

# PREDICTING THE NEAR-OPTIMAL MESH SPACING FOR A SIMULATION USING MACHINE LEARNING

Callum Lock<sup>1</sup>      Oubay Hassan<sup>2</sup>      Ruben Sevilla<sup>3</sup>      Jason Jones<sup>4</sup>

*Zienkiewicz Institute for Modelling, Data and AI, Faculty of Science and Engineering,  
Swansea University, Swansea, SA1 8EN, Wales, U.K.*

<sup>1</sup>*870431@swansea.ac.uk*

<sup>2</sup>*O.Hassan@swansea.ac.uk*

<sup>3</sup>*R.Sevilla@swansea.ac.uk*

<sup>4</sup>*J.W.Jones@swansea.ac.uk*

## ABSTRACT

This paper presents a novel approach, based on neural networks, to predict the mesh spacing for a simulation by making use of the large amount of simulations that are currently available to industry. The main idea is to compute a spacing function suitable to capture every solution available. The spacing function is then interpolated into a background mesh. A conservative interpolation strategy is proposed to ensure that the mesh spacing function in the coarse mesh does not under-resolve the spacing on the available fine mesh. This step allows to homogenise the data in such a way that a neural network can be employed to predict the spacing at the background mesh. Once the neural network is trained, it can be used to predict the spacing at the nodes of the background mesh, so that near-optimal meshes can be produced for new cases. Numerical examples are used to show the potential of the proposed approach. In addition, a detailed comparison is presented with respect to a recently proposed NN approach in which the location, strength and radius of influence of a set of point sources is predicted.

**Keywords:** Mesh generation, Spacing function, Machine learning, Near-optimal mesh prediction, Computational fluid dynamics

## 1. INTRODUCTION

The generation of unstructured meshes for complex geometric models is still one of the most time consuming parts of the simulation pipeline [1, 2, 3]. This is due to the large amount of human intervention and expertise that is required to produce suitable meshes for simulation.

Mesh generation techniques require the definition of a suitable spacing function that dictates the size of the elements to be generated. The objective is to produce a mesh that concentrate elements only in the regions where they are needed, i.e. regions with complex geometric features to be resolved or regions where complex solution features will be present. The spac-

ing function can be defined using multiple approaches. The more flexible approaches involve the use of point, line or triangular sources [4, 5] and via a structured or unstructured background mesh [6]. Other popular approaches include the refinement based on curvature of the boundary [4] and the definition of the required spacing on selected geometric entities. These approaches can be used independently but they are often combined to achieve a greater control of the spacing. Despite the flexibility of the available tools to produce a suitable spacing function, setting up the required sources or defining an appropriate spacing on a background mesh still requires a significant level of human intervention and expertise.

An alternative is found on mesh adaptive algo-

gorithms [7]. These methods start with a coarse mesh defined by the user and iteratively refine the mesh by identifying the regions where more elements are needed. The main advantage of this approach is the level of automation that can be achieved. However, the initial coarse mesh must be able to capture the solution features to some extent. Otherwise, despite many refinement loops are performed, the solution features will not be captured by the final mesh.

Approaches to utilise neural networks (NNs) to assist the mesh generation process have been also proposed. The earliest attempts to utilise NNs in a mesh generation framework date back to the 1990's and can be found in the field of magnetic device simulations [8, 9, 10]. In the last two years, NNs have been used to assist mesh adaptive algorithms [11, 12, 13] and to predict the spacing in terms of some characteristics of the problem such as the partial differential equation, the boundary conditions and the geometry [14, 15].

In this work we propose a novel approach based on NN to predict the spacing that is required on a background mesh to generate meshes suitable for simulations. The ultimate goal is to utilise the vast amount of data that is available in industry to accelerate the mesh generation stage. To obtain datasets that are suitable for training a NN, a new interpolation approach is presented to transfer the spacing from a fine mesh, where a solution is available, to a coarse background mesh. The interpolation approach is designed to ensure that the spacing in the coarse meshes is able to produce a mesh capable of capturing all the features of the original solution.

The proposed approach is also compared to a recently proposed strategy where a NN is used to predict the position and the spacing at a number of point sources [16]. The comparison is performed based on the time required to train the NNs, including the fine tuning of the hyperparameters, the size of the training dataset required to produce accurate predictions and the accuracy of the predicted spacing function. The comparison is performed by using an example that involves the prediction of near-optimal meshes for a three dimensional wing configuration in the context of inviscid compressible flow simulations. The approach proposed in this work is finally used for a more complex problem involving a full aircraft configuration.

