Non-parametric Surrogate Model Method based on Machine Learning

Jiajun Cao*[a], Qingbiao Li*[b], Liping Xu[a], Rui Yang[c], Yuejin Dai[c]

[a] *Whittle Laboratory,Department of Engineering,University of Cambridge ,Cambridge, United Kingdom CB3 0DY*
[b] *Computer Lab,Department of Computer Science and Technology,University of Cambridge,Cambridge, United Kingdom CB3 0FD*
[c] *Shanghai Turbine Works Co., Ltd. Shanghai, China 200240*

Abstract

In this paper, a novel "non-parametric" surrogate model method is introduced. The new method extracts geometric information from the surface mesh of the simulation domain using Graph Neural Networks (GNNs) and predicts the two-dimensional distributions of flow variables (in forms of contour maps) using Convolutional Neural Networks (CNNs). This method can automatically extract relevant geometric information from surface mesh, while existing data-driven surrogate model methods need manual parameterization, which may introduce additional uncertainties. Existing methods can only process geometries defined by their specific parameterization methods because the inputs of existing surrogate models are human-defined geometric parameters, while new methods can process any geometries with the same topology because its input is the surface mesh. This allows users to access more design variations from different sources to create a larger database. In addition, this novel surrogate model method is able to predict the distributions of variables, not only the integrated values of performance. This paper demonstrates this novel surrogate model method with its application on optimization of a low-pressure steam turbine exhaust system (LPES). The new surrogate model uses 10 surface meshes of the LPES as input

Email addresses: jc980@cam.ac.uk (Jiajun Cao*), ql295@cam.ac.uk (Qingbiao Li*), lpx1@cam.ac.uk (Liping Xu), yangrui@shanghai-electric.com (Rui Yang), daiyj3@shanghai-electric.com (Yuejin Dai)

*Jiajun Cao and Qingbiao Li contribute equally to this work.

and predicts the power contour at the exit of the last stage. To build the surrogate model, altogether 582 numerical simulation designs have been generated, which contains two types of geometries defined by different methods. Among them, 550 cases are used for training, and 32 cases are used for testing. The power output of the last two stages of the turbine predicted by the surrogate model has 0.86 % difference compared with those of numerical simulations. The structural similarity index measure (SSIM) is used to measure the differences between the simulated and predicted two-dimensional power distributions at the exit of the last rotor, where the average SSIM of 640 contours is 0.9594 (1.0 being identical).

Keywords: surrogate model method, optimization, graph neural networks, parameterization

1. Introduction

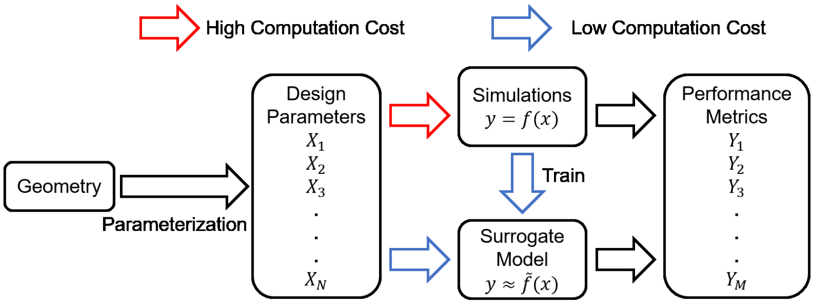


Figure 1: Process of existing surrogate model method.

The motivation of this study is to develop a surrogate model method for aerodynamics optimization, especially for fluid domains that have complex geometries. The surrogate model can accelerate the optimization by utilizing the results of previous simulations to evaluate the performance of new designs. As shown in Fig 1, one key step of building a surrogate model is manual parameterization, which is to choose some geometric parameters to describe geometry. In this step, if using too few parameters, some geometric information will be lost because it is insufficient to describe geometries with high-order surfaces (e.g., airfoils and blades). On the other hand, using too many parameters or choosing irrelevant parameters will cause the over-fitting problem [1]. It is recognized by the authors that manual parameterization is the bottleneck that prevents further improvement of the surrogate model method.

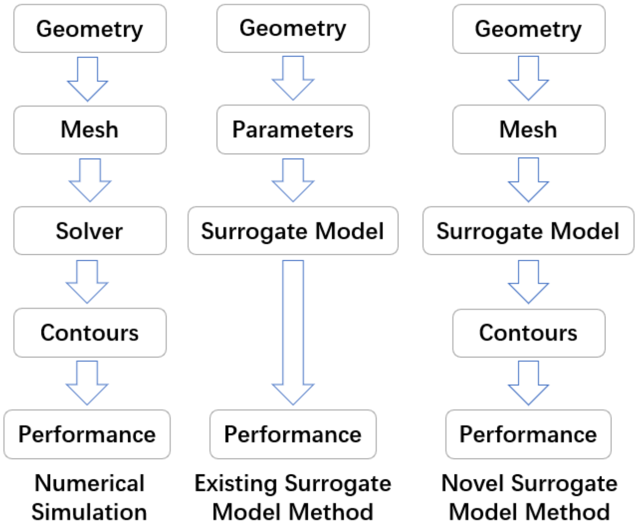


Figure 2: Comparison among numerical simulation (first column), existing surrogate model method (second column) and novel surrogate method (third column).

A novel surrogate model method has been developed in this study, which establishes a mapping relationship between the surface mesh of fluid domain and two-dimensional distributions of fluid variables (in the form of contours) with GNNs, CNNs and Conditional Variational Autoencoder (CVAE). As shown in Fig 2, the new method can process surface mesh directly and extract relevant geometric features according to their significance to the result. Compared with the existing surrogate model, the uncertainty is reduced by avoiding manual parameterization. This also allows utilizing different types of designs from different sources because the input of the model is the surface mesh, not user-defined parameters. In addition, the new method also has the ability to predict contours like numerical simulation.

In the recent decades, a lot of surrogate model methods appeared, which can be loosely categorized into: (1) Polynomial Response Surface Method [2]; (2) Kriging Model [3]; (3) Radial Basis Function and Extended Radial Basis Function [4]; (4) Artificial Neural Network [5]; (5) Support Vector Machine [6]. However, previous studies focus on the regression models, and the uncertainties generated in the parameterization limit the upper boundary of accuracy. Also, the surrogate models trained by geometric parameters can only process designs defined by the same parameterization method, which limits further increase of database. To overcome these problems, a new strategy is adapted with the help of new machine learning tools.

