doi:10.20944/preprints202104.0762.v3

Uncertainty quantification and sensitivity analysis of Parameterization-free surrogate model

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

Surrogate model based optimization method is widely-used to accelerate the design and optimization process [1]. The input of regression model used in the surrogate model are numbers, which requires users to parametrize the geometries. In this paper, a new parameterization-free surrogate model is introduced and its corresponding uncertainty quantification and sensitivity analysis method are discussed. The input of new surrogate model methods is surface mesh of simulation domain. Graph Neural Networks (GNNs) is used to extract geometric information, and Convolutional Neural Networks (CNNs) is used to predict contours. This framework bypasses parameterization,as a consequence, uncertainties introduced by manual parameterization is reduced. However, such changes compared with conventional surrogate model methods impose great challenge on uncertainties quantification and sensitivity analysis. Uncertainties quantification in this paper means the error bar of prediction results, which is calculated by Gaussian Process Regression method in current surrogate method. In this paper, a new quantification method achieved by Kullback-Leibler divergence (KLD) is introduced. And the sensitivity analysis is conducted by Automatic Differentiation, which gives a Jacobian matrix of inputs. The method and anal-

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)

ysis mentioned above are demonstrated by a low-pressure steam turbine rotator and its exhaust system.

Keywords: surrogate model method, aerodynamic optimization, graph neural networks, parameterization, steam turbine exhaust system

1. Introduction

Despite the rapid increase of renewable energy production, around 60% of the world electricity will still be generated by fossil fuel fired power plants by 2040 [2]. Fig 1 shows the electricity generation by energy source in USA from 24/06/2021 to 01/07/2021. From Fig 1, it can be seen that some thermal plants vary their load more frequently to offset the change of renewable energy (solar,wind and tidal power). As a consequence, maintaining efficiency when operational condition varies have become a big concern in the design of some steam turbines.

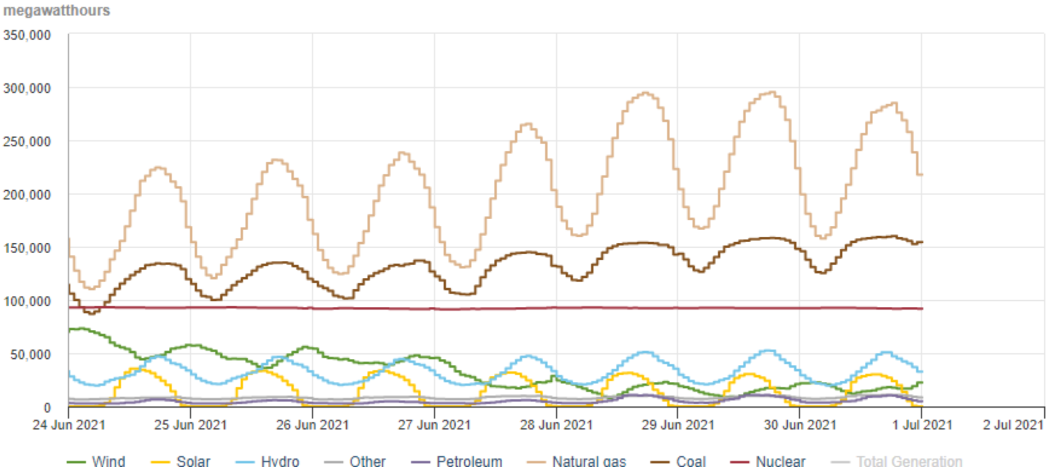


Figure 1: U.S. electricity generation by energy source from 24/06/2021 to 01/07/2021 [3]

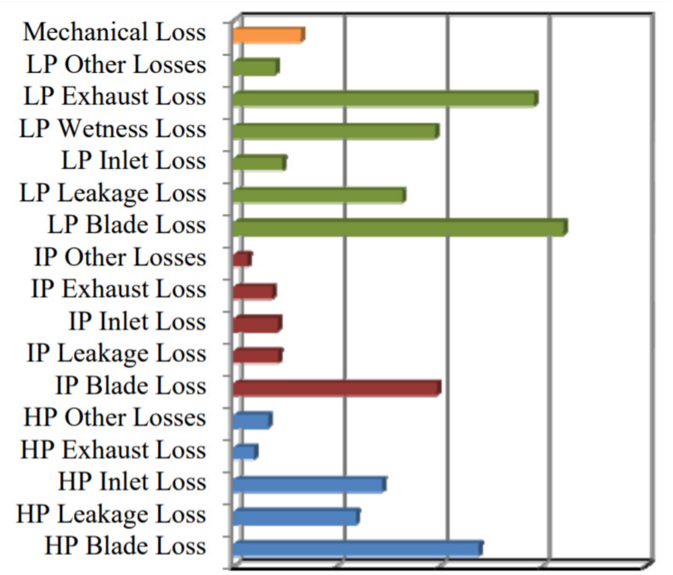


Figure 2: Breakdown of the typical losses of a recent large-scale steam turbine (relative fractions of each loss) [4]

