An airfoil geometric-feature extraction and discrepant data fusion learning method

Yu Xiang

University of Electronic Science and Technology of China <https://orcid.org/0000-0001-9622-7661>

Liwei Hu

University of Electronic Science and Technology of China

Jun Zhang

University of Electronic Science and Technology of China

Wenyong Wang (✉ wangwy@uestc.edu.cn)

University of Electronic Science and Technology of China

Research

Keywords: Airfoils, Manifold, Geometric-feature fusion, GF-Encoder

Posted Date: January 7th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1229857/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

RESEARCH

An airfoil geometric-feature extraction and discrepant data fusion learning method

Yu Xiang, Liwei Hu, Jun Zhang and Wenyong Wang*

*Correspondence:
wangwy@uestc.edu.cn
School of Computer Science and
Engineering, University of
Electronic Science and Technology
of China, Chengdu, China
Full list of author information is
available at the end of the article

Abstract

The perception of geometric-features of airfoils is the basis in aerodynamic area for performance prediction, parameterization, aircraft inverse design, etc. There are three approaches to percept the geometric shape of airfoils, namely manual design of airfoil geometry parameters, polynomial definition and deep learning. The first two methods directly define geometric-features or polynomials of airfoil curves, but the number of extracted features is limited. Deep learning algorithms can extract a large number of potential features (called latent features). However, the features extracted by deep learning lack explicit geometrical meaning. Motivated by the advantages of polynomial definition and deep learning, we propose a geometric-feature extraction method (named Bézier-based feature extraction, BFE) for airfoils, which consists of two parts: manifold metric feature extraction and geometric-feature fusion encoder (GF encoder). Manifold metric feature extraction, with the help of the Bézier curve, captures manifold metrics (a sort of geometric-features) from tangent space of airfoil curves, and the GF-encoder combines airfoil coordinate data and manifold metrics together to form novel fused geometric-features. To validate the feasibility of the fused geometric-features, two experiments based on the public UIUC airfoil dataset are conducted. Experiment I is used to extract manifold metrics of airfoils and export the fused geometric-features. Our proposed BFE and a classical Auto-Encoder are compared to verify whether smooth and realistic airfoils can be generated by the fused geometric-features. Experiment II, based on the Multi-task learning (MTL), is used to fuse the discrepant data (i.e., the fused geometric-features and the flight conditions) to predict the aerodynamic performance of airfoils. The results show that the BFE can generate more smooth and realistic airfoils than Auto-Encoder, and the fused geometric-features extracted by BFE can be used to reduce the prediction errors of C_L and C_D .

Keywords: Airfoils; Manifold; Geometric-feature fusion; GF-Encoder

Introduction

The geometry shape of airfoils greatly affects the aerodynamic performance, the parameterization as well as the aircraft inverse design [1, 2]. One of a traditional airfoil design method is to define airfoil geometry parameters manually, which is effective to perceive the variations in airfoil geometry structures [3]. However, this approach always fails on complex airfoils, that causes limited applications. Polynomial definition is an alternative efficient mathematical approach to approximate airfoil curves, for examples, Bézier curve [4] and B-spline [5], etc. This approach usually employ linear combinations of high degree polynomials to approximate airfoils to describe the geometry variation of airfoil structures. Nevertheless, polynomial approaches

can only give approximate expressions of airfoils and they are not powerful enough to deeply exploit features of airfoils from different aspects. Therefore it is difficult for polynomials to describe airfoils comprehensively.

In recent years, deep learning has achieved great success in feature extraction, for example Auto-Encoder [6], generative adversarial networks (GANs)[7, 8], convolution neural networks (CNNs)[9] and Multi-task learning (MTL) [10], etc. These methods conventionally take coordinates (X, Y) or pictures of airfoils as input. The problems existed in deep learning are: 1) the extracted features are lacking of explicit geometrical meaning; 2) some latent geometric-features cannot be further explored.

