An ANN-based advancing double-front method for automatic isotropic triangle generation

Peng Lu
Southwest University of Science and Technology

Nianhua Wang
China Aerodynamics Research and Development Center

Xinghua Chang
National Innovation Institute of Defense Technology

Laiping Zhang (zhanglp@swust.edu.cn)
National Innovation Institute of Defense Technology

Yadong Wu
Sichuan University of Science and Engineering

Hongying Zhang
Southwest University of Science and Technology

Research Article

Keywords: Isotropic triangular mesh generation, Artificial neural network, Advancing front method, Machine learning

Posted Date: February 18th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1292374/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
An ANN-based advancing double-front method for automatic isotropic triangle generation

Peng Lu¹,²,³ · Nianhua Wang³ · Xinghua Chang⁴ · Laiping Zhang¹,⁴,⁵ · Yadong Wu¹,⁵ · Hongying Zhang¹

¹) School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China
²) School of Intelligent Manufacturing Engineering, Chongqing University of Arts and Science, Yongchuan 402160, China
³) Stake Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang 621000, China
⁴) Unmanned Systems Research Center, National Innovation Institute of Defense Technology, Beijing 100071, China
⁵) School of Computer Science and Technology, Sichuan University of Science & Engineering, Yibin 644005, China

Corresponding author. E-mail address: zhanglp@swust.edu.cn (Laiping Zhang)

Abstract The advancing front method (AFM) is one of the widely used unstructured grid generation techniques. However, the efficiency is low due to the fact that only one cell is generated in the advancing procedure. In this work, a novel automatic isotropic triangle generation technique is developed by introducing an artificial neural network (ANN) based advancing double-front method (ADFM) to improve the mesh generation efficiency. First, an initial isotropic triangular mesh is generated by the traditional mesh generation method as the training set of ANN. Second, the ADFM is presented in detail to improve the advancing efficiency, including the structure of the ANN to distinguish the different patterns in the advancing procedure. Finally, several typical cases are tested to validate the effectiveness. The experimental results show that the ANN can accurately identify mesh generation patterns, and the mesh generation efficiency is 50% higher than that of the traditional single-front AFM.

Keywords Isotropic triangular mesh generation · Artificial neural network · Advancing front method · Machine learning

1 Introduction

Mesh generation is one of the most important issues in numerical simulations, especially for complex engineering applications, because mesh generation is the first step to get the numerical solution, and the mesh quality directly affects the accuracy of numerical solutions. So it plays an important role in numerical simulations, and it is still a grand challenge in engineering applications, despite several decades of effort. [1-3].

Generally, there are two main types of computational mesh, structured and unstructured grid. Compared with structured grids, unstructured grids have the following significant advantages: (1) Unstructured grids have the advantages of generality and flexibility to deal with complex geometries, and it is easy to control the size of the mesh cells and the density of mesh points. (2) The random data structure of the unstructured grids is conducive to mesh adaptation, therefore, unstructured grid generation requires little user time and effort. Additionally, the unstructured grid generation methods are well suited to inexperienced users because they require little user input.

Due to the outstanding advantages of unstructured grids, many scholars have proposed various unstructured grid generation methods. The widely used methods include the advancing front methods (AFM) [4-7], Delaunay-based methods [8–11], and the modified quadtree/octree methods [12-14].

The advancing front methods cut the boundary of the computing domain into many discrete fronts and all the fronts form an initial front stack. Generally, the fronts are a set of edges in two dimensions (2D) or triangular faces in three dimensions (3D). During the advancing procedure, the original fronts become gradually the interior faces of the generated cells and a new set of fronts are created, this process continues until the computational domain is filled. Since the advancing procedure is carried out from the chosen front one by one, so the efficiency is relatively low for large-scale grid generation.

The majority of Delaunay-based methods mainly utilize the incremental algorithm, which starts with an initial triangulation of several points covering the whole computational domain. After inserting new points according to a step size limit, local triangulation is reconstructed based on the well-known Delaunay criterion until the whole triangular grids are generated to meet the grid size distribution requirement. The efficiency is relatively high since several cells are generated simultaneously. However, special treatment should be carried out carefully to confirm the original geometry.

The idea of the modified quadtree/octree algorithms can be divided into the following steps. Firstly, the whole computational domain is covered with a multi-level rectangular (2D)/cubic (3D) tree structure. Secondly, according to the shape and flow field characteristics, the local mesh is subdivided by the tree structure until the grid size meets the computing requirements. Finally, the cells outside the boundaries are deleted, and the cells that intersect
An ANN-based advancing double-front method for automatic isotropic triangle generation

The boundary are cut into polygons or polyhedrons, then the polygons or polyhedrons can be further cut into simplex, e.g. triangles or tetrahedrons.

After several decades of effort of CFD researchers, unstructured grid generation techniques have made great progress. However, as we know, all the unstructured grid generation methods have their advantages and disadvantages. As mentioned in NASA’s CFD vision 2030 study [15], a high-quality automated mesh generation over complex configurations is still a grand challenge in CFD engineering applications. Even with modern commercial software (e.g., Ansys ICEM-CFD, Fluent Meshing, Hypermesh, and Pointwise) and open-source packages (e.g., NetGen and Gmsh), automation level and mesh quality are two of the most important issues in mesh generation. According to reference [16], mesh generation usually takes up about 60% of the manpower time in the entire computation cycle, a highly automated mesh generation method can undoubtedly save a lot of labor costs in CFD applications.