To further improve the operational flexibility and efficiency of steam turbine, exhaust hood loss is an important part that cannot be neglected. As Fig 2 shows, LP exhaust loss is the second highest loss among all losses. And despite the LP blade loss is still the highest, the actual margin for further improvement is rapidly diminishing after intensive research and development efforts in the past decades. In contrast, exhaust hood has relatively simpler structure. Its aerothermal performance can be improved with less concerns from mechanical aspects. On the other hand, the multi-objective optimization of low-pressure steam turbine exhaust system is quite expensive due to the large size of simulation domain and multiple workload conditions. And the design target could be frequently changed according to the user specifications, which makes surrogate model more beneficial to reduce the computational cost by utilizing numerical simulation results of previous optimizations.

In the recent decades, a lot of surrogate model methods appeared, which can be loosely categorized into: (1) Polynomial Response Surface Method [5];

(2) Kriging Model [6]; (3) Radial Basis Function and Extended Radial Basis Function [7]; (4) Artificial Neural Network [8]; (5) Support Vector Machine [9]. Fig **??** shows the process of existing surrogate model method, and one key step in this process is manual parameterization, which is to choose some important geometric parameters according to the experience of researchers to describe the geometry . In this step, if too few parameters were used, some geometric information will be lost because it is insufficient to describe geometric details with high-order surfaces (e.g., airfoils and blades). On the other hand, using too many parameters or choosing irrelevant parameters will cause problems of overfitting [10]. It is recognized by the authors that manual parameterization is the bottleneck that prevents further improvement of the performance of surrogate model method in terms of both accuracy and flexibility.

Recently, some neural networks and network structures have been applied to the field of computational mechanics, which inspired the idea of **parameter-free surrogate model method**. A widely used neural network is the Physical Information Neural Network (PINN). It has been applied to solid mechanics [11] and fluid mechanics [12]. PINN is built with multilayer perceptrons. Its inputs are still manually defined parameters. Another widely used neural network, Convolutional Neural Network (CNN), can predict two-dimensional data [13], but cannot extract geometric information. As a popular network structure, autoencoders can compress and reconstruct data [14]. Combined with CNNs, autoencoders can extract information from images and reconstruct images.

This paper proposes a new surrogate model method that uses GNNs, and a Conditional Variational Autoencoder (CVAE). Figure **??** shows the differences between numerical simulations used as analysis tools, existing surrogate model methods, and the proposed new surrogate model method. The new method can process the boundary surfaces of fluid domains from surface meshes used in numerical simulations and extract relevant geometric features according to their importance to the results. Compared to existing surrogate models, the new method contains less uncertainty introduced by manual parameterization. This new approach also allows the use of different types of designs from different

doi:10.20944/preprints202104.0762.v3

sources, as the geometric input to the model is a surface mesh, rather than user-defined parameters. In addition, the new method is able to predict the two-dimensional distribution of variables (in the form of contour plots) by processing the image with **CNNs**.

The ability to predict a two-dimensional distribution of variables is achieved by applying CNNs. Through the combination of convolutional layers, information can be extracted from the image and the convolutional results can be identified using a multilayer perceptron [15]. In this study, CNNs is used to predict contours from the underlying distribution.

The ability to use face meshes is achieved by applying GNNs. The graph operation nature of GNNs enables it to handle non-Euclidean domains by defining the connectivity of grid points, whereas CNNs can only handle regular Euclidean data like graphs. In existing GNN variants [16], GNNs are divided into three types: cyclic GNNs, spatial GNNs and spectral GNNs. In this study, the surrogate model is built based on Spectral GNNs because of its advantages in extracting features from grids with a large number of vertices [17]. Spectral GNNs is based on signal processing theory. The key step, the convolution operation, is done by the Chebyshev polynomial approximation [18]. In this study, GNNs is used to extract geometric information by optimizing parameters in a neural network through backpropagation of losses. This is to select relevant information based on the feedback of prediction error, avoiding geometric information loss and overfitting problems. Furthermore, since GNNs are able to handle non-Euclidean data, the input to the new surrogate model can be both unstructured grids and structured grids.

