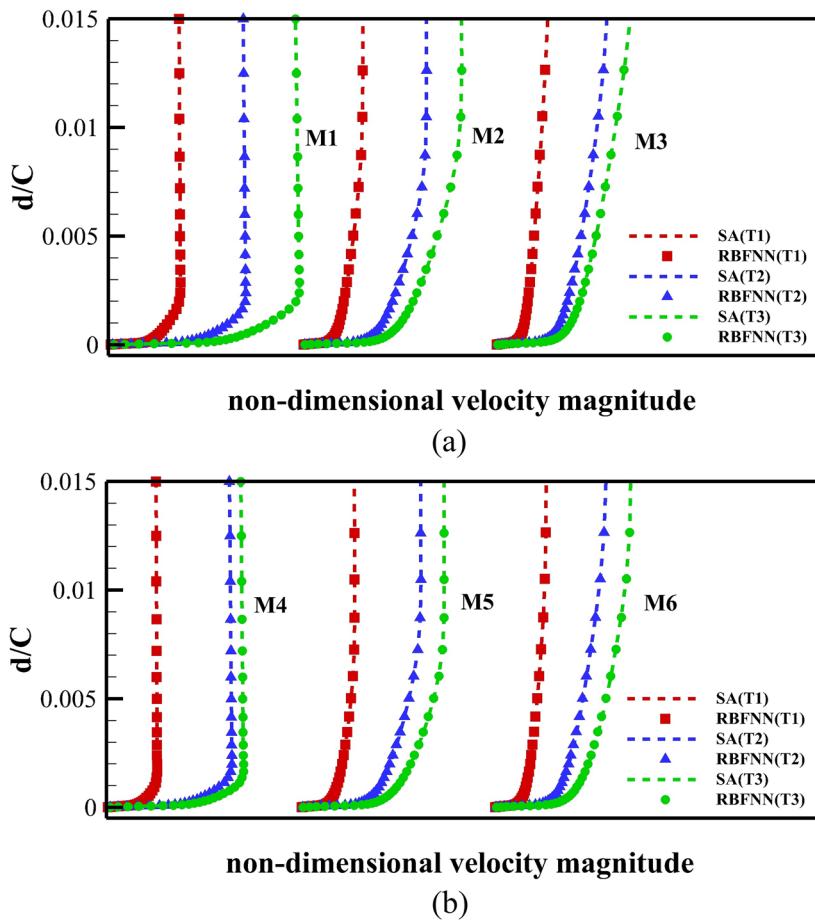


FIG. 11. The velocity magnitude profile of training cases at monitoring points along the normal direction of wall (a) upper surface (b) lower surface.

comparison of case 9 and case 10 with the experimental results around RAE2822 airfoil. In case 10, the shockwave location moved downstream slightly. It should be emphasized that, the skin friction coefficient in Fig. 4(b) is non-dimensionalized by the local boundary layer free stream condition rather than the incoming flow condition.⁶²

B. Result

Similar to Ref. 14 the results calculated by the CFD solver with the SA model⁶³ were regarded as the true value in this paper. Subsonic steady flows around NACA0012 airfoil, NACA0014 airfoil, and RAE2822 airfoil were investigated at fixed Reynolds number $Re = 3 \times 10^6$. A wall with a no-slip condition is set up for the airfoil surface, and a pressure-far-field condition is imposed on the far field. The results shown below mainly include the skin friction coefficient, velocity, and eddy viscosity profiles along the normal direction of wall and eddy viscosity contour. Specifically, six monitoring locations are selected from both the upper and lower surface of airfoil, which are $X/C = 0.09894$, $X/C = 0.4887$, $X/C = 0.8213$ on the upper surface and $X/C = 0.09894$, $X/C = 0.5001$, $X/C = 0.8006$ on the lower surface; see Fig. 5.

Three training cases of NACA0012 airfoil were sampled through the Latin hypercube sampling (LHS) method with the Mach number (Ma) from 0.1 to 0.5 and the angle of attack (α) from 0° to 5° . The predicting cases were chosen as the circumcenter P1 and the intersections of circumcircle with three perpendicular bisectors P2-P4 as well as P5 outside the circumcircle; see Fig. 6. The predicting examples include both interpolation and extrapolation besides two different airfoils, by which the generalization of proposed model is demonstrated.