To improve the traditional unstructured grid generation algorithms, some researchers tried to introduce artificial intelligence (AI) algorithms such as expert systems and neural networks to enhance the automation level of mesh generation and had obtained some preliminary results. For example, Dolšak and Jezernik proposed an expert system to decide the appropriate resolution for mesh generation, this expert system was used to create rules of mesh generation to replace previous expertise [17]. Yao et al. used an ANN-based element extraction method to generate a mesh automatically [18], the ANN was used to represent some element extraction rules and to train the relationship behind these rules. Yilmaz and Kuzuoglu proposed a particle swarm optimization method to optimize the quality of the hexahedral mesh, which solved the problem of the poor effect of the Laplacian mesh smoothing algorithm in concave regions [19]. In the past year, machine learning algorithms based on neural networks have been further developed in grid generation and optimization, which has further improved the level of grid intelligence [20-23]. Pan et al. [24] solved the self-learning problem of grid generation by combining reinforcement learning with supervised learning. However, these earlier works should be continuously promoted forward, especially with the rapid development of AI technology in recent years. To the authors’ knowledge, there is still no mature application of AI algorithms in high-quality automated mesh generation.

In our previous work [25], the authors have tried to develop an ANN-based AFM to improve the computational efficiency of unstructured grid generation. However, although some AI technologies have been introduced into unstructured grid generation and the efficiency is improved to a certain degree, only one front is advanced in each advancing step, so the efficiency of mesh generation can be further improved if two or more fronts are advanced simultaneously in each advancing step. That is the main point in this work. Therefore, in this paper, combining the advantages of ANN and AFM, an ANN-based advancing double-front method (ADFM) is proposed. In this way, a large number of geometric operations (tedious intersection checking with the existed fronts, the selection of active fronts, and candidate points) will be reduced to improve mesh generation efficiency.

The rest of this paper is organized as follows. Section 2 briefly reviews the traditional AFM and introduces the idea of ADFM. In Section 3, the new isotropic triangle mesh generation technology (ANN-based ADFM) will be introduced in detail, including the extraction of training data set, preprocessing of the training data, training of ANN, mesh size controlling, and quality optimization of initial mesh. In Section 4, several typical cases, such as the NACA0012 airfoil and the 30P30N three-element airfoil, are tested to validate the effectiveness. Finally, some concluding remarks are summarized and future works are discussed in Section 5.

2 Isotropic triangle generation algorithm of ANN-based advancing double-front method

2.1 Brief introduction of traditional advancing front method and its problem

Adopting AFM to generate triangular grids was first formulated by George [4], but this original publication did not receive significant immediate attention. The first journal publication of this method was proposed by Lo [5]. Subsequently, many researchers extended this method to 3D mesh generation [6,7].

The discrete fronts are some straight line segments (called edges) in 2D cases that distinguish the meshed area from the unmeshed area as shown in Fig.1. At the beginning of the mesh generation process, the initial fronts consist of the sequence of edges that connect consecutive boundary points. When new triangles are generated one by one, the meshed area continues to advance, and the unmeshed area shrinks simultaneously. When the data stack of the front is empty, it indicates that the whole computing domain has meshed, and the mesh generation procedure is completed. As shown in Fig.1, all the dashed lines represent the fronts, the segment \( \{ P(j), P(k) \} \) is chosen as an active front to generate a new triangle.

Fig.1 Schematic diagram of mesh generation by AFM
An ANN-based advancing double-front method for automatic isotropic triangle generation

Fig. 2 Flowchart for triangular mesh generation by the traditional AFM

As shown in Fig. 2, the procedure of the traditional AFM can be summarized in the following steps:

1. Input the original discrete geometric boundary, such as the discrete rectangle in Fig. 1. All the discrete edges are called the set of initial fronts, \( F \).

2. When \( F \) is empty, skip to Step 6; otherwise, search for the most suitable front \( f \in F \) as the active front to generate a new cell, usually the shortest one is chosen as the candidate.

3. Search the points near the selected front, \( f \), to obtain a set of candidate points to generate a triangle. The candidate point set \( P_{\text{set}} \) contains the newly created point \( P_{\text{new}} \) and the endpoints of some adjacent fronts. The position of \( P_{\text{new}} = P_m + S_{\text{adv}} \), where \( P_m \) represents the center point of the front \( f \), \( S_{\text{adv}} \) represents the advance distance at \( P_m \), and \( n_{\text{adv}} \) is the unit normal vector pointing to the computational domain. After the candidate point set \( P_{\text{set}} \) is created, and then sort these points by the mesh quality coefficient of the cells to be generated. To choose the points that already existed on the adjacent fronts, the quality coefficient of \( P_{\text{new}} \) needs to be discounted.

4. Pick the best candidate point \( P \), \( P \in P_{\text{set}} \), and connect with the active front \( f \) to form a new possible cell, \( \text{Cell}_1 \).

5. If \( \text{Cell}_1 \) passes through the validity checking such as intersection judgment, then add this cell into the generated mesh list, and update the active front \( F \), then go back to Step 2; otherwise, go back to Step 4 for the next candidate point.

6. Mesh quality optimization to improve the quality of the final mesh.

It is obvious from Fig. 2 that the necessary step is intersection checking in AFM, there will be a large number of searching and sorting operations for each new cell generation, so the advancing efficiency needs to be further improved.

2.2 Formulation of the improved advancing double-front method

To reduce the searching and sorting operations in traditional AFMs, the fronts around the active front are taken into account as the environment of mesh generation [26]. According to the angles \( \theta_1 \) and \( \theta_2 \) between the fronts, it can be concluded that there are three patterns of isotropic triangles generated by the traditional AFM as shown in Fig. 3.