The framework for this new surrogate model approach is CVAE. It is an extension of Variational Autoencoder (VAE). VAE is an extension of autoencoders. Its latent distribution is regularized to increase the power of interpolation and extrapolation [19]. On this basis, conditions are added to the underlying distribution to divide the samples into different groups. In this study, CVAE was used to normalize and classify the design space.

doi:10.20944/preprints202104.0762.v3

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 based on Genetic Algorithm (GA), which is to optimize the performance of LPES at five workload conditions. At the same time, the results of optimization are used as the database for the surrogate model because the designs generated by the optimizer are similar to designs that will appear in future optimization. In this study, the optimization part is only introduced as a tool for accumulating database.

2.1. Optimization

For optimization, the surrogate model is used to evaluate the performance of designs, which will replace numerical simulation and predict the performance of the design. But when the design is out of existing design space of the database, this optimization iteration still needs the results of additional numerical simulations and then the new results will be added to the surrogate model. Therefore, ideally, the optimization will run in the hybrid mode shown in Fig 3.

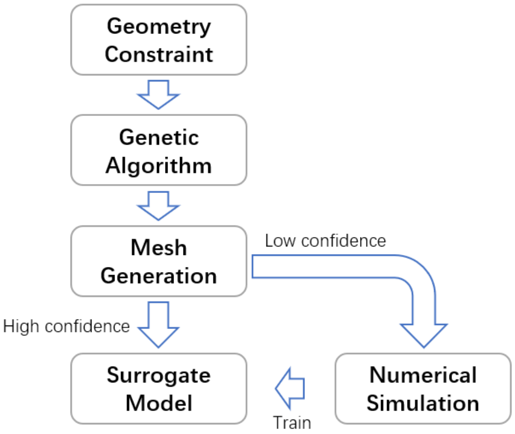


Figure 3: The pipeline of the optimization process using the surrogate model.

2.2. Machine Learning

The machine learning module is used for training the surrogate model. It consists of two main parts: mesh encoder and conditional variational contour decoder.

2.2.1. GNN-based Mesh Autoencoder

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.

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

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×F_{in} input and n×F_{out} output

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

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 [20]. In the encoder, important information is chosen by down-sampling layer and irrelevant information will be discarded. The convolution layers used in this study represents mesh in multi-scales so that mesh sampling layer can capture the local and global geometric information. 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 transformation matrix is optimized to minimize the surface error by quadric matrices [21].

2.2.2. CNN-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 [22]. The latent vector z can be estimated by observation vector x using the following equation:

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

However, $p(z|x)$ is usually very difficult to compute directly [19]. Therefore, another distribution is used to approximated $p(z|x)$ in the training process. The Kullback-Leibler divergence is used to measure the difference between two probability distributions, which is to 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 type of classical artificial neural network, which is inspired by pyramidal cells in the cerebral cortex. ResNet simulates this structure 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 [23]. In this study, more hidden layers are added to fit the highly non-linear relationship between input and output, but the performance

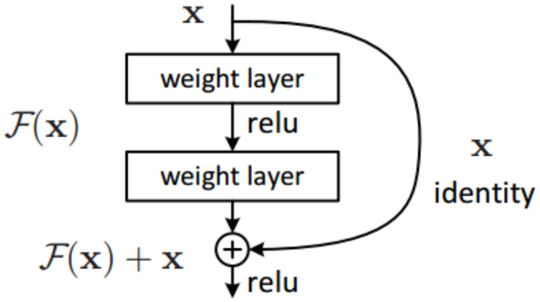

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4: A basic block of Residual neural network. Figure adapted from [23]

of neural network 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.

3. Surrogate Model Setup

The neural network is built under the framework of Pytorch. Fig **??** shows the main structure of network. Fig **??** shows the change of feature dimension through the network. The input is 10 bounding surface meshes of fluid domain, 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. A smaller filter size (16) in the front four blocks can capture local geometric features, and a larger filter size (32) in the rear two blocks can capture global geometric features better. 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.