The hybrid grid was adopted, of which the first layer height in boundary layer is $d_{y^+<1} = 8.0 \times 10^{-6}$ and the increasing rate is 1.2. Taking NACA0012 airfoil as an example, the whole domain and the local grid near the leading edge are shown in Figs. 7(a) and 7(b), respectively.

1. The training cases

The skin friction coefficient of three training cases is almost identical with the true value, with slight error on the lower surface and the rear part of upper surface; see Fig. 8. Taking T2 case as the example, the eddy viscosity contour agrees well with the true value except for the wake region; see Fig. 9. As for the eddy viscosity profile, the agreement is good at the locations in the front and middle of the airfoil, see Figs. 10(a) and 10(b), but near the trailing edge, some obvious deviations can be observed if the wall distance d/C is larger than 0.018; see Fig. 10(c). In this region, poor agreement is mainly caused by low sensitivity of input to output. Specifically, the change of mean flow variables is flat, while the change of the eddy viscosity is still sharp along the normal direction of wall. But these large discrepancies have little impact on the velocity profile; see Fig. 11.

2. The predicting cases

a. Part I NACA0012 airfoil. The skin friction coefficient of five predicting cases also shows good agreement except P2 case which has the largest deviation from training cases; see Fig. 12. The agreement of contour of the P4 case and eddy viscosity profiles is similar as the training cases; see Figs. 13 and 14. Excellent agreement of the velocity profiles along the normal direction of the wall at monitoring locations is shown

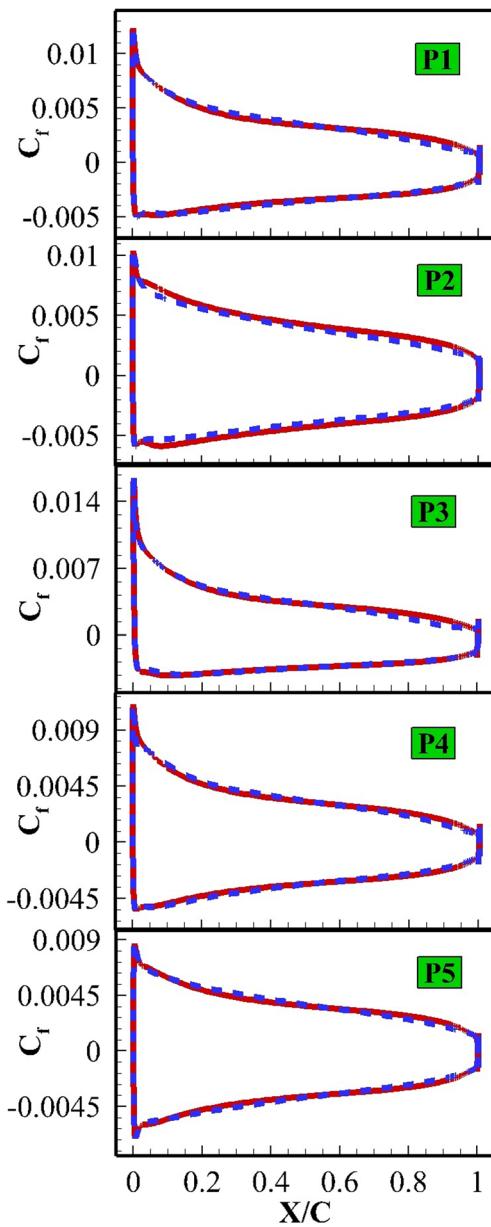


FIG. 12. Comparison of SA (solid red line) and RBFNN (dashed blue line) for predicting cases.