Fig. 3 Three basic patterns of the input-output relationship

In Fig. 3(b), where the solid line connected by \( P_1, P_2, \) is the active front which will generate a new triangle \( P_1P_2P_1\text{new} \). \( P_1 \) and \( P_2 \) are the directly adjacent points on the left and right ends of the active front \( P_1P_2 \). It should be noticed that the solid lines \( P_1P_2 \) and \( P_1\text{new}P_2 \) belong to the inactive fronts which constitute the external environment for the active front \( P_1P_2 \). Once a triangle is generated, the new active front should be stored by the right-hand rule, to ensure that the mesh generated by this new front grows towards the computational domain. It is obvious that pattern 1 and pattern 3 directly connect existing points without generating a new point. Therefore, in this way, some intersection checks can be reduced to improve efficiency. The following lists the rules in the system to distinguish between different patterns.

\[
\begin{align*}
\theta_1 & \geq 90^\circ, \ 0 < \theta_2 \leq 90^\circ \quad \Rightarrow \text{pattern 1} \\
\theta_1 & \geq 90^\circ, \ \theta_2 \geq 90^\circ \quad \Rightarrow \text{pattern 2} \\
0 < \theta_1 < 90^\circ, \ \theta_2 > 90^\circ \quad \Rightarrow \text{pattern 3}
\end{align*}
\]

Inspired by the above ideas, to further improve the efficiency of mesh generation, a novel method based on double-front advancing simultaneously is proposed in this paper. As shown in Fig. 4(a), the edge \( P_1P_2 \) is the first active front and \( P_1P_2 \) is the second one, which means that each of them will generate a new triangle. \( P_1 \) and \( P_2 \) are the directly adjacent points on the left and right ends of the two fronts. However, this is only a special case of the ADFM to generate two new grids. To improve grid generation automation, the environment for grid generation needs to be classified. According to the value of \( \theta_2 \), grid generation can be divided into three categories, corresponding to the three rows in Fig. 4, respectively. When \( \theta_2 > \theta_{\text{max}} \), there will generate two triangles that do not share one edge, such as patterns 1–4 in the first row; in other cases, when \( \theta_{\text{min}} \leq \theta_2 \leq \theta_{\text{max}} \), two new triangles will also be generated, but they will share one edge, such as patterns 5–8 in the second row; finally, when
An ANN-based advancing double-front method for automatic isotropic triangle generation

\[ \theta_2 < \theta_{\text{min}} \text{ the two active fronts will generate only one triangle. Here } \theta_{\text{max}} \text{ and } \theta_{\text{min}} \text{ are constants, in this paper, } \theta_{\text{max}} \text{ is 170 degrees and } \theta_{\text{min}} \text{ is 70 degrees. The above only discussed the influence of different sizes of } \theta_2 \text{ on the procedure of grid generation, and the angles } \theta_1 \text{ and } \theta_3 \text{ between the active front and its neighbors also determine the details of mesh generation. For example, in pattern 4 in the first row, where } \theta_1 < \theta_{\text{min}} \text{ and } \theta_3 > \theta_{\text{max}}, \text{ in this case, two new triangles will be generated, however, only two new edges will be generated and no point generated. The grid generation approach for other patterns is similar to the one described above. Therefore, all the external environments of mesh generation can be divided into 10 patterns according to the angles (} \theta_1, \theta_2, \theta_3). \text{ It should be noted that the case where } \theta_1 \text{ is greater than 180° is not shown in the figure, which is the same as pattern 1-4 and two triangular elements will be generated.}

Similar to the mesh generation patterns of single front in Fig.3, the classification of double-front advancing at the same time will be more complex. So to establish such a set of rules would be cumbersome, and sometimes, a fixed rule will lead to a poor quality mesh. Meanwhile, these rules are usually obtained through an interactive trial-and-error process which depends on the experience. An automatic method of obtaining rules is essential and even indispensable for this method to be applied to a wider range of problems. Due to the ability to approximate any complex nonlinear relationship and self-learning, the well-known ANN-based machine learning method is introduced here to solve the above problem.

In the procedure of grid generation, the probability of occurrence of the above 10 patterns must be different. For example, the probability of occurrence of pattern 1 is higher than that of pattern 10, these will be discussed in detail in Section 4.5. Because of the high probability of the occurrence of pattern 1, it is necessary to refine pattern 1 to further improve efficiency. Zhang et al. proposed a concept, the combination of symbolism and connectionism, of the third generation of artificial intelligence [27]. According to this method, in this work, the ANN is utilized to do the initial classification, and then the knowledge rules are adopted to subdivide some patterns. As shown in Fig.5, the core idea is to generate the new triangular cells based on the ANN according to the size of } \theta_1. \text{ This method can reduce the difficulty of neural network design and maximize the efficiency of grid generation.}