Motivated by deep learning and inspired by the Bézier theory, we propose a geometric-feature extraction method (named Bézier-based feature extraction, BFE) for feature extraction of airfoils. The BFE consists of two parts, namely manifold metric feature extraction and geometric-feature fusion encoder (GF-Encoder). In the manifold metric feature extraction, a 3-degree self-intersection-free Bézier curve is employed to build an Bézier manifold from airfoil coordinates [11]. In the constructed Bézier manifold space, the manifold metrics (as a sort of geometric-feature) are calculated with the inner-product of two perpendicular vectors. The GF-Encoder is composed of three components: an encoder, a fusion network and a Decoder. The encoder consists of a CNN which take airfoil coordinate data as input and a fully connected network (FCN) which take manifold metrics as input. Airfoil coordinate data and manifold metrics are encoded into fused geometric-features by the fusion network. The decoder, which take the fused geometric-features as inputs, consists of a deconvolution neural network (DeCNN) and another FCN. The output of the DeCNN is generated airfoil coordinate data, and the output of FCN in this decoder is generated manifold metrics. The fused geometric-feature is the final feature that represents the geometry of an airfoil.

To summarize, the contributions to our work are:

- 1 We prove that the self-intersection-free Bézier curve, as also a smooth manifold, its manifold metric can be used as a proper geometric-feature for representing airfoils;
- 2 we propose a geometric-feature extraction method named BFE to extract geometric-features from both Euclidean space and manifold space and then fuse these geometric-features together.
- 3 we propose a geometric-feature fusion encoder (GF-Encoder) to integrate airfoil coordinate data and manifold metrics together to form fused geometric-features;

The structure of the remainder of this paper is as follows. Related work section introduces the research status of polynomial definition and deep learning in the field of airfoil feature extraction. In methodology section, the details of our proposed BFE are elaborated. In experiments and results section, a public UIUC airfoil dataset is applied to validate the effectiveness of BFE and the feasibility of fused geometric-features in terms of predicting C_L and C_D of airfoils. The conclusions of our work are shown in conclusion section.

Related Works

In this section, we introduce current research status on polynomial definition and deep learning in the field of airfoil feature extraction.

The Bézier curve, B-spline and NURBS are typical polynomials that were used to derive equations of airfoil curves [4, 5, 12]. Among them, the Bézier curve is the most basic and common expression. An airfoil usually consists of multiple control points $\{x, y\}$ to a polynomial function. Usually, a n -degree Bézier curve that connects $n+1$ control points is chosen as basis to form a smooth curve which are used to approximate a part of a airfoil function. Then the airfoil curve function can be described as a liner combination of the basis. These polynomial expressions mentioned above are flexible and they can be combined with other parameterization methods to describe airfoil characteristics more accurately [5]. The class function/shape function transformation (CST) [13], developed based on polynomials, is the mainstream method for extracting features in the field of airfoil parameterization. The CST uses both the class function and the shape function to control the airfoil shape. The class function is applied to generate the basic shape of airfoils, and the shape function is used to correct the basic shape so as to obtain an accurate airfoil shape. The coefficients of the class function and shape function are parameters to be determined in CST. These polynomial approaches that are mathematically interpretable were widely used to parameterize airfoils. However, there have been some frontier studies which show that curves or surfaces of an airfoil exist in manifold space [14, 15]. Hence, existing polynomial approaches can only capture features from Euclidean space, and some latent features (e.g., geometric-features from manifold space) are omitted.

Deep learning, as a data-driven model, has made grate progress in feature extraction of airfoils. Chen et al. [16] adopted the info-GAN to generate massive airfoils. In their work, Chen established relations between three latent codes in the info-GAN and design parameters of airfoils, and the info-GAN was controlled to generate specific airfoils by adjusting the three latent codes. Yilmaz et al. [17] explored the relationship between NACA 63(2)-615 airfoil and its pressure distribution. With the help of the local feature perception ability of CNNs, Yilmaz constructed a deep learning model to predict the pressure at each airfoil coordinate. Jing et al. [18] combined an Auto-Encoder and a GAN together to generate target wall Mach distributions for the inverse design that matches locations of suction peak, shock and aft loading. The existing studies mostly take airfoil coordinates or pictures of airfoils as inputs, and the features extracted by them can not be interpreted clearly.

Our proposed BFE is different from both the polynomial definition and deep learning. For polynomial definition, the output of them (including CST) are polynomials with certain coefficients, which are used to represent features of airfoils. Relatively, a polynomial with certain coefficients is just an intermediate of BFE. And the final output of BFE are the fused geometric-features that clearly express the geometric shape of airfoils from both Euclidean space and manifold space. Therefore, BFE is a further development of polynomial definition. For deep learning, the features extracted are usually hard to interpret. However, The GF-Encoder combines airfoil coordinate data and manifold metrics together to from fused geometric-features. Because airfoil coordinates are physically interpretable and manifold metrics are math-

ematically interpretable, BFE, to some extent, improves the geometrical meaning of feature representation.