Recently, different kinds of neural network have attracted increasing attention in various domains. These powerful tools inspired the idea of **non-parametric surrogate model method** to directly process surface mesh of fluid domain and predict two-dimensional distributions of fluid variables. Compared with existing surrogate model methods, the new method has the ability of predicting two-dimensional distributions of variables (in forms of contour) by utilizing **CNNs to process the images**. With the combination of convolutional layers, it can extract information from graphs, and recognizes the convolutional result with multi-layer perceptron [7]. In this study, CNNs are used to predict the contours based on the latent distribution.

Another key contribution of this work is to introduce **GNNs to process surface mesh**. The nature of the graph operation of GNNs make it capable to process the non-Euclidean domain by defining the connectivity of mesh points, while CNNs are only able to process regular Euclidean data like figures. Among existing GNNs variants [8]. GNNs are categorized into three types: Recurrent GNNs, Spatial GNNs and Spectral GNNs. In this study, the surrogate model is built based on Spectral GNNs for its strengths in extracting features from large mesh [9]. Spectral GNNs are built on signal processing theory. The key step, convolutional operation, is done by Chebyshev polynomial approximation [10]. In this study, GNN can extract geometric information more comprehensively by optimizing parameters in the neural networks via back propagation of loss. This is to pick relevant information based on the feedback of prediction error, which avoids loss of geometric information and over-fitting problem. Because the new method uses the surface mesh as the input directly, it can process any geometry with the same topology structure. This enables the new model to access designs from different sources. The new surrogate model can process both unstructural mesh and structural mesh as the input thanks to the ability of GNN to process non-Euclidean data.

2. Methodology

The present surrogate model consists of two parts: optimization part (Sec. 2.1) and machine learning part (Sec. 2.2). The optimization part is used for accumulating the database for the surrogate model because the designs generated by the optimizer are similar to designs that will appear in future optimization. The designs are generated by Genetic Algorithm (GA) because it can explore the design space more comprehensive compared with the gradient-based optimization method.

2.1. Optimization

In the optimization scenarios, the surrogate model will be used to evaluate the performance of designs in the framework of optimization, which will replace numerical simulation and predict the performance of the design. But if the design is quite different from the existing designs in the database, this optimization iteration still needs the result of additional numerical simulation and then adds it to the surrogate model. Therefore, ideally, the optimization will run in the hybrid mode shown in Fig 3.

2.2. Machine Learning

The machine learning part is to train the surrogate model. It mainly consists of two parts: mesh encoder (Sec. 2.2.1) and conditional variational contour decoder (Sec. 2.2.2).

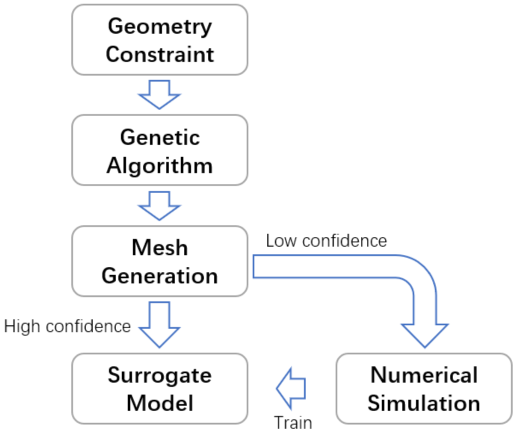


Figure 3: The pipeline of the optimization process using the surrogate model.

2.2.1. GNNs-based Mesh Encoder

Inspired by a mesh autoencoder [11], the mesh encoder used in this study compresses the surface mesh to a latent vector with a combination of different hidden layers. In this study, the surface mesh of the design is defined by the coordinates of n vertices and edges, M=(V,E). V is the n vertices in the Euclidean space, which is a n×3 vector. The edges, E, are represented by the sparse adjacency matrix A. Its size is n×n, where $A_{ij} = 1$ denotes a connection between vertex i and vertex j. Otherwise, $A_{ij} = 0$. The most important layers used in the mesh encoder is the fast spectral convolution layer. The mesh convolution operator $*$ is defined as a Hadamard product in Fourier space:

$$x * y = U((U^T x) \odot (U^T y)) \tag{1}$$

To reduce the computational cost, convolution is conducted by a kernel g_θ with Chebyshev polynomial of order K.

$$g_\theta\,(L) = \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{L}) \tag{2}$$

where $\widetilde{L} = 2L/\lambda_{max} - I_n$ is graph Laplacian matrix. It is defined as L=D-A, where diagonal matrix $D_{ii} = \sum_j A_{ij}$. And θ_k are the coefficients of the

Chebyshev polynomials. T_k can be expressed as:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \tag{3}$$

with the initial condition $T_0 = 1$ and $T_1 = x$. This represents a Chebyshev polynomial of order k.

With the mesh filter shown above, the fast spectral convolution layer can be expressed as the following equation with n$\times F_{in}$ input and n$\times F_{out}$ input

$$y_j = \sum_{i=1}^{F_{in}} g_{\theta_{ij}}(L)x_i \tag{4}$$

where y_j means the j^{th} feature.

Another important layer used in the mesh encoder is the mesh sampling layer, which includes the down-sampling layer and up-sampling layer in autoencoder [11]. The mesh sampling layer is to represent mesh in multi-scales so that convolution layers can capture the local and global geometric information. In this study, only a down-sampling layer is used in the surrogate model. The down-sampling operation is conducted by a transform matrix $Q_{down} \in \{0,1\}^{n\times m}$, where m is the number of vertices in the original mesh and n is the number of vertices in the down-sampled mesh. $Q_{down}(p,q) = 1$ means the q-th vertex is kept during the down-sampling, while $Q_{down}(p,q) = 0$ means the vertex is discarded. The transform matrix is optimized to minimize the surface error by quadric matrices [12].

2.2.2. CNNs-based Contour Autoencoder

Contour decoder is built by CNNs-based CVAE. AutoEncoder (AE) uses CNNs to compress graphical data to a latent vector and then reconstruct the graph with the latent vector. The neural network is trained to reconstruct the graphs with less error. Variational AutoEncoder (VAE) uses variational inference to estimate the latent vector rather than directly encoding from input graph [13]. The latent vector z can be estimated by observation vector x using the following equation:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{5}$$

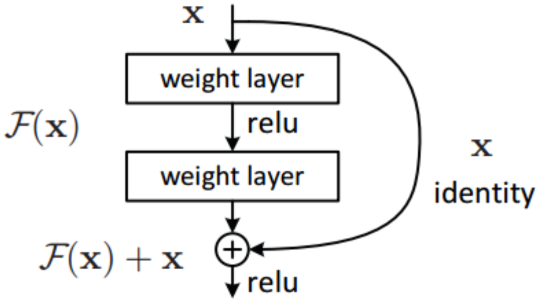

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4: A basic block of Residual neural network. Figure adapted from [14]

However, $p(z|x)$ is usually very difficult to compute directly. Therefore, another distribution $q(z|x)$ is used to approximate the $p(z|x)$. The Kullback-Leibler divergence is used to measure the difference between two probability distributions, which will be minimized during the training process. CVAE adds conditions into the latent distribution so that different classes of input data are categorized into different groups. In this study, conditions (blade passages index) are added into the latent distribution twice to label the input data.