In this work, an optimal mesh is considered to be a mesh with the minimum number of elements that captures all the features of the solution. The proposed technique is aimed at producing near-optimal meshes in the spacing function lies within 5% of the target spacing.

The remainder of the paper is organised as follows. In

Section 2 a brief summary of the two strategies considered to control spacing is presented, namely the use of sources and a background mesh. Section 3 describes the strategy used to compute the required spacing to produce a mesh that captures all the features of a given solution. In Section 4 the strategy to compute a set of global sources that is capable of producing the required spacing to capture a number of given solutions is described. Similarly, Section 5 describes the approach to compute the spacing on a background mesh that is capable of representing a number of given solutions. The use of a NN to predict the source characteristics or the spacing of a background mesh is presented in Section 6. Two examples are considered in Section 7. The first example is used to compare the approaches based on sources and the proposed approach based on a background mesh. The second example shows the potential of the proposed strategy on a large scale problem involving a full aircraft. Finally, Section 8 summarises the conclusions of the work that has been presented.

## 2. MESH SPACING AND CONTROL

This section introduces the fundamental concepts on how mesh spacing is defined within a mesh that are utilised when presenting the two proposed strategies to predict near-optimal meshes.

Within the aerospace industry, there is a preference for using unstructured meshes for CFD simulations, owing to their ability to efficiently discretise complex geometric domains. This is due to the possibility to locally refine targetted regions without inducing a refinement in regions that are not of interest. Refinement techniques can be classified into automatic techniques, based for instance on mesh adaptivity, or controlled manually based on the expertise of the user. Automatic adaptive algorithms can be used to localise the refinement only in the regions where high gradients of the solution are present. However, it is clear that if the initial mesh is not able to capture some solution features, they will not be captured even if a large number of refinement loops are undertaken.

There are different methods available for the user to control the spacing function, which can be used independently or in combination with one another. These techniques include the use point, line or triangular sources [4, 5], the use of a background structured or unstructured mesh [6, 4], the specification of the spacing at geometric entities or the refinement based on the curvature of the boundary [4]. The use of sources and a background unstructured mesh are considered here as the strategies that provide greater flexibility.

## 2.1 Mesh spacing controlled by sources

Sources provide the ability to control the spacing desired at a localised region of the domain. A point source consists of a given location  $\mathbf{x}$  a desired spacing  $\delta_0$  and a radius of influence  $r$ . The spacing function induced by the source is constant, and equal to  $\delta_0$ , within the sphere of centre  $\mathbf{x}$  and radius  $r$ . To ensure a smooth transition of the spacing outside the sphere of influence, an exponential increase of the spacing is defined by specifying a second radius,  $R$ , where the spacing doubles. The spacing at a distance  $d$ , from  $\mathbf{x}$ , is defined as

$$\delta(d) = \begin{cases} \delta_0 & \text{if } d < r \\ \delta_0 e^{\ln(2) \frac{d-r}{R-r}} & \text{otherwise} \end{cases}. \quad (1)$$

A line source is a natural extension of the concept of point sources. Line sources are made of two point sources. To determine the spacing induced by a line source at a given point,  $\mathbf{p}$ , the closest point to  $\mathbf{p}$  on the line is found, namely  $\hat{\mathbf{p}}$ . A linear interpolation of the radii and spacing of the two points forming the line source is used to determine the radius and spacing that must be associated to the projected point  $\hat{\mathbf{p}}$ . The spacing induced by the line source is computed by assuming that a point source is present at  $\hat{\mathbf{p}}$  with the interpolated radius and spacing. Similarly, it is possible to extend this concept to other geometric entities such as triangular sources.

It is worth noting that when multiple sources are used to control the spacing function, the minimum of all the spacings induced by all the sources is used to specify the required spacing at a given location.

## 2.2 Mesh spacing controlled by a background mesh

Alternatively, the spacing can be controlled by using a background structured or unstructured mesh [6, 4]. In this scenario a coarse mesh that covers the whole computational domain is generated, and the spacing is specified at each node of the background mesh. To determine the spacing at any point of the domain, the element of the background mesh that contains the point is first identified. Then, a linear interpolation of the nodal values of the spacing defined at the elements of the background mesh is employed.

## 3. TARGET SPACING

It is assumed that a dataset of accurate solutions is available. The solutions might have been computed with different numerical schemes, and very often, on overrefined meshes. For this reason, this work proposes a learning procedure that is based on the solution, rather than on the meshes that were used to

compute the solutions. However, the technique proposed can be modified to learn from existing meshes in cases where the meshes are considered to be optimal, i.e. obtained after an adaptive process or manually created by an expert, by obtaining the spacing distribution of said mesh.