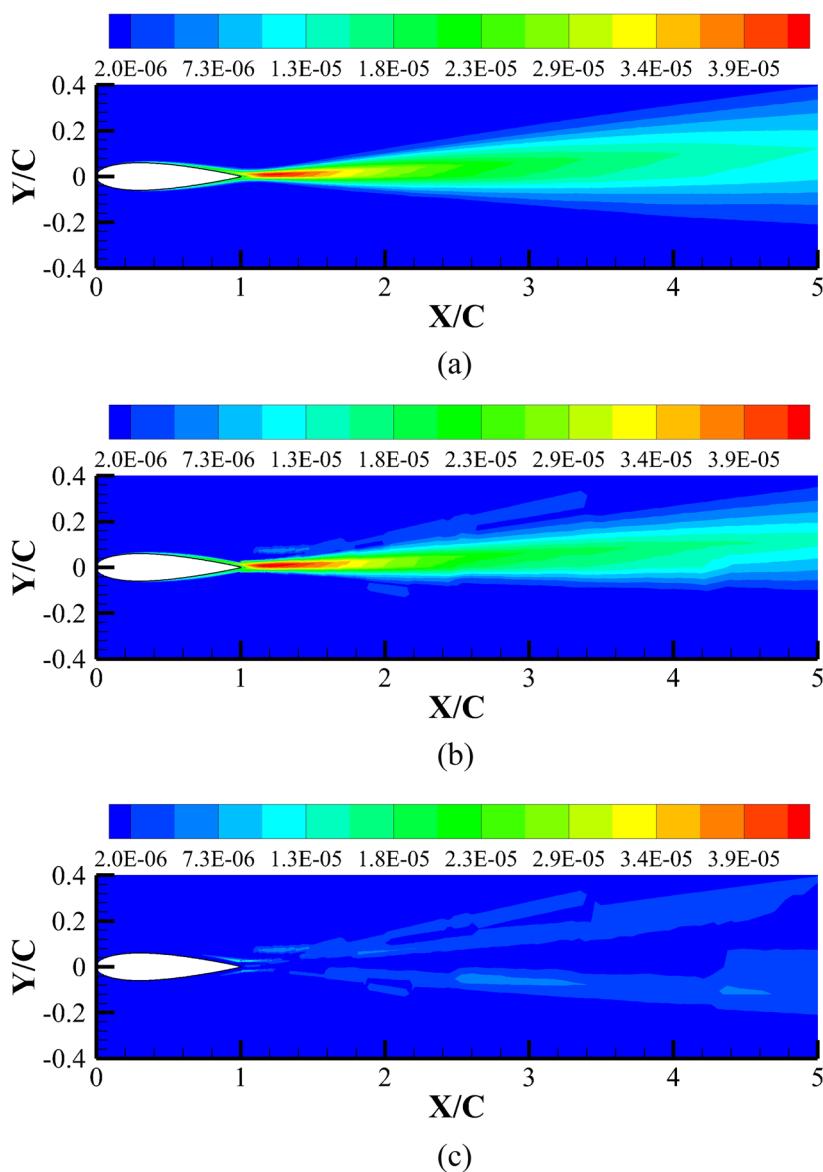


FIG. 13. The contour of eddy viscosity of P4 calculated by (a) SA model, (b) RBFNN model, and (c) the error contour.

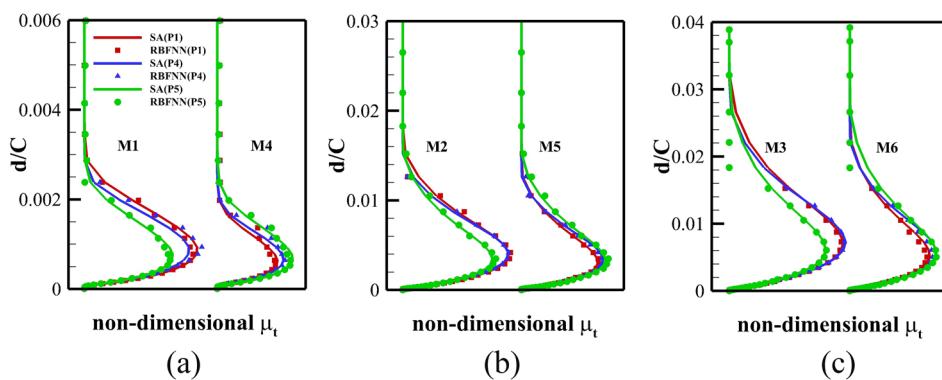


FIG. 14. The eddy viscosity profile of three predicting cases at monitoring points along the normal direction of wall; the panels (b) and (c) share the same legend as (a).