In the CVAE, the contour decoder used Residual neural Network (ResNet), a kind of classical artificial neural network, which is inspired by pyramidal cells in the cerebral cortex. ResNet simulates this by building shortcuts to skip some layer, rather than passing information layer by layer. Fig 4 shows a basic block of ResNet. F(x) is to fit the residual between x and target mapping H(x) rather than directly fitting H(x). It is easier for the optimizer to minimize the residual to zero [14]. In this study, more hidden layers are added to fit the highly nonlinear relationship between input and output, but the performance decreases rapidly with more layers. To solve the degradation problem, ResNet is adapted because it can pass information from front layers to rear layers, which reduces the loss of information in the hidden layers.

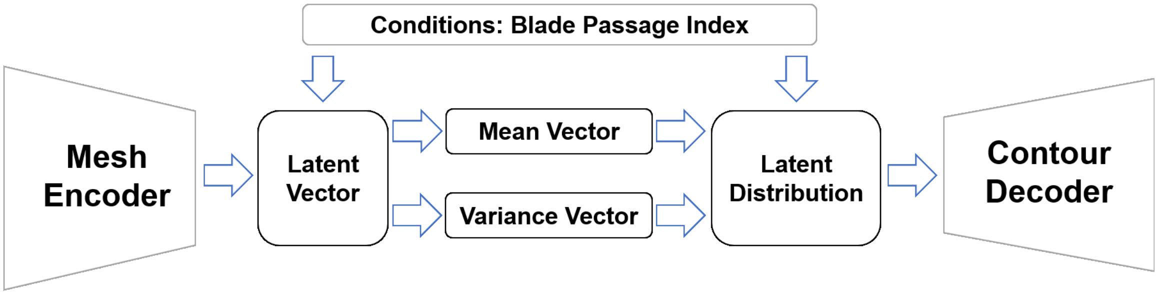


Figure 5: Neural network structure

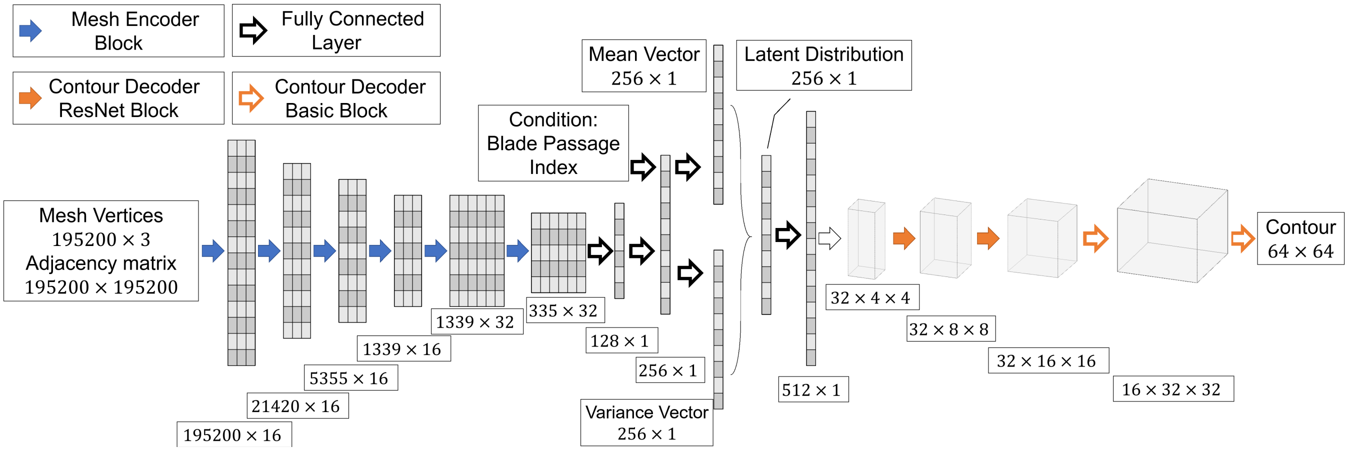


Figure 6: The change of feature dimension in the network.

3. Surrogate Model Setup

3.1. Neural Network Structure

The neural network is built under the framework of Pytorch. Fig 5 shows the main structure of network. Fig 6 shows the change of feature dimension through the network. The input is 10 surface meshes, which have 195200 vertices in total. All the samples need to be interpolated to the same number of mesh vertices to represent the geometry at the same details level. Therefore, the input data is the coordinates of vertices (195200×3) and adjacency matrix (195200×195200). The mesh encoder has 6 mesh encoder blocks. The size of filters in the front 4 blocks is 16, and it is 32 in the rear 2 blocks. A larger filter size is to capture global geometric features. After the mesh encoder, the mesh is compressed into a 128×1 latent vector. Conditions (blade passage index) are added to the latent vector. Then, two fully connected layers are used to estimate the mean vector and variance vector of latent distribution. Conditions (blade passage index) are added into the latent distribution again. The latent distribution is reshaped to be the input of the contour decoder. In the contour decoder, the first two blocks are ResNet blocks, and the rear two blocks are basic blocks.