The first stage involves the computation of the spacing function that would provide a mesh capable of reproducing a given solution. This is done by borrowing concepts of error analysis and relating the desired spacing to the second-order derivatives of the solution, namely

$$\delta_{\beta}^2 \left( \sum_{i,j=1}^N H_{ij} \beta_i \beta_j \right) = K, \quad (2)$$

where  $\beta$  is an arbitrary unit vector,  $\delta_{\beta}$  is the spacing along the  $\beta$  direction,  $\mathbf{H}$  is the Hessian matrix of a key variable  $\sigma$  and  $K$  is a user-defined constant.

Here, a recovery process [17, 18] is employed to numerically evaluate the second derivatives of the selected key variable. Next, by evaluating in the direction of each eigenvector of  $\mathbf{H}$ , the *optimal* value of the spacing at a node is defined as

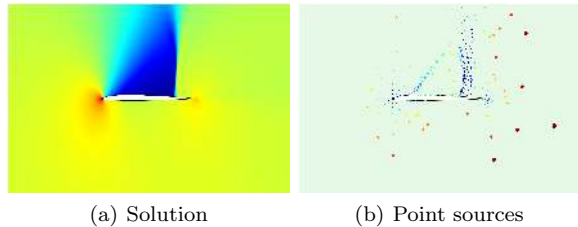
$$\delta_i = \min_{i=1,\dots,n} \left\{ \sqrt{\frac{K}{\lambda_i}} \right\}, \quad (3)$$

where  $\{\lambda_i\}_{i=1,\dots,n}$  denote the eigenvalues of  $\mathbf{H}$ .

The discrete spacing is uniquely defined after the user specifies the scaling factor,  $K$ . In regions where the solution is smooth, the scaling reflects the value of the mean square error that is to be considered acceptable.

In practice, to account for the possibility of vanishing eigenvalues of  $\mathbf{H}$ , the spacing of Equation (3) is bounded by a maximum allowable value. Similarly, to avoid an excessive refinement near elements with very steep gradients (e.g., near shocks), a minimum value of the spacing is also defined by the user.

At this stage a discrete representation of the spacing function is obtained. However, for each solution available the number of mesh nodes is generally different so two strategies are considered to homogenise the data in such a way that is suitable for training a NN. The first strategy, proposed in [16] consists of building a global set of sources that is capable of describing the spacing function of each case. The second approach, proposed here for the first time consists of building a spacing function by using a background mesh that is also suitable to describe the spacing function of each case. The two strategies are described in the next two sections.



**Figure 1:** Transonic flow CFD solution and point sources to create a spacing function capable of capturing the solution for a NACA1206.

## 4. SPACING DESCRIPTION USING SOURCES

The main idea is to construct a set of sources that induce a spacing function that closely represents the discrete spacing obtained using the strategy described in the previous section. The full details and the algorithmic implementation are described in detail in [16].

### 4.1 Generating point sources for one solution

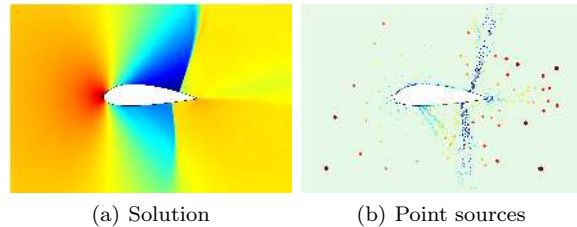
The process starts by grouping points based on the associated spacing. Point sources are created at the centre of a group of points and with a radius that covers all the points in the group.

The strategy developed guarantees that the spacing required at every node of the given mesh is represented by at least one point source. To simplify the implementation, this work assumes that the second radius of influence of a source is always double the first radius, namely  $R = 2r$ . It is also imposed that two values of the spacings are *close enough*, if they differ by at most 5% of the spacing at the node of interest. The process for creating a point source ends when the spacing at a surrounding layer is larger than the spacing of the point source at a distance equal to  $R$ .

Figure 1 shows the result of creating sources to represent the spacing required to capture a given solution. The solution corresponds to an actual two dimensional inviscid transonic flow simulation. It can be observed how the sources with smaller spacing (blue colour) are concentrated near the regions with steep gradients.

### 4.2 Generating global sources for a set of solutions

When the process described in the previous section is applied to a set of different solutions, the number of sources obtained is, in general, very different. As the objective is to utilise this data to train a NN, a procedure to obtain the same number of sources for a



**Figure 2:** Transonic flow CFD solution and point sources to create a spacing function capable of capturing the solution for a NACA4324.

set of solutions is devised.