3 ANN model and its training process in mesh generation

3.1 BP-ANN modeling

Backpropagation-Artificial Neural Network (BP-ANN) is a powerful tool in data processing, just like neurons in the brain. It can minimum error between predicted output and ideal output through continuous data learning [28-30]. As shown in Fig.6, a typical multilayer BP-ANN architecture usually contains three parts: an input layer, several hidden layers, and an output layer. Each layer is composed of some parallel single neurons, which are the core operation units of ANN. The input layer is used to load a known vector } X \text{ with multiple parameters, which directly determine the output of the system, such as the coordinates of the endpoints of a front and its surrounding adjacent points in fronts stack } F \text{ as mentioned in Section 2.1. The hidden layer contains all the input layer information and establishes a non-linear relationship between the input parameters and the output results. The output layer is utilized to calculate the unknown results of the hidden layer. In this study, the mesh generation patterns are predicted by the output layer. Once the number of hidden layers and neurons in each hidden layer were determined, the connection weights and biases between the neurons are optimized by searching the minimum error between the actual results and the expected results. The framework of a neural network involves two parts: the feed-forward propagation neural network and the backpropagation algorithm. In the early stage of the establishment of a neural network, the connection weights and biases can usually be set as random quantities ranging from 0 to 1. In the training stage of ANN, any error caused by the input parameters of the network will be sent backward through the backpropagation process to optimize the weights and biases of the ANN, the error is defined by the mean square error (MSE) which defined as

\[ \text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - k_i)^2 \] (2)

where } y_i \text{ and } k_i \text{ are the network output value and the target expected value respectively. } n \text{ is the total number. When the } \text{MSE} \text{ reaches a minimum or training steps reached the set value, the training process is finished, all the weights and biases will be stored to predict the mesh generation model in this study.}
An ANN-based advancing double-front method for automatic isotropic triangle generation

In a biological neural network, each neuron is connected to other neurons, when it gets "excited", it will change the electrical potential of the connected neurons by sending chemicals. If a neuron passes a certain threshold, it becomes "excited" and sends chemicals to other neurons. Fig. 7 illustrates the input-output processing of a single neuron, the neuron \( i \) received input signals \( x_1 \) to \( x_n \) from other neurons, these input signals are transmitted through weighted connections, the total input value will compare with the threshold of the neuron, and then processes by activation function to calculate the output of the neuron. The ideal activation function is the step function, which maps the output value to "0" or "1", "1" means the neuron is excited and "0" means the neuron is suppressed. Another common activation function is a sigmoid function which squeezes the output value with a wide range into the range of 0 to 1. For a multi-classification neural network system, the one-hot encoding (OHE) is widely used because all states are equidistant without bias in Euclidean space [31]. Intuitively, there are as many bits as states, only one bit is "1" means valid, and others are "0" means invalid at any time.

During the backpropagation process, the BP-ANN algorithm is based on the gradient descent strategy [32-34], the weights were adjusted in the direction of the negative gradient of the target value, the change of weight \( w_{ij} \) in the connection neuron can be written as

\[
\Delta w_{ij} = -\eta \frac{\partial \text{MSE}}{\partial w_{ij}}
\]

where \( \eta \) is the learning rate which is a key hyperparameter in ANN training, and it determines the update range for each parameter. If the amplitude is too large, the parameter may move back and forth on both sides of the extreme value. On the contrary, if the amplitude is too small, the optimization speed will be greatly reduced, so the variable learning rate based on an exponential decay method is usually adopted [35,36]. The main idea of the exponential decay method is to use a large learning rate to quickly get a better solution, and then gradually reduce the learning rate as the iteration continues to make the model more stable.

3.2 ANN-based advancing double-front method

To improve the efficiency of the traditional single-front AFM method in generating the isotropic triangles, the idea of ADFM is proposed in Section 2.2. However, the environment around the fronts to be advanced is complex and changeable, and the classification process needs large human judgment. So the multi-classification method of ANN is adopted here to identify and classify the external environment in this study, which is helpful to improve the intelligence level of mesh generation.

The procedure of the ANN-based ADFM can be summarized in Fig.8. An ANN model is trained to establish the mapping relationship between inputs and outputs. Training neural network model is essential to determine the weights matrix and the bias terms between neurons in different layers. This new method consists of the establishment of a neural network and mesh generation. The samples of ANN training are extracted from mesh data generated by traditional methods, the trained model is utilized to predict patterns in mesh generation.

3.3 References Construction and training of ANN

The design of a neural network directly affects the performance of the algorithm. The ability of ANN is greatly dependent on the architectures. As introduced in Section 3.1, the input of ANN is the coordinates of five points, the two fronts to be advanced and the points on the left and
An ANN-based advancing double-front method for automatic isotropic triangle generation

right adjacent fronts, the output layer is designed to solve the 10 classifications of mesh generation. Although there are some works contributed in this area [37], there is still no mature way to determine the number of hidden layers in an ANN and the number of neurons in each hidden layer for optimal architecture. In this work, the architecture of ANN is designed by the traditional empirical method with 10 neurons on the input layer, 10 neurons on the output layer, two hidden layers with 25 neurons in the first hidden layer, and 18 neurons in the second hidden layer.

3.4 Extraction and preprocessing of training sample data

The purpose of ANN is to learn the rules of mesh generation from existing mesh data generated by some traditional methods such as the Delaunay method and AFM, and so on. In this work, a variety of complex geometry mesh data are used to extract training data. For the convenience of introduction, a simple shape is used to describe the process of training data extraction. As shown in Fig.9, there are two training samples. The red lines represent the fronts to be advanced, according to the right-hand rule, the first active front is composed of Point 17 and Point 16, denoted as Front [17,16], the second active front denoted as Front [16,22], the red points (Point 25 and Point 19) are mesh points advanced by the corresponding active fronts. The blue lines represent the left and right adjacent fronts of the double fronts. In Fig 9(a), case 1 corresponds to pattern 3 in Fig.4, in this case, the first active front will generate a new mesh point (Point 25) and the second active front will use the existing point to form a triangle (Triangle [16,22,19]) with no new point generated. In Fig.9(b), case 2 is different from case 1 in which it generates two new points, this case belongs to pattern 1 in Fig.4.