3.2. Mesh Encoder Block

The structure of mesh encoder block is illustrated in Fig 7. One basic block is consist of a Chebyshev convolution layer, a normalization layer (batch normalization), an activation layer (the rectified linear unit function), a down-sampling layer and a pooling layer (max pooling layer). The mesh encoder is consist of several these basic blocks. The number of blocks depends on the size of mesh vertices and latent vector. More basic block means a smaller latent vector, which contains less geometric information. But larger latent vector needs more training cases to prevent over-fitting. Inside the basic block, the Chebyshev convolution layer is to scan the vertices with Chebyshev polynomial filter and convert them to a vector. The normalization layer is to normalize the value of vectors in the same batch. And the activation layer is to amplify the differences

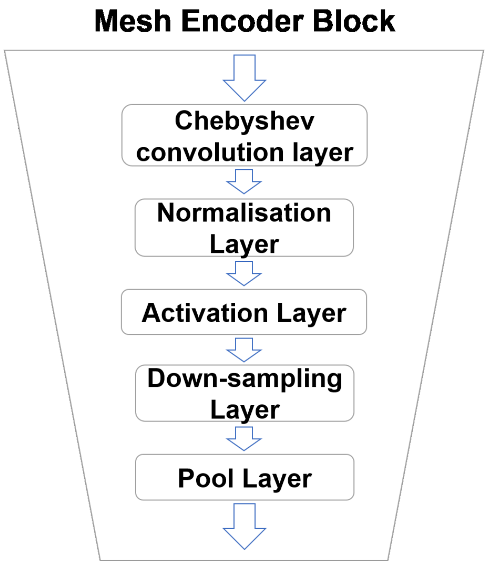


Figure 7: Mesh encoder basic block

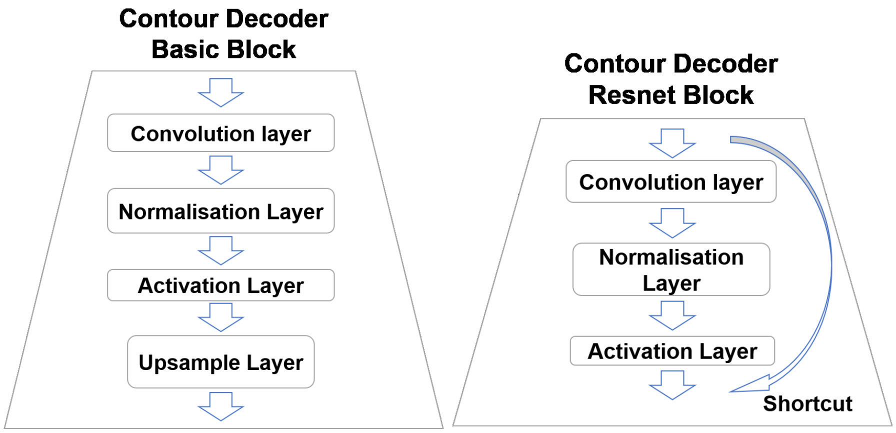
of values in the vector. The down-sampling layer is to drop out irrelevant vertices based on the transformation matrix. The pooling layer is to keep the most significant values and discard other values, which reduces the dimension of the vector. After several basic blocks, relevant geometric information is picked to form the latent vector.

3.3. Contour Decoder Blocks

Contour decoder consists of two types of blocks: ResNet block and basic block. ResNet block is shown in Fig 8a, which has three layers and one shortcut. The basic block is shown in Fig 8b, which has four layers. The contour decoder has two ResNet blocks, two basic blocks, and the number of blocks can increase or decrease according to the size of latent distribution and the contours.

3.4. Loss Function

In this study, the loss function of the neural network consists of three types of losses: mean squared error (MSE) loss, Kullback-Leibler divergence (KLD) loss and structural similarity loss.



a: Basic block

b: ResNet block

Figure 8: The basic block (a) and ResNet block (b) for contour encoder

Mean squared error measures the average of pixel-wise error between the contour predicted by model (Y_i) and contour predicted by numerical simulation $(\widehat{Y}_i)$. It is defined mathematically by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \tag{6}$$

KLD [15] measures the difference between one probability distribution and the reference probability distribution. In variational autoencoder, KL loss is the sum of all the KLD between the components in latent distribution and the standard normal distribution. With minimizing the KL loss, the latent distribution is closer to the standard normal, which can improve the interpolation and extrapolation ability of the surrogate model. KLD can be defined by:

$$\begin{aligned} KLD(p,q) &= -\int p(x)logq(x)dx + \int p(x)logp(x)dx \\ &= \frac{1}{2}log(2\pi\sigma_2^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}(1 + log2\pi\sigma_1^2) \\ &= log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2} \end{aligned} \tag{7}$$

As it is to measure the KLD between the components in latent distribution and the standard normal $(\sigma_2 = 1, \mu_2 = 0)$, it can be simplified as the following equation for convenience:

$$KL_{loss} = \sum_{i=1}^{n}(\sigma_i^2 + \mu_i^2 - log(\sigma_i) - 1) \tag{8}$$

where μ is the mean vector, σ is the variance vector.

Structural similarity loss, or Structural Similarity Index Measure (SSIM) [16], is a method to measure the similarity between two figures. Here, it is used to optimize the neural network to make predicted contours and simulated contours more structurally similar. It is defined by:

$$SSIM(Y_i, \widehat{Y}_i) = \frac{(2\mu_{Y_i}\mu_{\widehat{Y}_i} + c_1)(2\sigma_{Y_i\widehat{Y}_i} + c_2)}{(\mu_{Y_i}^2 + \mu_{\widehat{Y}_i}^2 + c_1)(\sigma_{Y_i}^2 + \sigma_{\widehat{Y}_i}^2 + c_2)} \tag{9}$$

where $\mu_{Y_i}, \mu_{\widehat{Y}_i}$ are the mean of Y_i and $\widehat{Y}_i$, $\sigma_{Y_i}^2, \sigma_{\widehat{Y}_i}^2$ are the variances of Y_i and $\widehat{Y}_i$, $\sigma_{Y_i\widehat{Y}_i}$ is the covariance of Y_i and $\widehat{Y}_i$, c_1, c_2 are two variables to stabilize the division with weak denominator.

Finally, the loss function of the surrogate model is defined by the following equation:

$$loss = k_1MSE + k_2KLD + k_3(1 - SSIM) \tag{10}$$

where three coefficients k_1, k_2 and k_3 are user-defined hyperparameters.

4. Demonstration of Non-parametric Surrogate model Method

To demonstrate the new method, a **Low Pressure Steam Turbine Exhaust System (LPES)** is used as an example. In this section, the mesh generation process and numerical simulation setup of this example are introduced in details.

4.1. Introduction Low-pressure Steam Turbine Exhaust System

LPES is designed to recover the kinetic energy leaving the low-pressure steam turbine and convert it to static pressure rise for the condenser. It usually consists of three parts: an axial-to-radial diffuser, an asymmetric collector, and an extension.