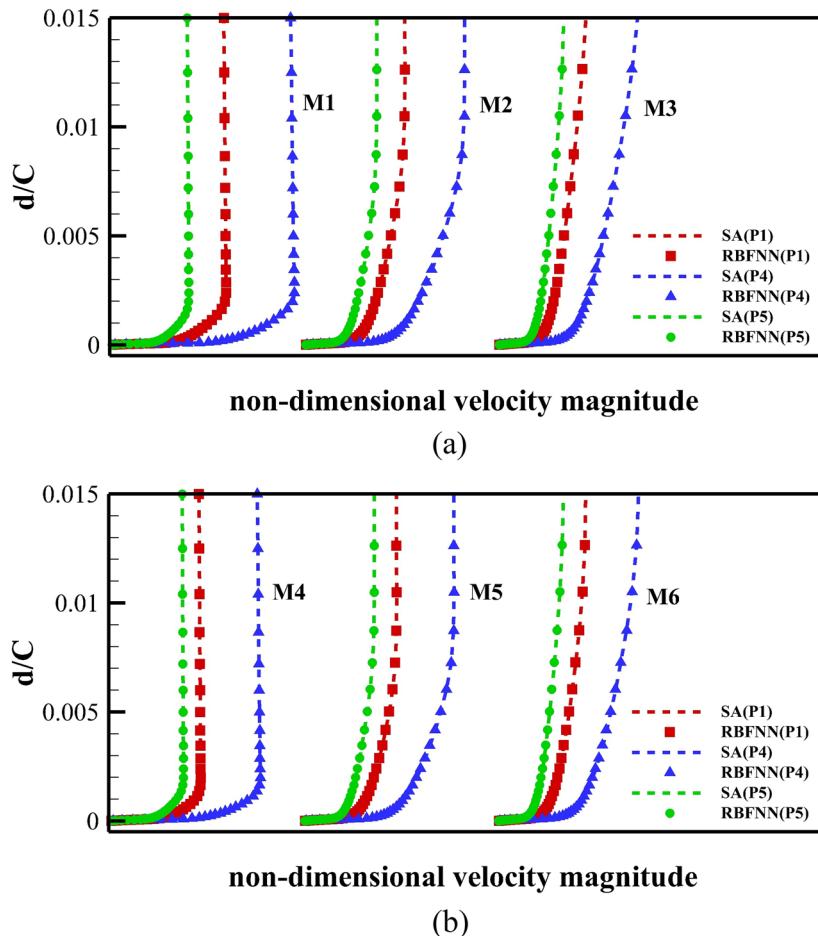


FIG. 15. The velocity magnitude profile of three predicting cases at monitoring points along the normal direction of wall (a) upper surface and (b) lower surface.

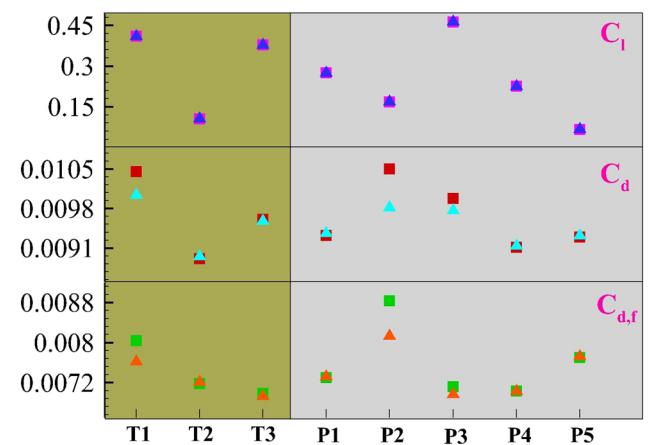


FIG. 16. Comparison of SA (square) and RBFNN (delta) for both training and predicting cases.

in Fig. 15. Both the lift coefficient C_l and drag coefficient C_d are shown in Fig. 16, with agreement of the lift coefficient almost identical. The mean relative error of the drag coefficient is only 1.79%, and the maximum corresponding to the P2 case

is 0.0007, which is basically caused by the error of the friction drag coefficient $C_{d,f}$. The generalizability of the model stems mainly from two aspects. One the one hand, although the training dataset consists of only three cases, in fact, each case contains abundant local flow information, which ensures the diversity of training samples. Furthermore, the proposed model is a global model which fits the output of a single sample by the whole information rather than the neighboring information like the local model. On the other hand, there is no qualitative difference between training and predicting cases. All cases have neither shock waves nor separations. Therefore, the proposed model shows good agreement when generalized to both interpolated and extrapolated cases.