As shown in Fig.9, the input data of ANN are the coordinates of five points on the active fronts, and the differences of these coordinate values are very large, some of them can reach several orders of magnitude, which are not beneficial to the convergence of ANN. Therefore, it is necessary to preprocess the training data. The most popular method of data preprocessing is data normalization, which is used to reduce data with different representations to the same scale. The common scale ranges are [-1, 1] and/or [0, 1]. For ANN and support vector machine (SVM), it is essential to use normalization in data preprocessing. Of course, data normalization is not required for some machine learning methods, such as decision trees. In this study, the transition, scaling, and rotation operations are used for data normalization [38], and Fig.10 presents the normalized data graphs of the two extracted training samples in Fig.9.

Fig.9 Extract training samples generated by the traditional method

(a) case 1

(b) case 2

Fig.10 Normalized data of the two extracted training samples
3.5 The training, verification and test of ANN

The experiment was implemented on a computer with an Intel Core i5-3230M, 16GB of random memory, and a graphics card with 4 GB. The Tensorflow framework was used to train the network parameters of the proposed mesh generation pattern recognition algorithm. The software environment is as follows: WIN10, Tensorflow-1.10, and Python-3.6 from the Anaconda-3 distribution to create an independent Python environment.

Fig.11 shows a 2D mesh over a fighter with a store and a rae2822 airfoil as training sample data, the numbers of the isotropic triangles are 8708 and 10352 respectively, any two connected edges and their left and right adjacent points will be used as a set of training data. The mesh training samples were randomly divided into the training set, test set, and verification set according to the proportions of 70%, 15%, and 15%.

As shown in Fig.12, the early stage of ANN design mainly consists of training and verification, the Adam optimization algorithm is used to minimize the loss function value during training, when the loss of the network is less than 1.0e-4, it indicates that the ANN has converged, and the weights and biases of all the layers will be saved as a model. When the accuracy rate reaches 99%, then the training process is finished. At last, the test set will be utilized to check the prediction performance of the system. The verification is similar to the test process except that the test process does not adjust the weights and biases of ANN. Fig.13 shows that the loss value changes significantly and has a decreasing trend, indicates that ANN gradually converges and tends to be stable.

Mesh size controlling has a great influence on mesh quality and flow solution accuracy. It is usually expected that the grids should be refined in key areas such as geometric curvature and large flow gradient, while in the outer far-field, grids can be very large since the flow gradient is almost zero, and the transition from the solid wall to far-field should be as smooth as possible.

Generally, the control methods of unstructured mesh size mainly include background mesh method, interpolation method, function designation method, and adaptive method of flow field characteristics. In this study, the exponential function method is used to determine the mesh step size in the computational domain [39]. The mesh size can be written as:

$$y = \frac{e^{ax} - 1}{e^x - 1} \quad (4)$$

where $x$ is the distance between the active front and the nearest initial front on the wall, the parameter $x$ has been normalized to between 0 and 1, and $y$ (0 ≤ $y$ ≤ 1) is the normalized mesh size calculated for the corresponding
An ANN-based advancing double-front method for automatic isotropic triangle generation

front, the desired mesh size is obtained by the anti-normalization process of $y$. In the actual mesh generation process, the mesh size of the first layer and the far-field boundary should be given, the Newton-Raphson iteration method is used to obtain the value of $A$. Fig. 14 shows the relationship between the distance $x$ and mesh size $y$ with different values of $A$. It should be noted that when $x = 0$, the mesh size needs to be given separately according to the current fluid environment and the geometry considered.

Fig. 14 Relation between distance and mesh size

3.7 Mesh quality enhancement options

Generally, the initial grids generated by ANN-based ADFM need to be optimized to improve mesh quality. In this paper, the Delaunay criterion is introduced to carry out diagonal transformation for elements with poor quality [40], as shown in Fig. 15. Meanwhile, the spring relaxation method is introduced to optimize the coordinates of non-boundary nodes [41, 42], which improves the element’s regular shape by repositioning a node at the barycenter of the polygon formed by its neighbor elements, as shown in Fig. 16. Numerical experiments show that better mesh quality can be obtained by alternatively using the two optimization methods after several iterations.

Fig. 15 Diagonal transformation based on Delaunay criteria

Fig. 16 Mesh quality optimization by spring method

4 Results and discussion

Once the ANN has been trained, it can be used to automatically predict mesh generation patterns for other similar problems. To demonstrate the capability of the present ANN-based ADFM, some typical cases will be illustrated in this section.

4.1 Special characters “AI+CFD”

Fig. 17 shows the isotropic triangular mesh over some special characters (“AI+CFD”), which contains many types of concave and convex angles. These characters are automatically generated by general computer-aided design software (CAD2021) and then discretized by Pointwise software. In this case, because the influence of the flow field is not considered, uniform distribution is adopted to discrete the object surfaces, the far-field size is 5 times that of the object surface. On the other hand, these characters are of multi-connected domains, and each character contains at least one closed connected domain. The ability of the present method to deal with multi-connected domains is verified. This mesh includes 1978 points and 3506 isotropic triangular cells in the computational domain. It can be seen that the mesh transition from the object surface to the far-field is reasonable, a smooth mesh distribution around the characters can be obtained.

Fig. 17 shows the isotropic triangular mesh over some special characters (“AI+CFD”), which contains many types of concave and convex angles. These characters are automatically generated by general computer-aided design software (CAD2021) and then discretized by Pointwise software. In this case, because the influence of the flow field is not considered, uniform distribution is adopted to discrete the object surfaces, the far-field size is 5 times that of the object surface. On the other hand, these characters are of multi-connected domains, and each character contains at least one closed connected domain. The ability of the present method to deal with multi-connected domains is verified. This mesh includes 1978 points and 3506 isotropic triangular cells in the computational domain. It can be seen that the mesh transition from the object surface to the far-field is reasonable, a smooth mesh distribution around the characters can be obtained.