Methodology

Overview

Fig.1 describes the structure of BFE which consists of two parts: manifold metric feature extraction and GF-Encoder. In the manifold metric feature extraction part, a 3-degree self-intersection-free Bézier curve is chosen as the basis of a polynomial, then an airfoil shape can be described as a Bézier manifold that are composed of multiple basis. A manifold metric calculation module is responsible for calculating the inner-product of two perpendicular vectors from the airfoil manifold tangent space. In the GF-Encoder part, the airfoil coordinate data and manifold metrics are all encoded into fused geometric-features, based on which the generated airfoil coordinate data and the corresponding generated manifold metrics are exported. The fused geometric-features are the final outputs that express geometric shape of airfoils from both Euclidean and manifold space. In this section, we will introduce the manifold metric feature extraction and the GF-Encoder separately.

The Manifold Metric Extraction

Given an 2D airfoil coordinate set $D = \{P_i = (x_i, y_i) | i = 1, 2, \dots, M\}$, where M denotes the number of coordinate points. A 3-degree Bézier curve can be built:

$$r(D; t) = \sum_{i=0}^n P_i B_{i,n}(t), t \in [0, 1], n = 3 \quad (1)$$

where $r(D; t)$ is the function of Bézier curve, D denotes the sample space where the airfoil coordinates located, t is the parameter of this Bézier curve, n is the degree of Bézier curve and $B_{i,n}(t)$ denotes the coefficient which satisfies:

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} [i = 0, 1, \dots, n] \quad (2)$$

A n -degree Bézier curve with $n \geq 3$ may have self-intersections (Fig.2 (a)), which cannot be used to construct a manifold [19]. There are two solutions to avoid the

self-intersections, see Fig.2 (b) and (c). In the subgraph (b), the self-intersection A is deleted and the remaining curves construct two manifold. In the subgraph (c), the sequence (or position) of four control points makes the Bézier curve has no self-intersections. In addition, adjusting the sequence of control points is easier than deleting self-intersections. Therefore, we only care about the n -degree Bézier curves as in subgraph (c).

First, we prove that a self-intersection-free Bézier curve [19] forms a smooth Bézier manifold. We introduce the lemma of manifold [20, 21]:

Lemma 1 *Let $\mathcal{M}$ be a non-empty Hausdorff space. $\mathcal{M}$ is called an m -dimensional topological manifold if, for every point $P \in \mathcal{M}$, there exists an open neighborhood U of the point P that satisfied $U \subset \mathcal{M}$ and a homeomorphic $\phi : U \rightarrow \mathbb{R}^m$ from U to an open set of the m dimensional Euclidean space $\mathbb{R}^m$.*

Proof Because that the space $\mathcal{M}$ where a Bézier curve $r(D; t)$ located is a subset of $\mathcal{R}$, i.e., $\mathcal{M} \subset \mathcal{R}$, then $\exists \mathcal{T}$, s.t. $(r, \mathcal{T})$ constitutes a topological space (i.e. Hausdorff space). This is the prerequisite of Lemma 1.

Let the Euclidean space where $D = \{P_i = (x_i, y_i) | i = 1, 2, \dots, M\}$ located be $\mathbb{R}^2$. Because of the smoothness of $r(D; t)$, for $\forall t_0 \in r(D; t)$, there must exists an open neighborhood U s.t. $t_0 \in U$. For $\forall t_1, t_2 \in U$, and $t_1 \neq t_2$, $\exists P_1 \in \mathbb{R}^2$ s.t. $r : t_1 \rightarrow P_1$ and $\exists P_2 \in \mathbb{R}^2$ s.t. $r : t_2 \rightarrow P_2$.

Suppose that $P_1 = P_2$, then $P_1 = r(D; t_1)$ and $P_2 = r(D; t_2)$, therefore $r(D; t_1) = r(D; t_2)$ which contradicts the functional properties of $r(D; t)$, i.e. $P_1 \neq P_2$. $r(D; t)$ is a homeomorphic.