b. Part II NACA0014 airfoil and RAE2822 airfoil. In this part, NACA0014 airfoil and RAE2822 airfoil were adopted to test the generalization of the proposed approach to different airfoil shapes. Considering both the interpolation and extrapolation, P1, P2, and P5 were selected as the computing cases. The results show that the skin friction coefficients are in good agreement except the P2 case; see Fig. 17. Although there are sharp shifts of residual during the computation process, the

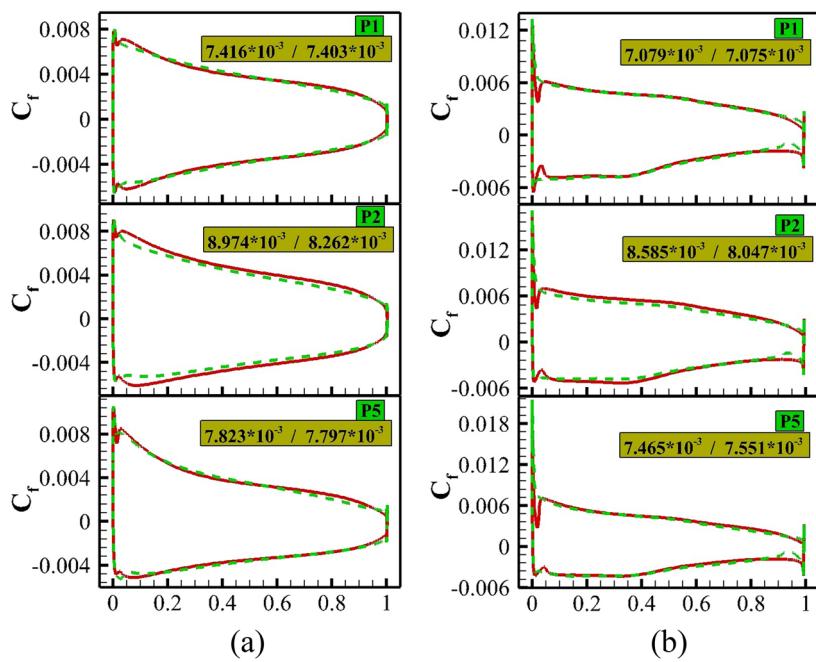


FIG. 17. Predictions for NACA0014 (a) and RAE2822 (b) airfoil at P1, P2, and P5 cases. The data inside are $C_{d,f}$ values calculated by the SA/RBFNN model.

CFD solver embedded with the present model still achieved satisfying convergence. The residual evolution of P5 case is shown in Fig. 18.

High efficiency is also one of the targets in our work. The one hidden layer neural network is a concise framework without solving transport equations. We listed the computing time of five predicting cases about NACA0012 airfoil and three predicting cases about NACA0014 airfoil and RAE2822 airfoil. It should be noted that the model training time is not included.

TABLE III. Comparison of turbulence model's computing time as the residual was down to $O(10^{-9})$. The black, green, and blue data are corresponding to NACA0012, NACA0014, and RAE2822 airfoil, respectively.

Computing time (s)	SA model	RBFNN
P1	1975.5/1135.8/1519.8	1111.5/680.5/867.2
P2	858.9/430.7/984.5	717.82/641.1/738.6
P3	1995.6	760.2
P4	1864.3	714.4
P5	1708.1/971.6/1349.8	693.50/693.0/787.7

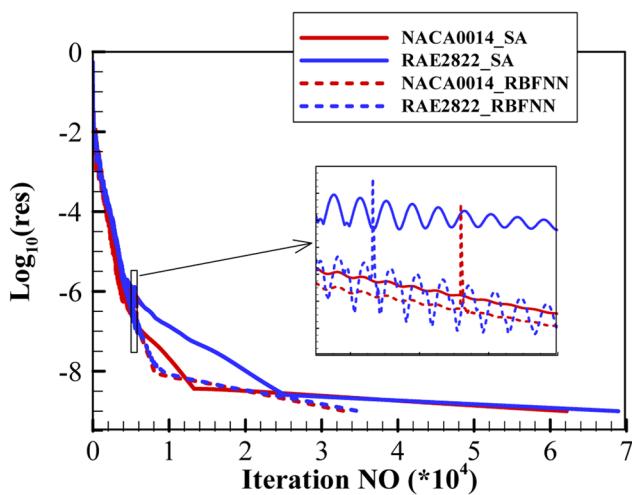


FIG. 18. Residual evolution at P5 case.