(a) Initial boundary for mesh generation  
(b) Final triangular meshes over characters
An ANN-based advancing double-front method for automatic isotropic triangle generation

4.2 NACA0012 airfoil

Here, a more practical geometric shape (NACA0012 airfoil) is taken as an example. The computational mesh is shown in Fig. 18, which includes 1191 mesh points, 2234 triangular cells, 3425 faces. There are 148 points at the solid wall and 49 points at the far-field boundary. The far-field boundary is designed as a circle and the center of the circle is the middle point of the airfoil. Thus, when the far-field boundary is rotated around its center, the solution field can remain unchanged. Because the gradient of the flow at the leading-edge and trailing-edge regions of the airfoil varies dramatically, more dense grids are needed to obtain the appropriate numerical solution, the mesh density in the leading and trailing edges is ten times that of the middle of the airfoil, a gradual transition is adopted from dense mesh area to sparse mesh area. In the whole computational domain, the mesh is smoothly transferred from the solid wall to the far-field circle.

4.3 30P30N three-element airfoil

The last case is the well-known 30P30N three-element airfoil, which is designed by McDonnell-Douglas as a test case for aerodynamic prediction of three-element high-lift devices including slats and flaps. The final isotropic triangular grids are shown in Fig.19, including 4206 triangular cells, 6534 faces, and 2326 points. There are 390
An ANN-based advancing double-front method for automatic isotropic triangle generation points at the geometry boundary and 59 points at the far-field boundary. It is similar to NACA0012 airfoil, the grids are refined near the leading edges and the trailing edges of each component of the three-element airfoil where the curvature varies greatly. Obviously, this mesh distribution is more suitable for CFD simulations.

![Global view](image1)

(a) Global view

![Meshes near the three-element airfoil](image2)

(b) Meshes near the three-element airfoil

![Enlarged view over leading edge and main foil](image3)

(c) Enlarged view over leading edge and main foil

![Enlarged view over concave corner of main foil](image4)

(d) Enlarged view over concave corner of main foil

Fig.19 Triangular mesh generation over a three-element airfoil.

4.4 Treatment of special circumstances

The data sets used to train the model of ANN are relatively standard meshes generated by some traditional unstructured mesh generation methods. However, in the actual mesh generation process, the mesh quality may be inferior in some special cases, and the neural network had not been trained with these cases in the past, which may cause the neural network cannot correctly recognize these patterns. To solve this problem, it is necessary to perform coordinate translation processing on these mesh coordinate points to make it more similar to the characteristics of the data set for training the neural network. Experiments have verified that the modified method is effective.

![Special unreasonable geometric shape normalization processing](image5)

(b) Mesh shape corrected by scaling and translation

As shown in Fig.20(a), this is just one case, and other cases are similar to it. The lengths of the four fronts are quite different. According to the method mentioned above, the first step is taking the length of Front $[P_1, P_2]$ as the reference value, and adjust other fronts to make their lengths close to the reference length. In other words, the coordinates of $P_1$ and $P_2$ remain unchanged, and the positions of $P_3$ and $P_r$ are changed by coordinate scaling. Finally, $P_r$ is moved by coordinate translation to obtain its modified value. Through the series of operations, the shape in Fig.20(b) is obtained, which is relatively standard input data.

4.5 Mesh quality and time-consuming assessment

In this section, the quality assessment of isotropic meshes is discussed. The long-term CFD practice shows that mesh
An ANN-based advancing double-front method for automatic isotropic triangle generation

quality plays an important role in the accuracy of calculation results, so the mesh generated by any mesh generation method must be tested for quality. Here, we choose two criteria to quantify mesh quality. The first is index $r$ which represents the ratio of a triangle area to its neighbors; the other one is index $q$, which measures how similar a triangle is to an equilateral condition:

$$q_i = \frac{4\sqrt{3}S_i}{a_i^2 + b_i^2 + c_i^2}$$

where $S_i$ is the area of triangle $i$, and $a_i$, $b_i$, and $c_i$ are the three edge lengths of the triangle, respectively. The parameter $q_i$ varies between 1 (equilateral triangle) and 0 (degenerated triangle), the closer $q_i$ is to 1, the better mesh quality is.

As shown in Table 1, based on these two mesh quality testing criteria, the average and minimum values of these two indicators are calculated here. The present method is compared with commercial software ‘Pointwise’, experimental results show that the mesh quality generated by this method can reach the level of commercial software.