3.1. Mesh Encoder Block

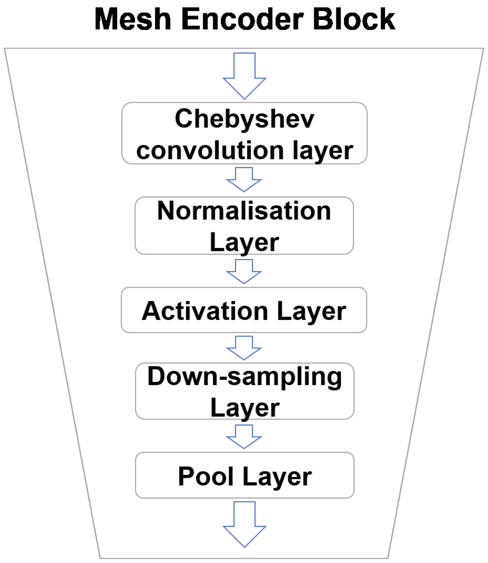
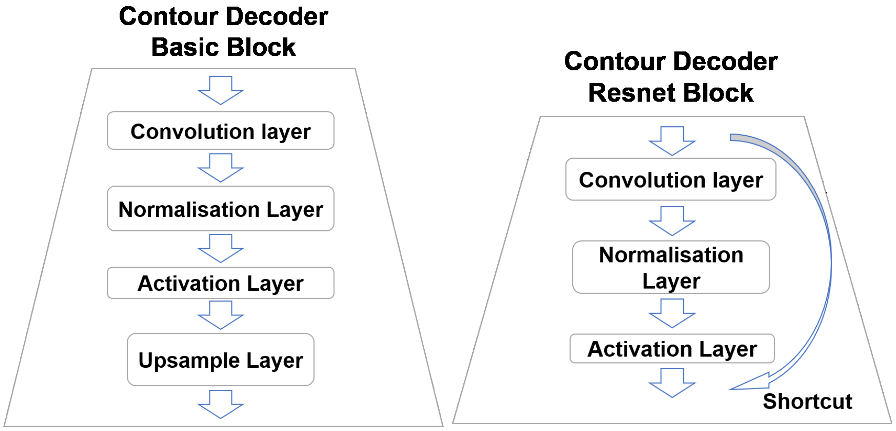


Figure 5: Mesh encoder basic block

The structure of mesh encoder block is illustrated in Fig 5. 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 such basic blocks. The number of blocks depends on the size of mesh vertices and latent vectors. 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 change the dimension of mesh vertices vectors. The normalization layer is to normalize the value of vectors in the same batch. And the activation layer is to zero the negative value in the vector. The down-sampling layer is to drop out irrelevant vertices based on the transformation matrix. The pooling layer is to



a: Basic block

b: ResNet block

Figure 6: The basic block (a) and ResNet block (b) for contour encoder

keep the largest value and discard other values in the filter, which reduces the dimension of the vector. After several basic blocks, relevant geometric information is picked to form the latent vector. During a training process, the optimizer will optimize the parameters of each layer according to the loss function. In this way, the mesh encoder can keep important geometric information and discard insensitive information.

3.2. Contour Decoder Blocks

Contour decoder consists of two types of blocks: ResNet block and basic block. ResNet block is shown in Fig 6a, which has three layers and one shortcut. The basic block is shown in Fig 6b, which has four layers. The contour decoder has two ResNet blocks, two basic blocks, and the number of blocks can be increased or decreased according to the size of latent distribution and the contours. In these two blocks, convolution layer is to change the dimension of vector with convolutional filter.And then the vectors in the same batch are normalized by normalization layer. Activation layer is to zero the negative value in the vector. Upsample layer is to increase the number of elements in the vector.

3.3. 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.

MSE measures the average of pixel-wise error between the contours predicted by model (Y_i) and contours 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 [24] 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) [25], 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

doi:10.20944/preprints202104.0762.v3

more structurally similar. It is defined by:

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

where $\mu_{Y_i}, \mu_{\widehat{Y}_i}$ are the mean of Y_i and $\widehat{Y}_i$, $\sigma^2_{Y_i}, \sigma^2_{\widehat{Y}_i}$ 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:

$$loss = k_1 MSE + k_2 KLD + k_3(1 - SSIM) \tag{10}$$

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

4. parameterization-free Surrogate model

In this section, the process of building a parameterization-free surrogate model for **Low Pressure Steam Turbine Exhaust System (LPES)** is introduced in details. This section includes the introduction of geometry, mesh generation, numerical simulation setup and post process of numerical simulation results.