According to Lemma 1, there exists a homeomorphic $r(D; t) : U \rightarrow \mathbb{R}^2$, s.t. the space $\mathcal{M}$ is a manifold, we call it Bézier manifold.

The proof is finished. □

The Bézier manifold constructed from the control points of airfoils is actually a segmented smooth manifold because a 3-degree Bézier curve is determined by four control points, see Fig.3. In this figure, the control points A, B, C, and D determine segment 1, and the control points D, E, F and G determine segment 2, and so on. Multiple smooth segments are connected end to end to form a segmented smooth Bézier manifold.

The construction of Bézier manifold build a map between 2D airfoil coordinates space $\mathbb{R}^2 = \{x_i \in \mathcal{R}, y_i \in \mathcal{R}\}$ and a new 1D manifold space $\mathcal{M}$ with parameter t ,

which is an efficient approach: on one hand, a Bézier manifold can sense the airfoil curve by controlling parameter t ; on the other hand, the Bézier manifold can be used to calculate the manifold metric feature at each t conveniently.

Then, the geometric-feature called manifold metric is extracted by [11]:

$$g_{vw}(t) = \partial_v r(D; t) \partial_w r(D; t) \quad (3)$$

where $g_{vw}(t)$ is the manifold metric of an airfoil manifold at point t , $\partial_v = \frac{\partial}{\partial t^v}$ denotes directions of partial derivative. Considering that $r(D; t)$ is an 1D manifold, therefore, $v = w$:

$$g_{vw}(t) = g_{vv}(t) = (\partial_v r(D; t))^2 = g_{ww}(t) \quad (4)$$

The manifold metric $g_{vw}(t)$ denotes the inner-product of two perpendicular vectors in the tangent space $\{\frac{\partial}{\partial t^v}\}$ of $r(D; t)$. The two vectors that make up this inner-product can be regarded as the basis of any vector in the tangent space of $r(D; t)$. As a result, $g_{vw}(t)$ is chosen as geometric-feature that represents the geometric characteristics of the tangent space of airfoil curves.

The GF Encoder

The aim of our proposed GF-Encoder is to combine the airfoil coordinate data and the manifold metrics together to form fused geometric-features. The GF-Encoder (Fig.4) is composed of three modules: an Encoder, a fusion network and a Decoder. The Encoder consists of a CNN which takes airfoil coordinate data P_{ij} as input and a FCN (called FCN_1) which takes manifold metric $g_{vw}^i(j)$ as input. The FCN (called FCN_2) combines the output from CNN and FCN_1 together to output fused geometric-features to express geometric-features of airfoils. The Decoder consists of a deconvolution neural network (called DeCNN) and another FCN (called FCN_3). The output of the DeCNN is generated airfoil coordinate data P'_{ij} , and the output of FCN_3 is generated manifold metric $g_{vw}^{'i}(j)$.

The loss function of the GF-Encoder is:

$$\begin{cases} \mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \\ \mathcal{L}_1 = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (P_{ij} - P'_{ij})^2 \\ \mathcal{L}_2 = \frac{1}{NM'} \sum_{i=1}^N \sum_{j=1}^{M'} (g_{vw}^i(j) - g_{vw}^{'i}(j))^2 \end{cases} \quad (5)$$

where N is the number of input data, M is the number of coordinate points of an airfoil, M' is the number of manifold metrics of an airfoil, $P_{ij} = (x_{ij}, y_{ij})$ denotes the j th coordinate data for the i th input airfoil, P'_{ij} denotes the exported coordinate data, $g_{vw}^i(j)$ is manifold metric of the i th airfoil at $t = j$ and $g_{vw}^{'i}(j)$ denotes the exported manifold metric.

Experiments and Results

To validate the feasibility of BFE, two experiments based on the public UIUC airfoil dataset are conducted. Experiment I is used to extract the geometric-features of airfoils. Experiment II is used to fuse the discrepant data (i.e., the fused geometric-features extracted from the experiment I and the flight conditions) to predict the C_D and C_L of airfoils.

Dataset

The public UIUC database^[1] provides more than 1500 real airfoils, each of which is discretized by 2D coordinates. However, the UIUC dataset has many defects: 1) some airfoils only have the upper surface coordinates (we call them damaged airfoils), which cannot represent complete airfoils; 2) most airfoil coordinates are sorted in the order of the trailing edge, the upper surface, the leading edge, and the lower surface. However, some airfoil coordinates are not sorted in the above order (we call them out-of-order airfoil coordinates), which may build Bézier curves with self-intersections; 3) the number of coordinates of different airfoils are different. Therefore, the pre-processing of UIUC dataset is necessary.