Table 1. Quality statistics of 2D triangular meshes

| Case          | Software       | min.$r$ | ave. $r$ | min.$q$ | ave. $q$ |
|---------------|----------------|---------|----------|----------|----------|
| Special characters | ANN-ADFT      | 0.358   | 0.862    | 0.624    | 0.965    |
| NACA0012      | Pointwise     | 0.348   | 0.853    | 0.608    | 0.931    |
| 30P30N        | ANN-ADFT      | 0.431   | 0.852    | 0.736    | 0.927    |
|               | Pointwise     | 0.435   | 0.827    | 0.635    | 0.914    |
|               | ANN-ADFT      | 0.326   | 0.842    | 0.732    | 0.962    |
|               | Pointwise     | 0.317   | 0.826    | 0.645    | 0.968    |

Table 2. Comparison of the time consumed by different methods to generate meshes

| Case          | Number of intersection checks | Time cost/s |
|---------------|------------------------------|-------------|
|               | AFM | ANN-AFM | ANN-ADFM | AFM | ANN-AFM | ANN-ADFM |
| Special characters | 201196 | 134805 | 101104 | 13.98 | 9.78 | 7.02 |
| NACA0012      | 156703 | 109419 | 85346 | 10.89 | 5.25 | 4.15 |
| 30P30N        | 293079 | 187181 | 146012 | 20.36 | 14.26 | 11.24 |

Table 3. The statistics of the number of various patterns in the grid generation procedure

| Case          | Pattern 1 | Pattern 2 | Pattern 3 | Pattern 4 | Pattern 5 | Pattern 6 | Pattern 7 | Pattern 8 | Pattern 9 | Pattern 10 |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| Special characters | 884      | 460       | 230       | 137       | 747       | 71        | 39        | 25        | 476       | 26         |
| NACA0012      | 680       | 358       | 192       | 106       | 580       | 52        | 28        | 18        | 374       | 20         |
| 30P30N        | 1352      | 681       | 359       | 209       | 1104      | 45        | 51        | 34        | 704       | 37         |

On the other hand, the purpose of this method is to improve the efficiency of mesh generation, some unnecessary intersection checks are eliminated and two or more triangles are generated in each advancing step. As shown in Table 2, the computation time of the above three meshes generated by the traditional AFM, the single front ANN-based AFM, and the ANN-based ADFM are compared. The number of intersection checks of ANN-based ADFM is only half of the traditional AFM. In terms of mesh generation efficiency, the ANN-based ADFM is almost twice higher than the traditional AFM.

Finally, the statistics of the number of various patterns in the grid generation procedure is helpful to further analyze the internal reasons for the efficiency improvement. As shown in Table 3, the number of patterns 1 and 5 accounts for more than 50 percent of the total. Therefore, it is very important to deal with these patterns well to improve the efficiency of grid generation.

5 Conclusions

In this work, a novel automatic triangular mesh generation method is proposed for complex 2D geometries, which combines the most favorable advantages of ANN and AFM. The BP neural networks are established to identify external patterns during mesh generation. The method of automatic extraction and preprocessing of neural network training data set is introduced in detail. Finally, some typical 2D geometries are tested to validate the capability of the proposed method. The mesh generated by the new method has shown that the ANN can accurately classify external patterns to reduce the large number of intersection checks introduced by the traditional AFM. The mesh generation efficiency of the new method is twice that of the traditional AFM. We can imagine that the present double-front advancing process can be extended directly to multi-front advancing cases, then the mesh generation efficiency will be further improved. We will try it in our future work.

As with most supervised learning approaches, there is a trade-off between the time spent in the training phase of the machine learning algorithm and the reduction of large user interactions with traditional methods. Nonetheless, this can be a worthwhile expense when many solutions are required to deal with a large number of related problems, such as design and optimization issues. Finally, we note that in this case, machine learning algorithms can assist engineers in
determining certain key parameters.

After decades of CFD development in research and engineering techniques, a large amount of mesh data has been obtained with the empirical knowledge of researchers. How to design more advanced AI methods combined with mesh generation is our final goal. On the other hand, mesh generation technology in 2D is relatively mature, the next step is to extend this method to surface mesh generation and volume mesh generation in 3D, especially for viscous flow simulations.

Data availability

The datasets generated during and/or analyzed during the current study are available in the Github repository, https://github.com/lupeng396/ADFM.git.