4.1. Mesh Generation

Fig 7 shows the framework of mesh generation process. Mesh generation starts from the coordinates of control points given by GA. The control points are 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 meshes are generated with the method similar to [26], which is by solving Laplace equation:

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

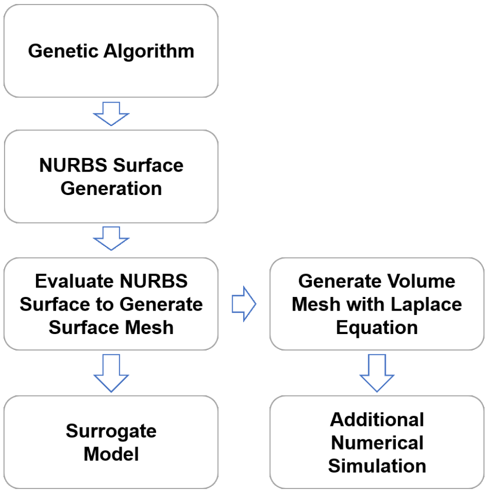

Genetic Algorithm
NURBS Surface Generation
Evaluate NURBS Surface to Generate Surface Mesh
Generate Volume Mesh with Laplace Equation
Surrogate Model
Additional Numerical Simulation

Figure 7: Framework of mesh generation process.

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. And the refinement of mesh can be done by refining the surface mesh, and then the volume mesh is also refined because surface mesh is the boundary condition of Laplace equation. The surface mesh can be refined by re-evaluating the NURBS surface with different distribution functions. This mesh generation method is also able to generate meshes for the fluid domain with the same topology, regardless changes in geometry.

4.2. Numerical Simulation Setup

4.2.1. Simulation Domain

The researches about LP steam turbine exhaust system began in the late 1960s. In the early stage, research community has performed 1D analysis by theoretical analysis or experiments, which has proved that conventional diffuser data is not applicable to LP exhaust system.

In the recent decades, 3D numerical simulation has been applied in studying the flow features in the LP exhaust system, which gives a deeper understanding about the factors that influence the performance of system. The literature review written by Burton [**?**] has summarised the achievements by new research tools in the recent years.

Since this report is about the inlet flow condition of LP exhaust system at part load condition, the literature review, therefore, will more focus on interaction between last stage and hood and part load performance.

5. Flow features in the LP exhaust system

Identifying flow features is the first step to understand the physics and loss mechanism in the LP exhaust system because flow features like separations and shock waves will largely affect the performance of the system.There are many literature about the flow features inside the exhaust system and the reason why these flow features form, which help researchers to further understand the source of loss.

The flow inside the LP exhaust system is three-dimensional and complicated. But generally, it can be divided into two separate parts: axial-to-radial flow in the diffuser and down flow to the condenser

At the inlet of diffuser, the flow is highly non-uniform in radial direction and circumferential direction, which is generated by last stage and affected by the axis-asymmetric flow. In addition, the high turning angle of diffuser also contributes to the separations in the LP exhaust system.Fig **??** shows the flow features of axial-to-radial flow, including bearing cone separation, flow guide separation, flow guide tip separation and additional vortices.

5.0.1. 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 [27].