For damaged airfoils, we removed airfoils with only upper or lower surface to guarantee that the remaining airfoils in the dataset are complete. For out-of-order airfoil coordinates, we sorted them in the order of the trailing edge, the upper

^[1]http://m-selig.ae.illinois.edu/adscoor_database.html

surface, the leading edge, and the lower surface to avoid the generation of self-intersections. For the problem of different number of coordinates, we smooth the original airfoil coordinates through a 3-degree Bezier curve, and then sample 279 coordinates for each airfoil.

Experiment I: Feature Extraction Experiment

In this experiment, the BFE is used to extract the manifold metrics from the manifold space of airfoil curves, and then the manifold metrics and coordinates of airfoils are combined to form fused geometric-features. We compare the GF-Encoder and a classical Auto-Encoder to verify whether smooth and realistic airfoils can be generated by the fused geometric-features.

Figure 5 Schematic of the GF encoder (a) and a classic Auto-Encoder (b). The first convolutional layer consists of sixteen 2×3 filters, the stride of these filters are 2×1 and the padding of convolution layer is 'SAME'. The remaining nodes are similar.

Hyper Parameters of Models

Fig.5 describes the structure of the GF-Encoder and the classic Auto-Encoder. The hyper-parameters of models mentioned in this experiment are shown in Tab.1. We see that the hyper-parameters of the GF-Encoder and the Auto-Encoder are the same, and only the type of inputs and model structure are different.

The mean square error (MSE) of airfoil coordinate data was chosen to measure the distance between the input airfoil coordinates and the output airfoil coordinates:

$$MSE = \frac{1}{N} \sum_{i=1}^N (P_i - P'_i)^2 \quad (6)$$

Figure 6 The airfoils generated by the BFE and the Auto-Encoder.

where, P_i denotes the input airfoil coordinates and P'_i denotes the exported airfoil coordinates.

Table 1 Hyper-parameters of GF Encoder and Auto-Encoder.

parameter type	parameter name	value
hardware platform	GPU	four Tesla K80
software platform	programming language	Python 3.7.6
	deep learning framework	Tensorflow 1.13.1
hyper parameters of models	optimization algorithm	Adam(Beta1=0.9,Beta2=0.999)
	Data normalization	max-min normalization
	active function	Relu
	batch normalization	Yes
	batch size	128
	learning rate	0.001
	epoch	2000
	validation	10-fold cross validation
	traing set: validation set: test set	8: 1: 1

In order to conduct 10-fold cross validation [22], the whole dataset is randomly divided into 10 subsets. Of the 10 subsets, one single subset is selected for the testing set, and the remaining 9 subsets are selected for the training process. Among the 9 subsets, we randomly select 1 subset for the validation set, and the remaining 8 subsets are used for the training set, i.e. the training set, validation set and testing set are randomly divided at a ratio of 8:1:1. There is no intersection among all subsets, besides, every subset has the same opportunity to serve as a testing set. The statistical errors in this paper are the average errors of 10 folds.

Results

The experimental MSE is shown in Tab.2. In this table, five typical airfoils are analyzed. We see that the MSE between the real airfoils and the airfoils exported by the BFE is smaller than that of the Auto-Encoder. Fig.6 illustrates the above five airfoils exported by the GF-Encoder and the Auto-Encoder. Although we used a 10-degree Bézier curve to smooth all generated airfoils, the airfoils generated by the Auto-Encoder are still not smooth, which indicates that the Auto-Encoder cannot effectively extract airfoil features, and those non-smooth airfoils cannot be applied in the field of airfoil design. On the contrary, the airfoils generated by GF-Encoder are closer to the real airfoils and more smooth than those generated by the Auto-Encoder. The experimental results of this part show that our proposed BFE has the ability to accurately extract the geometric-features of airfoils.