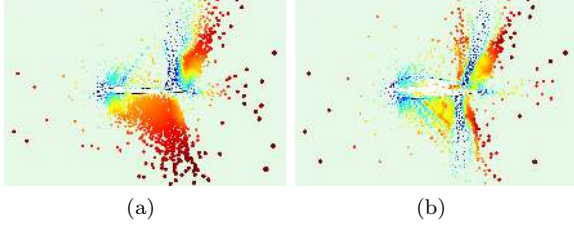
The process starts by initialising the set of global sources to be the set of sources of the first case and creating a mapping that relates global sources to the local sources of each case. The rest of the cases are then considered sequentially. To ensure an efficient implementation, the sources of all cases are inserted in an alternating digital tree (ADT). The process then considers each one of the remaining cases sequentially. The ADT is employed to identify global sources that are in close proximity to the unprocessed local sources. When a global source is not found in close proximity to a local source, a new source is added to the global list and the mapping between local and global sources is updated. In contrast, when a global source is in close proximity to a local source, no new sources are added and the mapping is updated. After the set of global sources is produced, it is customised to accurately represent the spacing function associated to each solution.

Figure 2 shows the result of creating sources to represent the spacing required to capture a solution different to the one shown in Figure 1, i.e. a different geometry and different flow conditions. It can be observed that the number of sources will significantly differ depending on the flow conditions and geometry.

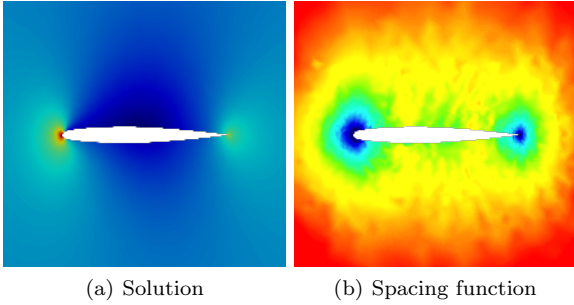
By using the process briefly described in this section a set of global sources is created for each case, as depicted in Figure 3. Both set of global sources are different but they have the same number of sources, which is necessary to ensure that this data can be used to train a NN.

## 5. SPACING DESCRIPTION USING A BACKGROUND MESH

A novel approach to build a spacing function that ensures uniformity of the data and therefore its possible use for training a NN is presented here. The process consists of creating a coarse background mesh and to devise a strategy to compute the spacing at each node of the background mesh that induces the



**Figure 3:** Global sources for (a) the case of Figure 1 and for (b) the case of Figure 2.



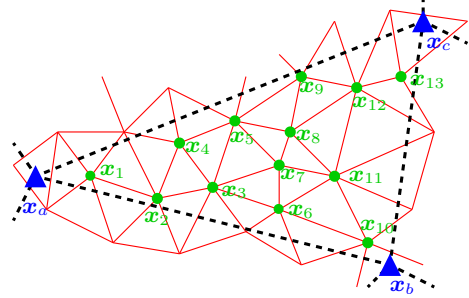
**Figure 4:** A solution and its corresponding calculated spacing function that describes the optimal spacing suitable for capturing the solution.

required spacing to capture a given solution.

First, the spacing is computed on the mesh where the solution is provided. Figure 4 shows a solution and the spacing function that will provide the required mesh to capture the solution.

The implementation of this strategy introduces several advantages when compared to the existing strategy of using sources. First, the computation of the spacing at the nodes of the background mesh is simpler than the computation of the sources as it only requires interpolating the spacing from a fine mesh to a coarse mesh. This is described in detail in this section. Second, uniformity of the data is guaranteed if the topology of the background mesh is unchanged. For the examples considered in this work, with design parameters not affecting the geometry of the domain, a fixed background mesh can be used for all cases. For more complex scenarios, a mesh morphing algorithm would be required to ensure that the same background mesh can be used for all cases. This is out of the scope of the current work.

The fixed background mesh is produced using a combination of curvature control and minimum spacing defined at each individual surface of the geometry; an octree is then used to propagate the spacing into the domain.



**Figure 5:** Detail of two triangular meshes, the fine mesh where the solution is computed – denoted by continuous red edges, and a coarser background mesh denoted by discontinuous black edges. The green circles denote the nodes of the fine mesh contained in one element of the background mesh. The blue triangles denote the nodes of the background mesh where the interpolated element spacing is to be computed.

## 5.1 Interpolating the spacing on a background mesh

As mentioned above, the proposed strategy to use a background mesh requires interpolating the discrete spacing from a mesh where a solution is available to a coarse background mesh. Interpolating a field from one mesh to another is a relatively easy task but special care must be taken when the quantity to be interpolated is the spacing. If a naïve interpolation is employed, many features of the solution can be unresolved by the spacing function embedded in the background mesh.