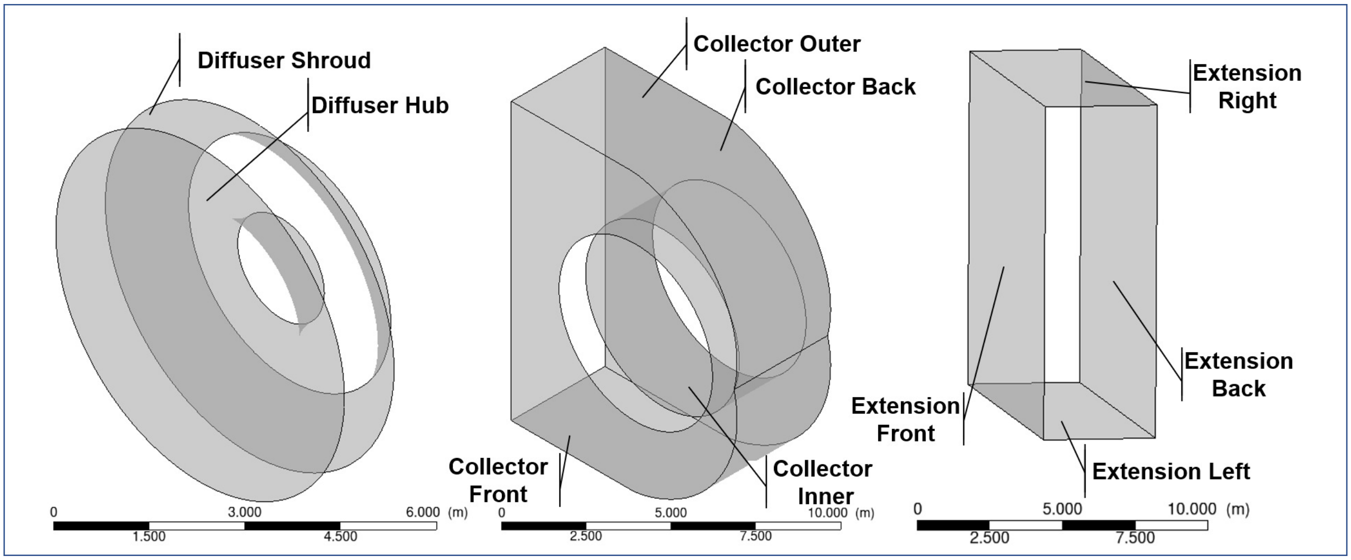


Figure 8: 10 surface meshes for surrogate model

Table 1: Summary of Mesh

Component	The Last Two Stages	Diffuser	Collector	Extension
Number of Vertices	1.72 millions	0.96 millions	0.67 millions	2.15 millions
Important Boundaries	Last Rotator Hub	Last Rotator Shroud	Last Rotator Blade Surface	Diffuser
$Y+$	15.76	59.55	17.70	77.97

The inlet boundary condition is applied at the inlet of the simulation domain, 255 which has total pressure and total enthalpy. The flow direction at the inlet is normal to boundary. And the outlet of the simulation domain is at the end of extension, which is applied with static pressure boundary conditions. The flow direction at the outlet is also normal to boundary. To perform the part-load simulations, the total pressure is reduced at the inlet to reduce the mass flow 260 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. The mesh size of each components and average $Y+$ of important boundaries is shown in Table 1.

Another setup worth mentioning is the interface treatment method between stages and the inlet of the diffuser. Because the downstream of the diffuser are asymmetric, it is necessary to model the circumferential non-uniformity. The multiple mixing plane method [27] is used in this study. Fig 9 shows only four-blade passages are simulated to generate inlet flow conditions for the low-pressure exhaust hood, which means the outlet boundary condition of one blade passage are copied to cover a 90-degree section of the diffuser. Multiple mixing plane method, though losing accuracy with only 4 blade passages, can reduce the computational cost considerably.

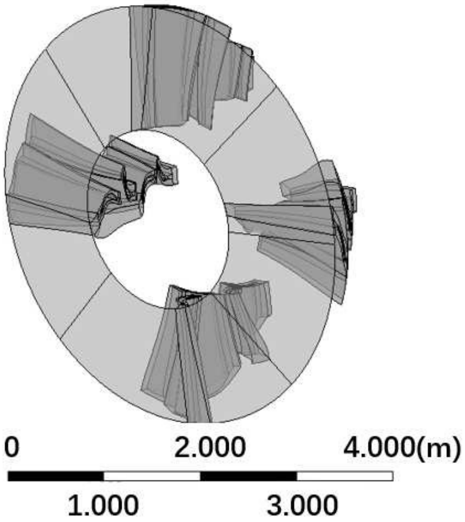


Figure 9: Demonstration of multiple mixing plane method. Four blade passages are modelled. Each of them generates inlet boundary condition for a 90-degree section of exhaust hood.

5.1. 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 the total energy fluxes pass through inlet and outlet of the last two stages, which is summed value of all the elements of contour map of energy flux. 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 contours and mass flow rate contours at the outlet of the last two stages are extracted from numerical simulations to generate power contours 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.

6. Test and Result

6.1. 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.