Table 2 The MSE of typical airfoils decoded by BFE and the Auto-Encoder.

airfoils	BFE	Auto-Encoder
airfoil_1	6.90E-4	9.37E-4
airfoil_2	6.22E-4	7.40E-4
airfoil_3	5.03E-4	1.78E-3
airfoil_4	1.14E-3	1.82E-3
airfoil_5	7.07E-4	1.33E-3

Experiment II: Aerodynamic Performance Prediction Experiment

In this experiment, the MTL [10, 23] is employed to fuse discrepant data to verify whether the fused geometric-features can accurately predict aerodynamic performance (i.e., C_L and C_D) of airfoils.

Hyper Parameters of Models

The hyper-parameters of models mentioned in this experiment are still shown in Tab.1. In addition, Fig.7 shows the structures and inputs of MTL. The MTL described in subgraph (a) (we call it MTL_g) takes the fused geometric-features and the flight conditions as inputs. While, the MTL described in subgraph (b) (we call it MTL_c) takes the coordinates and flight conditions as inputs. The flight conditions in this paper contain Mach number (Ma) and Angle of Attack (Aoa).

The mean square error and mean absolute error (MAE) were chosen to measure the predicted aerodynamic performance:

$$\begin{cases} MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \\ MAE = \frac{1}{N} \sum_{i=1}^N ||y_i - \hat{y}_i|| \end{cases} \quad (7)$$

where y_i denotes the predicted performance and $\hat{y}_i$ denotes the true performance.

Results

Tab.3 shows the prediction errors of MTL_g and MTL_c. We see that MTL_g is more accurate than MTL_c, which means that the fused geometric-features extracted by BFE indeed reflect the geometric shape of airfoils, and can be fused with flight conditions to reduce the prediction error of C_L and C_D .

Table 3 The MSE of MTL_g and MTL_c.

models	MSE of C_D	MAE of C_D	MSE of C_L	MAE of C_L
MTL_g	2.29×10^{-8}	8.02×10^{-5}	4.85×10^{-5}	5.37×10^{-3}
MTL_c	8.43×10^{-8}	1.53×10^{-4}	9.85×10^{-5}	7.51×10^{-3}

Fig.8 shows the C_L and C_D predicted by the MTL_g and MTL_c, respectively. The C_L predicted by MTL_g and MTL_c is similar. However, the C_D predicted by the MTL_g is more accurate than that predicted by the MTL_c. The visualization results of predicted C_L and C_D are shown in Fig.9. The results show that the C_L and C_D predicted by the MTL_g are more close to the true value calculated by CFD.

As a result, experiment I proved that the fused geometric-features extracted by the BFE reflect the geometric shape of airfoils, and experiment II shows that the fused geometric-features can be fused with flight conditions to reduce the prediction errors of C_L and C_D of airfoils.

Conclusion

The conclusions of this paper are as follows:

- 1 the airfoils exported by the GF-Encoder are more accurate and smooth than those exported by the Auto-Encoder.
- 2 the proposed geometric-features (i.e., manifold metrics) indeed capture latent geometric-features of airfoils;
- 3 the fused geometric-features can be used to reduce the prediction errors of C_L and C_D of airfoils;
- 4 the proposed BFE is proved to be feasible.

Formula (4) only represent one latent geometric-feature of airfoil curves. In the future, more latent geometric-features of airfoil curves will be defined and extracted, and the fused geometric-feature will be enriched and more geometrically significant.

Availability of data and materials

The airfoil dataset used in our experiments is UIUC Airfoil Coordinates Database. All data exported or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work was supported by the National Numerical Wind Tunnel Project 13RH19ZT6B1, 13RH19ZT62B and Basic Scientific Research Project of Central Universities A030202063008039.

Authors' contributions

Yu Xiang and Liwei Hu contribute equally.

Yu Xiang: Conceptualization, Methodology and Writing.

Liwei Hu: Conceptualization, Methodology and Writing.

Jun Zhang: Software, Validation, Investigation and Formal analysis Review and Editing.

Wenyong Wang: Investigation, Formal analysis, Data Curation and Visualization.

Acknowledgements

The authors would like to thank Dr. Wenzheng Wang, Research Fellow, Dr. Yanqing Cheng, Associate Research Fellow, and Dr. Weiqi Qian, Research Fellow, all from China Aerodynamics Research and Development Center for their valuable suggestions with our project and this paper.