References

1. Frey P J, George P L. Mesh Generation - Application to Finite Elements. 2nd ed. Hoboken: John Wiley&Sons Inc., 2008.
2. Thompson J F, Soni B K, Weatherill N P. Handbook of Mesh Generation. Boca Raton: CRC Press, 1998.
3. Baker T J. Mesh generation: Art or science? Progress in Aerospace Science, 2015; 41:29-63.
4. George A J. computer implementation of the finite element method [PhD Thesis]. Stanford: Stanford University, 1971
5. Lo S H. A new mesh generation scheme for arbitrary planar domains. Int. J. Numer. Methods Eng. 1985; 21:1403–1426
6. Löhner R, Parikh P. Generation of three-dimensional unstructured meshes by the advancing- front method. AIAA paper 88-0515, 1988.
7. Löhner R. Extensions and improvements of the advancing front mesh generation technique. Comput. Num. Meth. Eng., 1996; 12:683-702.
8. Rowyer A. Computing Dirichlet tessellations. Comput. J., 1981, 24:162-166.
9. Watson D. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. Comput. J., 1981; 24:167-173.
10. Weatherill N P, Hassan O. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. Int. J. Num. Meth. Eng., 1994; 37:2005-2039.
11. Lee D T, Lin A K. Generalized Delaunay triangulation for planar graphs, Discrete Comput. Geom. 1986; 1:201–217.
12. Yerry M A, Shepard M S. Automatic three-dimensional mesh generation by the modified-octree technique. Int. J. Num. Meth. Eng., 1984; 20:1965-1990.
13. Yerry M A, Shepard M S. Finite element mesh generation based on a modified-quadtree approach. IEEE Computer Graphics and Applications, 1983; 3(1):36-46.
14. Shepard M S, Georges M K. Automatic three-dimensional mesh generation by the finite octree technique. International Journal for Numerical Methods in Engineering, 1991; 32(4):709-749.
15. Slotnick J, Khadadoust A, Alonso J, et al. CFD vision 2030 study: A path to revolutionary computational aeroesciences. 2014; NASA/CR–2014-218178
16. Chawner JR, Taylor NJ. Progress in geometry modeling and mesh generation toward the CFD vision 2030. AIAA Aviation Forum, June 2019, Texas.
17. Dollik B, Jerzemnik A, Brutto J. A knowledge base for finite element mesh design. Artificial Intelligence in Engineering, 1994; 9(1):19-27.
18. Yao S, Yan B, Chen B, Zeng Y. An ANN-based element extraction method for automatic mesh generation. Expert Systems with Applications. 2005; 29:193-206
19. Yilmaz A E, Kuzuzgolu M. A particle swarm optimization approach for hexahedral mesh smoothing. International Journal for Numerical Methods in Fluids, 2010; 60(1):55-78.
20. Papagiannopoulos A, Clausen P, Avellan F. How to teach neural networks to mesh: Application on 2-D simplicial contours. Neural Network. 2021; 136:152-179.
21. Guo Y, Wang C, Ma Z, et al. A new mesh smoothing method based on a neural network. Computational Mechanics, 2021.
22. Zhang Z, Jimack P K, Wang H. MeshingNet3D: Efficient generation of adapted tetrahedral meshes for computational mechanics. Advances in Engineering Software, 2021; 157-158.
23. MitsuSh K, Funke S W, Kuchta M. Hybrid FEM-NN models: Combining artificial neural networks with the finite element method. Journal of Computational Physics, 2021; 446.
24. Pan J, Huang J, Wang Y, et al. A self-learning finite element extraction system based on reinforcement learning. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 2021; 35 (2): 180-208.
25. Wang NH, Lu P, Chang XH, et al. Preliminary investigation on unstructured mesh generation technique based on advancing front method and machine learning methods. Chinese Journal of Theoretical and Applied Mechanics 2021; 53(3):740–51 [Chinese].
26. Saxena M, Perucchio R. Element extraction for automatic meshing based on recursive spatial decompositions. Computers & Structures, 1990; 36(3):513-529.
27. Zhang B, Zhu J, Su H. Toward the third generation of artificial intelligence (in Chinese). Scientia Sinica Informaticae, 2020, 50:1281-1302.
28. Haykin S. Neural networks: a comprehensive foundation. New York: Macmillan, 1994.
29. Talee V, Mathew V K, Sonawane N. CFD and ANN approach to predict the flow pattern around the square and rectangular bluff body for high Reynolds number. Materials Today: Proceedings, 2021; https://doi.org/10.1016/j.matpr.2021.06.285.
30. António C C, Afonso C F. Air temperature fields inside refrigeration cabins: A comparison of results from CFD and ANN modelling. Applied Thermal Engineering, 2011; 31(6):1244-1251.
31. Al-Najjar H A, Pradhan B, Kalantar B, et al. Landslide Susceptibility Modeling: An Integrated Novel Method Based on Machine Learning Feature Transformation[J]. Remote Sensing, 2021; 13(16):3281.
32. Rumelhart D. E, H. Geoffery E, Williams R J. Learning Representations by Back-Propagating Errors. Nature.1986; 5:533-536.
33. Schmidhuber, J. Deep Learning in Neural Networks: An Overview, Neural Netw, 2015, 61:85-117.
34. Eiji M, Stuart E D S, Kenichi N. On Derivation of MLP Backpropagation from the Kelley-Bryson Optimal-Control Gradient Formula and Its Application, Proceedings of the IEEE International Joint Conference on Neural Networks, 2000.
35. Li, Y., Fu Y., Li, H., et al. The Improved Training Algorithm of Back Propagation Neural Network with Self-Adaptive Learning Rate, 2009 International Conference on Computational Intelligence and Natural Computing. 2009; 73–76.
36. Zheng Y, Jia D C. Training memristor-based multilayer neuromorphic networks with SGD, momentum and adaptive learning rates, Neural Networks, 2020; 128:142-149.
37. Islam M M, Sattar M A, Amin M F, et al. A New Constructive Algorithm for Architectural and Functional Adaptation of Artificial Neural Networks. IEEE Trans Syst Man Cybern B Cybern, 2009; 39(6):1590-1605.
38. Haarn D, Baker M P, Carithers W R. Computer graphics with OpenGL. London, Britain: Prentice Hall, 1988.
39. Zhang L P, Chang X H, et al. Mesh generation techniques in computational fluid dynamics. Beijing, China: Science Press, 2017. (in Chinese)
40. Shewchuk J R. Delaunay refinement mesh generation [PhD Thesis]. Pittsburgh: Carnegie Mellon University, 1997.
41. Petersen S B, Rodrigues J M C, Martins P A F. Automatic generation of quadrilateral meshes for the finite element analysis of metal forming processes. Finite Elements in Analysis and Design, 2000; 35:157-168.
42. Field, D A, Laplacian smoothing and Delaunay triangulations, Communications in Applied Numerical Method. 1988; 4:709-712.

Acknowledgments

This work was supported in part by the Innovation Foundation of State Key Laboratory of Aerodynamics under Grant SKLA190104, the National Key Project under
An ANN-based advancing double-front method for automatic isotropic triangle generation

Grant GJXM92579, the Open Research Program of State Key Laboratory of Aerodynamics under Grant SKLA20200203, and the Technology Innovation and Application Development Special Project of Chongqing

Author contributions
P.L., X.C., L.Z., were responsible for conceptualization; P.L., N. W. were responsible for programming; Y.W., L.Z. were responsible for providing research materials. The manuscript was written by P.L., N. W. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.