For nearly all the cases, the proposed approach is more efficient, especially for those flow cases with better accuracy; see Table III.

IV. CONCLUSIONS AND FUTURE WORK

Based on three training cases of turbulent flows around NACA0012 airfoil, this paper constructs an eddy viscosity mapping function by the radial basis function neural network for subsonic attached flows. By comparing the proposed approach with the original SA model, the accuracy and generalization capability to different airfoils and flow cases are validated. The conclusions are stated as follows:

- (1) By partition and building the model separately, the outliers caused by large data range can be decreased effectively, which is helpful to obtain satisfied accuracy in vital domains. Coupled with Navier-Stokes equations, the proposed approach also achieves the final convergence.

- (2) The present model is a global model with appropriate dimensions. Driven by only three training cases, the proposed model achieved reasonable accuracy and enough generalization to other subsonic cases and airfoils. For both the training cases and predicting cases, the velocity profile and the skin friction distribution agree well with the results of the original SA model, which demonstrates the promising prospect of machine learning methods in future studies about turbulence modeling.
- (3) The proposed approach is more efficient than the original SA model. On the one hand, the one-hidden layer neural networks with about a hundred neurons are a concise framework without complex calculations. On the other hand, less iteration steps are needed for achieving the final convergence standard.

This paper is still a preliminary work toward modeling high Reynolds number turbulent flows with data-driven methods. Separated flows and other more complex turbulent flows will be further investigated in future studies.

ACKNOWLEDGMENTS

This paper was mainly supported by the National Science Fund of China (Grant Nos. 91852115 and 11622220) and 111 Project of China (Grant No. B17037).