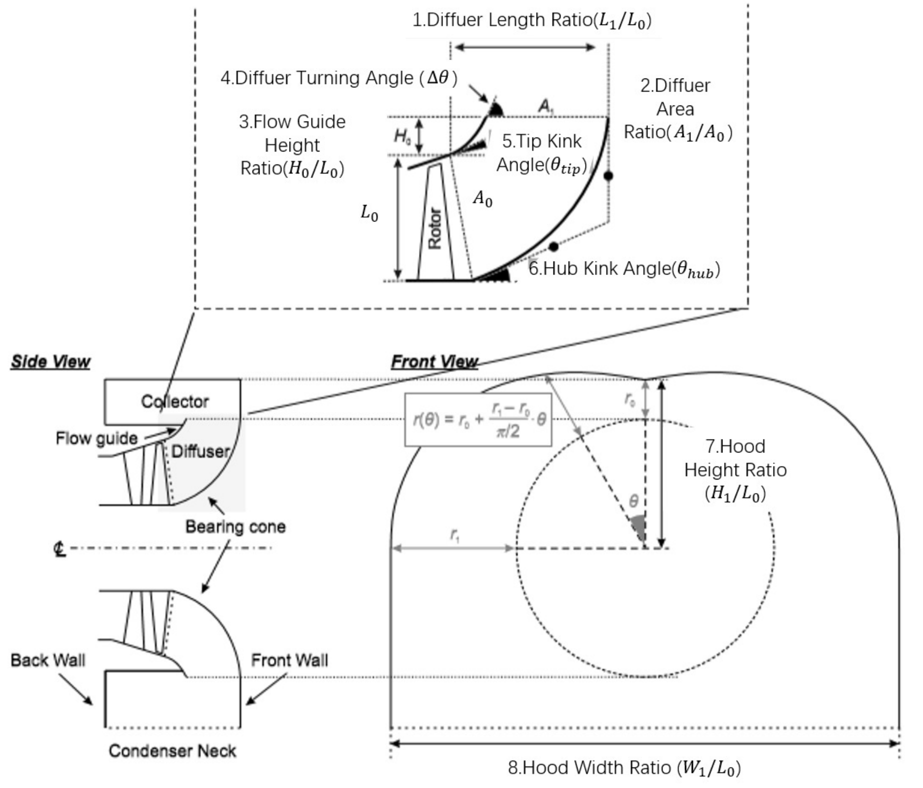

1.Diffuser Length Ratio(L_1/L_0)
4.Diffuser Turning Angle ($\Delta\theta$)
2.Diffuser Area Ratio(A_1/A_0)
3.Flow Guide Height Ratio(H_0/L_0)
H_0
A_1
5.Tip Kink Angle(θ_{tip})
A_0
L_0
Rotor
6.Hub Kink Angle(θ_{hub})
Side View
Front View
Collector
Flow guide
Diffuser
Bearing cone
Back Wall
Front Wall
Condenser Neck
$r(\theta) = r_0 + \frac{r_1 - r_0}{\pi/2} \cdot \theta$
r_0
r_1
θ
7.Hood Height Ratio (H_1/L_0)
8.Hood Width Ratio (W_1/L_0)

Figure 9: A typical down-flow type low pressure exhaust system for large steam turbine. Figure adapted from [17].

It is quite challenging to build surrogate model for LPES with **existing surrogate model method** because parameterization of geometry is problematic. Fig 9 shows a parameterization method to define the geometry of the system, namely, the diffuser length ratio ($\frac{L_1}{L_0}$), diffuser area ratio ($\frac{A_1}{A_0}$), flow guide height ratio ($\frac{H_1}{L_0}$), diffuser turning angle ($\Delta\theta$), tip kink angle (Θ_{tip}), hub kink angle (Θ_{hub}), hood height ratio ($\frac{H_1}{L_0}$) and hood width ratio ($\frac{W_1}{L_0}$). With these geometric parameters, there are still many geometric features missing, for example, the curvature distribution of the diffuser, the height change of the collector, the width change of the extension and many more details. If using more parameters to describe the geometry, it needs more training cases. And the irrelevant parameters falsely selected by users will cause the over-fitting problem and re-

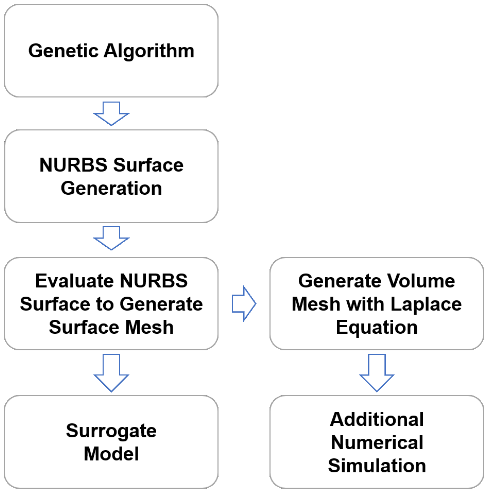

Genetic Algorithm
NURBS Surface Generation
Evaluate NURBS Surface to Generate Surface Mesh
Generate Volume Mesh with Laplace Equation
Surrogate Model
Additional Numerical Simulation

Figure 10: Framework of mesh generation process.

duce the accuracy of prediction. In summary, LPES is a proper example to demonstrate the advantages of new method.

In the dataset of this study, there are two types of geometries. One is defined by 95 parameters, the other one is defined by 66 parameters. The first one firstly defines the cross-section of diffuser and collector, and then revolves it with an ellipse equation to generate circumferential distribution. It also has asymmetric features in extension. The second one directly defines the cross-section along the axial direction with ellipse equations. These two types of geometries are used to test the ability to process geometries from different sources.

4.2. Mesh Generation

As shown in Fig 10, mesh generation starts from the coordinates of control points given by GA. The control points will be used to generate Non-Uniform Rational B-Spline (NURBS) surfaces, and then evaluate the NURBS surfaces to generate the surface mesh. These surface meshes are the input of surrogate model. Volume mesh will be generated by solving the Laplace equation:

$$
\begin{aligned}
\frac{\partial^2 x}{\partial i^2} + \frac{\partial^2 x}{\partial j^2} + \frac{\partial^2 x}{\partial k^2} = 0 \\
\frac{\partial^2 y}{\partial i^2} + \frac{\partial^2 y}{\partial j^2} + \frac{\partial^2 y}{\partial k^2} = 0 \\
\frac{\partial^2 z}{\partial i^2} + \frac{\partial^2 z}{\partial j^2} + \frac{\partial^2 z}{\partial k^2} = 0
\end{aligned}
\tag{11}
$$

where x, y, z are the coordinates of mesh vertices and i, j, k are indices of mesh vertices. The boundary condition is defined by the coordinates of surface mesh. Since the Laplace equation represents a potential field, equipotential lines do not intersect and are orthogonal at vertices. The volume mesh can be generated by solving x, y, and z coordinates potential field respectively. The mesh generation method used in this study is able to generate mesh for the fluid domain with the same topology, regardless of the change of geometry.