Author details

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

References

1. Wang, Y., Shimada, K., Farimani, A.B.: Airfoil gan: Encoding and synthesizing airfoils for aerodynamic-aware shape optimization. *arXiv preprint arXiv:2101.04757* (2021)
2. Hu, L., Zhang, J., Xiang, Y., Wang, W.: Neural networks-based aerodynamic data modeling: A comprehensive review. *IEEE Access* **8**, 90805–90823 (2020)
3. Schnepf, E.: Geometry generator for cfd and applied aerodynamics. In: Sobieczky, H. (ed.) *New Design Concepts for High Speed Air Transport*, 1st edn., pp. 137–157. Springer, New York (1997)
4. Derksen, R., Rogalsky, T.: Bezier-parsec: An optimized aerofoil parameterization for design. *Advances in engineering software* **41**(7–8), 923–930 (2010)
5. Farin, G.: *Curves and Surfaces for Computer-aided Geometric Design: a Practical Guide*. Elsevier, San Diego (2014)
6. Yonekura, K., Hattori, H.: Framework for design optimization using deep reinforcement learning. *Structural and Multidisciplinary Optimization* **60**(4), 1709–1713 (2019)
7. Chen, W., Ahmed, F.: Padgan: A generative adversarial network for performance augmented diverse designs. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 84003, pp. 11–11010 (2020). American Society of Mechanical Engineers
8. Hu, L., Wang, W., Xiang, Y., Zhang, J.: Flow field reconstructions with gans based on radial basis functions. *arXiv preprint arXiv:2009.02285* (2020)
9. Zhang, Y., Sung, W.J., Mavris, D.N.: Application of convolutional neural network to predict airfoil lift coefficient. In: *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 1903 (2018)
10. Hu, L., Xiang, Y., Zhang, J., Shi, Z., Wang, W.: Aerodynamic data predictions based on multi-task learning. *Applied Soft Computing*, 108369 (2021)
11. Alexakis, S., Balehowsky, T., Nachman, A.: Determining a riemannian metric from minimal areas. *Advances in Mathematics* **366**, 107025 (2020)
12. Zhang, D., Wang, Z., Ling, H., Zhu, X.: Kriging-based shape optimization framework for blended-wing-body underwater glider with nurbs-based parametrization. *Ocean Engineering* **219**, 108212 (2021)
13. Kulfan, B.: Modification of CST airfoil representation methodology (2009)
14. Du, X., Ren, J., Leifsson, L.: Aerodynamic inverse design using multifidelity models and manifold mapping. *Aerospace Science and Technology* **85**, 371–385 (2019)
15. Nagawkar, J., Ren, J., Du, X., Leifsson, L., Koziel, S.: Single-and multipoint aerodynamic shape optimization using multifidelity models and manifold mapping. *Journal of Aircraft* **58**(3), 591–608 (2021)
16. Chen, W., Chiu, K., Fuge, M.: Aerodynamic design optimization and shape exploration using generative adversarial networks. In: *AIAA Scitech 2019 Forum*, p. 2351 (2019)
17. Yilmaz, E., German, B.: A convolutional neural network approach to training predictors for airfoil performance. In: *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 3660 (2017)
18. Jing, W., Runze, L., Cheng, H., Haixin, C., CHENG, R., Chen, Z., ZHANG, M.: An inverse design method for supercritical airfoil based on conditional generative models. *Chinese Journal of Aeronautics* (2021)
19. Said Mad Zain, S.A.A.A., Misro, M.Y., Miura, K.T., *et al.*: Generalized fractional bézier curve with shape parameters. *Mathematics* **9**(17), 2141 (2021)
20. Tang, Z.: Metric. In: Yue, C. (ed.) *Fundamentals of Riemannian Geometry in Chinese*, 1st edn., pp. 29–33. Beijing Normal University Publishing House, Beijing (2011)
21. Chen, W.: Riemannian metric. In: Zhang, J. (ed.) *Introduction to Riemann Geometry in Chinese*, 1st edn., pp. 83–90. Beijing Normal University Publishing House, Beijing (2004)
22. Barnhart, S.A., Narayanan, B., Gunasekaran, S.: Blown wing aerodynamic coefficient predictions using traditional machine learning and data science approaches. In: *AIAA Scitech 2021 Forum*, p. 0616 (2021)
23. Zhang, J., Zhang, G., Cheng, Y.: A multi-task learning method for large discrepant aerodynamic data. *Acta Aerodynamica Sinica*, 1–10 (2021)