REFERENCES

- ¹J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, "CFD vision 2030 study: A path to revolutionary computational aerosciences," Technical Report NASA/CR-2014-218178, 2014.
- ²P. A. Durbin, "Some recent developments in turbulence closure modeling," *Annu. Rev. Fluid Mech.* **50**, 77–103 (2018).
- ³B. S. Baldwin and H. Lomax, "Thin-layer approximation and algebraic model for separated turbulent flows," AIAA Paper No. 78-257, 1978.
- ⁴S. B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).
- ⁵J. Mollicone, F. Battista, P. Gualtieri, and C. Casciola, "Effect of geometry and Reynolds number on the turbulent separated flow behind a bulge in a channel," *J. Fluid Mech.* **823**, 100–133 (2017).
- ⁶J. Mollicone, F. Battista, P. Gualtieri, and C. Casciola, "Turbulence dynamics in separated flows: The generalised Kolmogorov equation for inhomogeneous anisotropic conditions," *J. Fluid Mech.* **841**, 1012–1039 (2018).
- ⁷Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *IEEE International Conference on Acoustics (IEEE, 2017)*, pp. 4845–4849.
- ⁸T. Y. Lin, A. Roychowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**, 1309–1322 (2018).
- ⁹X. Wen, Y. Z. Liu, Z. Y. Li, Y. J. Chen, and D. Peng, "Data mining of a clean signal from highly noisy data based on compressed data fusion: A fast-responding pressure-sensitive paint application," *Phys. Fluids* **30**, 097103 (2018).
- ¹⁰J. Q. Kou and W. W. Zhang, "A hybrid reduced-order framework for complex aeroelastic simulations," *Aerospace Sci. Technol.* **84**, 880–894 (2019).
- ¹¹M. Milano and P. Koumoutsakos, "Neural network modeling for near wall turbulent flow," *J. Comput. Phys.* **182**, 1–26 (2002).
- ¹²M. Hocevar, B. Sirok, and I. Grabec, "A turbulent wake estimation using radial basis function neural networks," *Flow, Turbul. Combust.* **74**, 291–308 (2005).
- ¹³K. Duraisamy, P. R. Spalart, and C. L. Rumsey, "Status, emerging ideas and future directions of turbulence modeling research in aeronautics," Report No. NASA/TM-2017-219682, 2017.
- ¹⁴B. D. Tracey, K. Duraisamy, and J. J. Alonso, "A machine learning strategy to assist turbulence model development," AIAA Paper 2015-1287, 2015.
- ¹⁵K. Duraisamy, Z. J. Zhang, and A. P. Singh, "New approaches in turbulence and transition modeling using data-driven techniques," AIAA Paper 2015-1284, 2015.
- ¹⁶A. P. Singh, S. Medida, and K. Duraisamy, "Machine learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA J.* **55**, 2215–2227 (2017).
- ¹⁷X. W. Jin, P. Cheng, W. L. Chen, and H. Li, "Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," *Phys. Fluids* **30**, 047105 (2018).
- ¹⁸A. P. Singh and K. Duraisamy, "Using field inversion to quantify functional errors in turbulence closures," *Phys. Fluids* **28**, 045110 (2016).
- ¹⁹K. Duraisamy, A. P. Singh and S. Pan, "Augmentation of turbulence models using field inversion and machine learning," AIAA Paper 2017-0993, 2017.
- ²⁰E. J. Parish and K. Duraisamy, "A paradigm for data-driven predictive modeling using field inversion and machine learning," *J. Comput. Phys.* **305**, 758–774 (2016).
- ²¹J. X. Wang, J. L. Wu, and H. Xiao, "Physics informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," *Phys. Rev. Fluids* **2**, 034603 (2017).
- ²²H. Xiao, J. L. Wu, J. X. Wang, and E. G. Paterson, "Physics informed machine learning for predictive turbulence modeling: Progress and perspectives," AIAA Paper No. 2017-1712, 2017.
- ²³J. X. Wang, J. L. Wu, and H. Xiao, "Physics informed machine learning for predictive turbulence modeling: Toward a complete framework," in *Proceedings of the Summer Program* (Stanford University, 2016).
- ²⁴J. L. Wang, "Physics-informed, data-driven framework for model-form uncertainty estimation and reduction in RANS simulations," Ph.D. dissertation (Virginia Tech, 2017).
- ²⁵J. L. Wu, J. X. Wang, H. Xiao, and J. Ling, "A priori assessment of prediction confidence for data-driven turbulence modeling," *Flow, Turbul. Combust.* **99**, 25–46 (2017).
- ²⁶C. X. He, Y. Z. Liu, and L. Gan, "A data assimilation model for turbulent flows using continuous adjoint formulation," *Phys. Fluids* **30**, 105108 (2018).
- ²⁷J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," *J. Comput. Phys.* **318**, 22–35 (2016).
- ²⁸J. Ling and J. Templeton, "Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty," *Phys. Fluids* **27**, 085103 (2015).
- ²⁹J. Ling, A. Kuzawski, and J. Templeton, "Reynolds averaged turbulence modeling using deep neural networks with embedded invariance," *J. Fluid Mech.* **807**, 155–166 (2016).
- ³⁰J. Ling, "Using machine learning to understand and mitigate model form uncertainty in turbulence models," in *IEEE International Conference on Machine Learning and Applications (IEEE, 2015)*, pp. 813–818.
- ³¹M. Gamabara and Y. Hattori, "Searching for turbulence models by artificial neural network," *Phys. Rev. Fluids* **2**(5), 054604 (2017).
- ³²K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annu. Rev. Fluid Mech.* **51**, 357 (2018).
- ³³Z. She, X. Chen, and F. Hussain, "Quantifying wall turbulence via a symmetry approach: A lie group theory," *J. Fluid Mech.* **827**, 322–356 (2017).
- ³⁴X. Chen, F. Hussain, and Z. S. She, "Quantifying wall turbulence via a symmetry approach. Part 2. Reynolds stresses," *J. Fluid Mech.* **850**, 401–438 (2018).
- ³⁵K. J. Nathan, "Deep learning in fluid dynamics," *J. Fluid Mech.* **814**, 1–4 (2017).
- ³⁶J. S. Brock, "A modified Baldwin-Lomax turbulence model for turbomachinery wakes," M.S. thesis, Virginia Tech, 1991, <https://vtworks.lib.vt.edu/handle/10919/44534>.