4.3. Numerical Simulation Setup

4.3.1. Simulation Domain

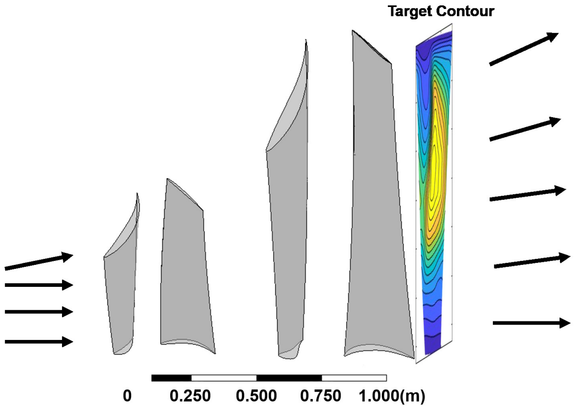


Figure 11: The geometry of the last two low pressure stages. Since the inlet boundary condition on the left-hand side is known, the surrogate model is to predict target contour at the right-hand side.

The fluid domain of numerical simulations includes: two low-pressure stages of turbine, an axial-to-radial diffuser, a collector and an extension. Fig 11 shows the geometry of two low-pressure stages, which is from a typical large

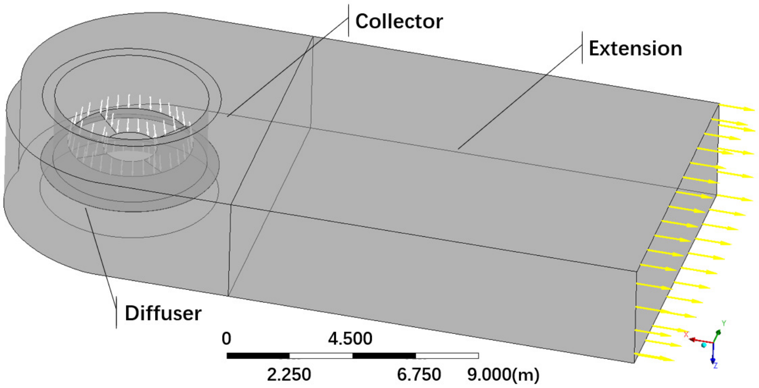


Figure 12: The geometry of exhaust hood. The white arrows represent inlet boundary and the yellow arrows represent outlet boundary.

steam turbine. It can generate a representative inlet boundary condition for the exhaust hood. Fig 12 illustrated the axial-to-radial diffuser, collector and extension in the downstream.

4.3.2. Simulation Setup

The solver used for the numerical simulations is Ansys CFX, which is a widely-used commercial CFD solver for the research community and industry. The simulation is a Reynolds-averaged Navier–Stokes (RANS) simulation, which uses $k - \epsilon$ turbulence model [18].

The inlet boundary condition is applied at the inlet of the simulation domain, which has total pressure and total enthalpy. And the outlet of the simulation domain is at the end of extension, which is applied with static pressure boundary conditions. To perform the part-load simulations, the total pressure is reduced at the inlet to reduce the mass flow rate and work output. The static pressure at the outlet is 6.2kPa due to the steam property at the condenser. For the property of the working fluid, steam, IAWPS data has been used, which is embedded in CFX and also widely used in the research community and industry.

Another setup worth mentioning is the interface treatment method between stages and the inlet of the diffuser. Because the downstream of the last two stages are asymmetric, it is necessary to model the circumferential non-uniformity. Multiple mixing plane method from literature [18] is used in this study. As shown in Fig 13, only four-blade passages are simulated to generate inlet flow conditions for a low-pressure exhaust hood, which means one blade passage is responsible for a 90-degree section of the diffuser. The outlet boundary conditions of blade passages are copied to cover the section. Multiple mixing plane method, though losing accuracy with only 4 blade passages, can reduce the computational cost considerably.

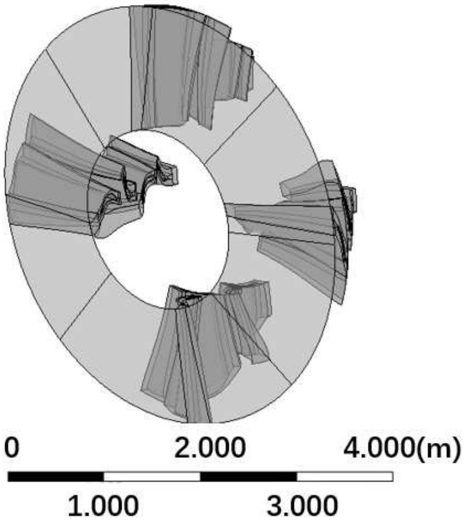


Figure 13: Demonstration of multiple mixing plane method. Four blade passages are modelled. Each of them generates inlet boundary condition for a 90-degree section of exhaust hood.

4.4. Processing of Numerical Simulation Results

The objective of GA-based optimization is to increase power output of the last two low pressure steam turbine stages. It is calculated by the difference between power goes through inlet and outlet of the last two stages, which is summed value of all the elements of contour map. Assuming the system is adiabatic, the power of each element is obtained by the product of local total enthalpy and local mass flow rate:

$$\begin{aligned}\Delta\dot{W} &= \dot{m}h_{02} - \dot{m}h_{01} \\ &= \sum_{i=1}^{n} \dot{m}_{i_2} h_{i_2} - \sum_{i=1}^{n} \dot{m}_{i_1} h_{i_1}\end{aligned} \tag{12}$$

Since the inlet boundary condition is known in the simulation, only the total enthalpy contour and mass flow rate contour at the outlet of the last two stages are extracted from numerical simulations to generate power contour as shown in Equation 12, which will also be the output of the surrogate model. Because of the multiple mixing plane method, there are four blade passages for each simulation, and five workload conditions for each design. Admittedly, there is certainly some uncertainties in the numerical simulations, but it is not primary concern in this paper since the key of this study is to develop a new surrogate model method.

5. Test and Result

5.1. Test Setup

To test the surrogate model, 32 cases are randomly selected from 582 cases. Therefore, there are 640 contours to predict in total because each case has 4 blade passages and 5 workload conditions. And during the training, there is a k-fold cross validation, which means 55 cases of the remaining 550 cases will be used for validation every 10 epochs.