As mentioned above, there are two types of sub-datasets. One has 333 training cases, and the other one has 249 training cases. The designs in two sub-datasets are defined by different parameterization methods. They are incompatible for existing surrogate model because their parameter sets are different. To prove the ability of processing designs from different sources, two sub-datasets are mixed with each other during the training and test. In the test, both of test sub-datasets contains 64 contours.

doi:10.20944/preprints202104.0762.v3

6.2. Test Metrics

Similar with the loss function (Sec. 3.3), 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 contours 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 contours, and $\widehat{y}_i$ is the value of ith pixel of target contours. This error also indicates the error in the predicting averaged value, like averaged pressure, averaged temperature, averaged velocity of a surface.

6.3. Test Result

7. 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 are solutions of partial differential equations, traditionally using finite element analysis and electromagnetic analysis methods.

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 can import the desired two-dimensional distributions of physical properties into the contour encoder, and the designs could be created by the mesh decoder.

8. Conclusion

This study proposes a novel nonparametric surrogate model approach for design and optimization. The new method directly takes the face mesh as input, reducing the uncertainty and loss of geometric information introduced by manual parameterization. This capability gives the new method a greater advantage in building surrogate models for designs with complex geometries.

This approach also exhibits a high degree of flexibility and compatibility. Since the input to the new method is a face mesh, it can adopt the same topology as the database geometry. This means that it is compatible with geometries defined by different parametric methods. This is useful for the ability to further increase the size of the proxy model database using variable data sources.

Compared to existing surrogate model methods, this new method can also predict the two-dimensional distribution of variables (contours) based on face meshes. Outlines can help designers identify physical mechanisms, improve designs, and serve many other purposes. In testing, the average similarity score of 640 contours is 0.9594, which indicates that most of the relevant flow features are captured.

The essence of this new method is to establish a mapping relationship between the surface mesh of the simulation domain and the distribution of physical variables on some 2D profile. Using this approach, engineering systems with complex geometric features such as LPES can be evaluated using databases created in various ways, accelerating the optimization process.

Acknowledgment

This project is supported by by Shanghai Turbine Works Ltd.

References

[1] R. T. Marler, J. S. Arora, Survey of multi-objective optimization methods for engineering, Structural and multidisciplinary optimization 26 (6) (2004) 369–395.

[2] J. Conti, P. Holtberg, J. Diefenderfer, A. LaRose, J. T. Turnure, L. Westfall, International energy outlook 2016 with projections to 2040., USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis., 2016.

[3] E. I. Administration, Hourly electric grid monitor, `https://www.eia.gov/electricity/gridmonitor/dashboard/electric_overview/US48/US48` Accessed July 1, 2021.

[4] T. Tanuma, Y. Sasao, S. Yamamoto, S. Takada, Y. Niizeki, N. Shibukawa, H. Saeki, Numerical investigation of exhaust diffuser performances in low pressure turbine casings, in: ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition, American Society of Mechanical Engineers, 2011, pp. 2347–2357.

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

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

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

[8] 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.

[9] 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.

[10] 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.

[11] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Computer Methods in Applied Mechanics and Engineering 379 (2021) 113741. doi:https://doi.org/10.1016/j.cma.2021.113741. URL https://www.sciencedirect.com/science/article/pii/S0045782521000773

[12] Z. Mao, A. D. Jagtap, G. E. Karniadakis, Physics-informed neural networks for high-speed flows, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112789. doi:https://doi.org/10.1016/j.cma.2019.112789. URL https://www.sciencedirect.com/science/article/pii/S0045782519306814

[13] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, Journal of Computational Physics 394 (2019) 56–81. doi:https://doi.org/10.1016/j.jcp.2019.05.024. URL https://www.sciencedirect.com/science/article/pii/S0021999119303559

[14] Y. Zhu, N. Zabaras, Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification, Journal of Computational Physics 366 (2018) 415–447.

doi:10.20944/preprints202104.0762.v3

doi:https://doi.org/10.1016/j.jcp.2018.04.018.
URL https://www.sciencedirect.com/science/article/pii/S0021999118302341

[15] 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.

[16] 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.

[17] 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.

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

[19] D. P. Kingma, M. Welling, An introduction to variational autoencoders, arXiv preprint arXiv:1906.02691.

[20] 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.

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

[22] 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.

[23] 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.

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

[25] 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.

[26] S. Spekreijse, Elliptic grid generation based on laplace equations and algebraic transformations, Journal of Computational Physics 118 (1) (1995) 38–61. `doi:https://doi.org/10.1006/jcph.1995.1078`. URL `https://www.sciencedirect.com/science/article/pii/S0021999185710789`

[27] 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).