- ³⁷R. Huang, H. Li, and Y. Zhao, "Nonlinear reduced-order modeling for multiple-input/multiple-output aerodynamic systems," *AIAA J.* **52**, 1219–1231 (2014).
- ³⁸R. Huang, H. Li, H. Hu et al., "Open/closed-loop aeroservoelastic predictions via nonlinear, reduced-order aerodynamic models," *AIAA J.* **53**, 1812–1824 (2015).
- ³⁹E. J. Hartman, J. D. Keeler, and J. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximators," *Neural Comput.* **2**, 210–215 (1990).
- ⁴⁰C. Shu, H. Ding, and K. S. Yeo, "Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier Stokes equations," *Comput. Methods Appl. Mech. Eng.* **192**, 941–954 (2003).
- ⁴¹Y. L. Liu, W. W. Zhang, Y. W. Jiang et al., "A high-order finite volume method on unstructured grids using RBF reconstruction," *Comput. Math. Appl.* **72**, 1096–1117 (2016).
- ⁴²W. W. Zhang, J. Q. Kou, and Z. Y. Wang, "Nonlinear aerodynamic reduced-order model for limit-cycle oscillation and flutter," *AIAA J.* **54**, 3304–3312 (2016).
- ⁴³W. W. Zhang, B. B. Wang, and Z. Y. Ye, "Efficient method for limit cycle flutter analysis by nonlinear aerodynamic reduced-order models," *AIAA J.* **50**, 1019–1028 (2012).
- ⁴⁴J. Q. Kou and W. W. Zhang, "Nonlinear aerodynamic reduced-order modeling for aeroelasticity with varying Mach numbers," *J. Aerosp. Eng.* **31**, 04018105 (2018).
- ⁴⁵M. J. L. Orr, Introduction to radial basis function networks, 1996, <https://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf>.
- ⁴⁶J. K. Kou and W. W. Zhang, "Layered reduced-order models for nonlinear aerodynamics and aeroelasticity," *J. Fluids Struct.* **68**, 174–193 (2017).
- ⁴⁷M. Y. El-Bakry, "Radial basis function neural networks model for mean velocity and vorticity of capillary flow," *Int. J. Numer. Methods Fluids* **67**, 1283–1290 (2011).
- ⁴⁸S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks* **2**, 302–309 (1991).
- ⁴⁹J. B. Gomm and D. L. Yu, "Selecting radial basis function network centers with recursive orthogonal least squares training," *IEEE Trans. Neural Networks* **11**, 306–313 (2000).
- ⁵⁰F. Schwenker, H. A. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural Networks* **14**, 439–458 (2001).
- ⁵¹M. K. Muezzinoglu and J. M. Zurada, "Projection-based gradient training of radial basis function networks," in 2004 IEEE International Joint Conference on Neural Networks (IEEE, 2004), pp. 1297–1302.
- ⁵²A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.* **31**, 651–666 (2010).
- ⁵³K. A. A. Nazeer and M. P. Sebastian, *Improving the Accuracy and Efficiency of the K-Means Clustering Algorithm* (WCE, 2009), ISBN: 978-988-17012-5-1.
- ⁵⁴A. Likas, N. Vlassis, and J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.* **36**(2), 451–461 (2003).
- ⁵⁵Z. H. Zhou, *Machine Learning* (Tsinghua University Press, 2016).
- ⁵⁶M. H. Dunham, *Data Mining—Introductory and Advanced Concepts* (Pearson Education, 2006).
- ⁵⁷R. Savitha, S. Suresh, and N. Sundararajan, "A fully complex-valued radial basis function network and its learning algorithm," *Int. J. Neural Syst.* **19**, 253–267 (2009).
- ⁵⁸P. H. James, A. S. James, and J. H. John, "A neural network for L1 norm linear regression," *Proc. SPIE* **1196**, 178–188 (1990).
- ⁵⁹D. C. Jespersen, T. H. Pulliam, and M. L. Childs, "Overflow turbulence modeling resource validation results," NASA Technical Report NAS-2016-01, 2016.
- ⁶⁰B. Einfeld, "Implementation of Reynolds stress models into the DLR-flower code," IB 124-2004/3, Institute of Aerodynamics and Flow Technology, 2004.
- ⁶¹S. L. Krist, R. T. Biedron, and C. L. Rumsey, CFL3D User's Manual (Version 5.0), NASA TM 208444, NASA Langley Research Center, 1998.
- ⁶²J. E. Bardina, P. G. Huang, and T. J. Coakley, *Turbulence Modeling Validation, Testing, and Development* (NASA Technical Memorandum, 1997), p. 110446.
- ⁶³P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," AIAA Paper No. 92-0439, 1992.