The result of surrogate model are contours of power at the outlet of the last rotor. There are 4 blade passages modelled for each simulation, and 5 workload condition for each design. This means there are 20 contours for one design.

As mentioned above, there are two types of sub-datasets. One has 333 training cases, and the other one has 249 training cases. During the training and test, two sub-datasets are mixed with each other to test the ability of processing different kinds of geometries. In the test, there are two test sub-datasets, both of which contains 64 contours.

5.2. Test Metrics

Similar with the loss function (Sec. 3.4), the performance of surrogate model is also evaluated by three different metrics:

- **Mean squared error** is used to measure the average of pixel-wise error between the model prediction contour (Y_i) and the ground truth contour ($\widehat{Y}_i$);
- **SSIM** is used to evaluate the structure similarity level between the model prediction contour (Y_i) and the ground truth contour ($\widehat{Y}_i$). SSIM is an index between 0 and 1, where the SSIM = 1 two contour are identical;
- **Summed value error (SVE)** measures the differences of summed value, which is defined by the following equation:

  $$\mathrm{SVE} = \frac{\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \widehat{y}_i}{\sum_{i=1}^{n} \widehat{y}_i} \tag{13}$$

  where y_i is the value of ith pixel of predicted contour, and $\widehat{y}_i$ is the value of ith pixel of target contour. This error also indicates the error in the predicting averaged value, like averaged pressure, averaged temperature, averaged velocity of a surface.

5.3. Test Result

Fig 14 shows some representative results of the test cases, which presents some typical flow features of 5 workload conditions. Contour results are illustrated in Fig 14, which shows that it can predict most flow features of contours. For example, the separations at the hub part are well captured in the contours of 50% workload condition. And the vortices are also well predicted in the contours of 70% workload condition.

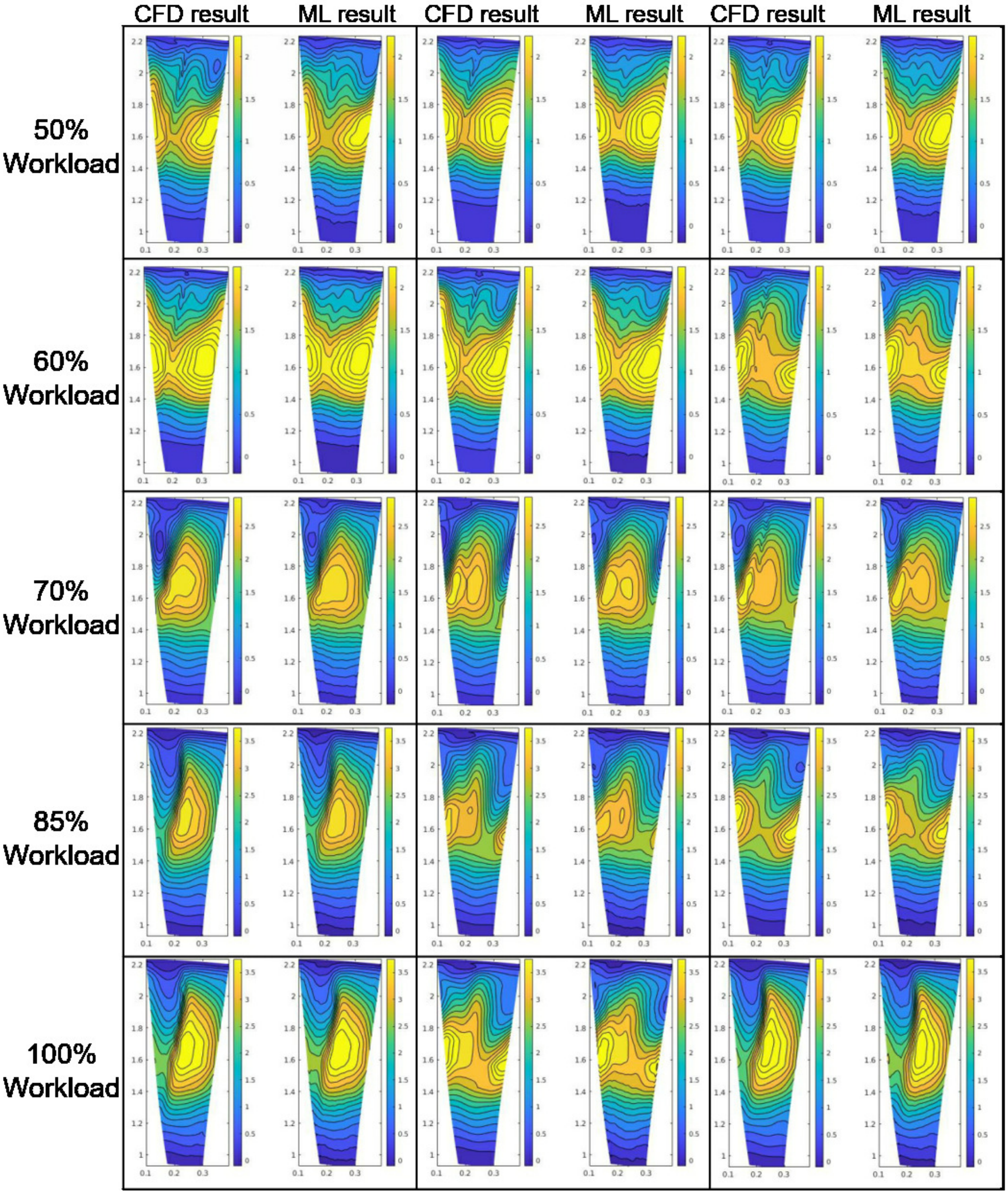


Figure 14: Contour results of computational fluid dynamics (CFD) and machine learning (ML) of some typical flow features at five workload conditions (50%, 60%, 70%, 85% and 100%). Unit: $1 \times 10^4 kJ/s$

Table 1: Summary of Result. Sub-dataset1 are geometries defined by 95 parameters and sub-dataset2 are defined by 66 parameters. Details in Sec. 4.1

Work Condition	Performance Metrics	Dataset	Sub-dataset1	Sub-dataset2
50%	MSE Loss	0.0047	0.0038	0.0056
	Similarity Measure	0.9541	0.9520	0.9561
	Summed Value Error	0.0060	0.0068	0.0052
60%	MSE Loss	0.0068	0.0042	0.0094
	Similarity Measure	0.9482	0.9517	0.9447
	Summed Value Error	0.0096	0.0122	0.0070
70%	MSE Loss	0.0017	0.0016	0.0019
	Similarity Measure	0.9632	0.9602	0.9661
	Summed Value Error	0.0089	0.0095	0.0082
85%	MSE Loss	0.0016	0.0017	0.0016
	Similarity Measure	0.9649	0.9618	0.9679
	Summed Value Error	0.0090	0.0104	0.0076
100%	MSE Loss	0.0019	0.0019	0.0020
	Similarity Measure	0.9668	0.9641	0.9695
	Summed Value Error	0.0094	0.0108	0.0081
Mean	MSE Loss	0.0033	0.0026	0.0041
	Similarity Measure	0.9594	0.9580	0.9609
	Summed Value Error	0.0086	0.0099	0.0072

Table 1 shows the result of all 640 test contours. Among them, there are two sub-datasets that contain different types of geometries, both of which contain 16 cases and 320 contours. In the Table 1, the averaged SVE of dataset is 0.86 %, which shows it is able to predict performance for optimization. The SSIM of sub-dataset1 and sub-dataset2 are 0.9580 and 0.9609 respectively, which has no significant difference. This results proves the generalization capability of the proposed method in different kinds of geometries.

In the Fig 15, x-axis represents the results of surrogate model and y-axis

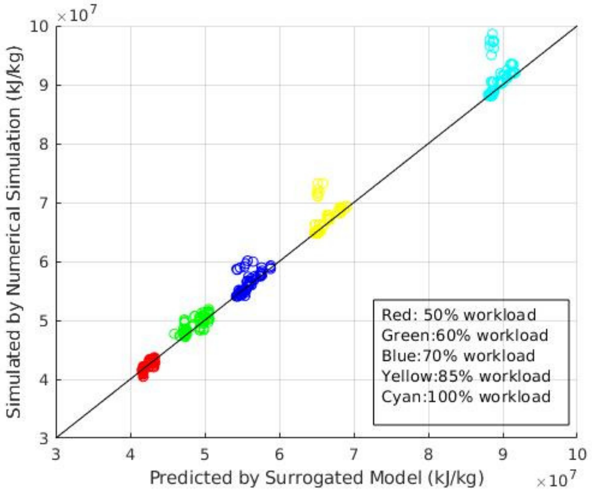


Figure 15: Comparison between results of numerical simulation and surrogate model. The closer to the line $y = x$ means higher accuracy.

represents the results of numerical simulation. The results of 5 workload conditions is categorized into 5 kinds of colors. Fig 15 shows most of points are close to $y = x$, which means lower prediction error. There are 8 points for each workload condition significantly further to $y = x$. They are result of 2 designs, which is quite different from designs in the training dataset.

6. Discussion

The new surrogate model method established a mapping relationship between the surface mesh of fluid domain and two-dimensional distribution of flow variables. The application of this new method can be extended beyond the area of aerodynamics optimization. Since it can process both structural and unstructural mesh, it is also applicable in various problems in different fields, which need to solve partial differential equations, like finite element analysis and electromagnetic analysis.

This method can also be used as an inverse method. It can be achieved by exchanging the input and output of the mapping relationship built in this paper.

To be more specific, users import the two-dimensional distributions of physical properties they want into the contour encoder, and the designs are generated by the mesh decoder.

7. Conclusion

This study presents a novel non-parametric surrogate model method, which is demonstrated using LPES. The new method directly takes surface mesh as input, which reduces the uncertainties introduced by manual parameterization and loss of geometric information. This feature gives the new method great advantages in building surrogate model for designs with complex geometries. In the test, the average summed value error of 128 contours is 0.86 %.

This method also shows high flexibility and compatibility. Because the input of new method is surface mesh, it can take geometries with the same topology as database. This means it is compatible with geometry defined by different methods. It is very useful for further increasing the size of database of surrogate model using variable sources of data.

Compared with existing surrogate model methods, this new method can also predict two-dimensional distribution of variables (contour) based on surface mesh. Contours can help designers to discover physical mechanism, improve their designs and many other purposes. In the test, the average similarity score of 640 contours is 0.9594, which can reproduce most flow features.

The essence of this new method is the creation of a unique mapping relationship between the surface mesh of simulation domain and two-dimensional distribution of variables. With this method, the optimization process can be accelerated by utilizing the result of database to evaluate the performance of new designs.

Acknowledgment

The financial support of Shanghai Electric is gratefully acknowledged.

References

[1] G. Claeskens, N. L. Hjort, Lack-of-fit and goodness-of-fit tests, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2008, p. 227–247.

[2] R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[3] S. Sakata, F. Ashida, M. Zako, Structural optimization using kriging approximation, Computer methods in applied mechanics and engineering 192 (7-8) (2003) 923–939.

[4] H.-M. Gutmann, A radial basis function method for global optimization, Journal of global optimization 19 (3) (2001) 201–227.

[5] T. Mengistu, W. Ghaly, Aerodynamic optimization of turbomachinery blades using evolutionary methods and ann-based surrogate models, Optimization and Engineering 9 (3) (2008) 239–255.

[6] A. Lal, B. Datta, Development and implementation of support vector machine regression surrogate models for predicting groundwater pumping-induced saltwater intrusion into coastal aquifers, Water Resources Management 32 (7) (2018) 2405–2419.

[7] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, N. Chervyakov, Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, Mathematics and Computers in Simulation 177 (2020) 232 – 243.

[8] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (1) (2009) 61–80.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems.

[10] D. K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis 30 (2) (2011) 129–150.

[11] A. Ranjan, T. Bolkart, S. Sanyal, M. J. Black, Generating 3D faces using convolutional mesh autoencoders, in: European Conference on Computer Vision, 2018, pp. 725–741.

[12] M. Garland, P. S. Heckbert, Surface simplification using quadric error metrics, in: Conference on Computer Graphics and Interactive Techniques, 1997, pp. 209–216.

[13] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians, Journal of the American statistical Association 112 (518) (2017) 859–877.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[15] S. Kullback, R. A. Leibler, On information and sufficiency, The Annals of Mathematical Statistics 22 (1) (1951) 79–86.

[16] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: Asilomar Conference on Signals, Systems & Computers, Vol. 2, Ieee, 2003, pp. 1398–1402.

[17] B. Ding, Aerodynamics of low pressure steam turbine exhaust systems, Ph.D. thesis, University of Cambridge (2019).

[18] Z. Burton, G. L. Ingram, S. Hogg, A literature review of low pressure steam turbine exhaust hood and diffuser studies, Journal of Engineering for Gas Turbines and Power 135 (6).