To illustrate the proposed strategy, let us consider the scenario of Figure 5. The extract of the mesh with continuous red edges corresponds to the mesh where the solution is available and where the spacing required at the nodes has been computed, as described in Section 3. The extract of the mesh with discontinuous black edges corresponds to the background mesh. The objective is to obtain the spacing at the nodes of the background mesh,  $x_a$ ,  $x_b$  and  $x_c$  in Figure 5 so that the spacing at the nodes of the fine mesh can be accurately reproduced.

A naïve interpolation approach would consider the element of the fine mesh containing each node of the coarse mesh and perform a linear interpolation of the nodal values. However, this will make very limited use of the rich information available in the fine mesh. If values of the spacing are interpolated in this way, it is possible to obtain very large values of the spacing at the nodes of the background mesh even if very small values are present in the vicinity of the nodes of the background mesh. Referring to the example of Figure 5, if for instance the spacing is very small at nodes  $x_6$ ,  $x_7$ ,  $x_8$  and  $x_9$ , but it is very large at the remain-



ing nodes, a naïve interpolation will compute a large value of the spacing at the nodes  $\mathbf{x}_a$ ,  $\mathbf{x}_b$  and  $\mathbf{x}_c$  of the background mesh. This will induce a spacing function not suitable to capture the initial solution.

To avoid this problem, a different strategy is proposed to interpolate the spacing. For each element of the background mesh, the list of nodes of the fine mesh that are contained in the background element is identified. In the example of Figure 5 all the numbered nodes, from  $\mathbf{x}_1$  to  $\mathbf{x}_{13}$  are identified. A very conservative approach is then adopted in this work, which is to define the spacing at the element nodes of the background mesh as the minimum of the spacing of all the nodes of the fine mesh contained in the element. This strategy will ensure that the resulting spacing is certainly able to capture the required solution. Other strategies that can be explored include the use of the arithmetic mean, the harmonic mean, or a weighted arithmetic mean.

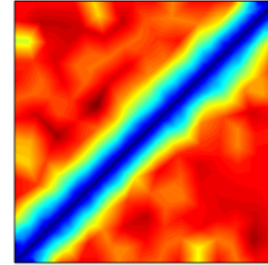
The process is finalised by assigning to each node of the background mesh the minimum of the spacings computed from each element sharing this node.

An example is shown in Figure 6 to illustrate the process. The spacing function obtained on a reference mesh is transferred to a background mesh using a naïve interpolation approach and the proposed approach. It can be clearly observed that the naïve approach does not produce an accurate representation of the original spacing function. When used for mesh generation, this background spacing will lead to a mesh that is not capable of representing the features of the target solution. In contrast, with the proposed interpolation, a conservative approach is favoured and the resulting spacing will lead to a finer mesh, ensuring that all the features of the target solution are captured when a mesh is generated with this spacing.

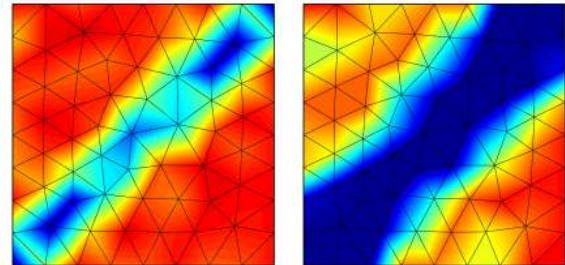
Figure 7 shows the interpolated spacing on a background mesh for the example of Figure 7. The example clearly illustrates the ability of the proposed interpolation strategy to capture the required spacing on a coarse background mesh. It is worth noting that the proposed approach is designed to produce a spacing capable of capturing all the required solution features. However, when the background mesh is excessively coarse, it can produce a spacing function that leads to an over-refined mesh.

## 6. USING A NEURAL NETWORK TO PREDICT THE SPACING

The two strategies presented above are designed to preprocess a dataset of available solutions and produce a dataset suitable for training a neural network. The inputs of the neural network are design parameters (e.g., boundary conditions, geometry). The examples

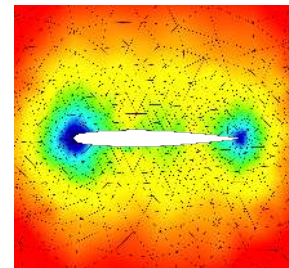


(a) Original spacing



(b) Naïve interpolation (c) Proposed interpolation

**Figure 6:** Illustrative example of two possible interpolations of the spacing onto a background mesh.



**Figure 7:** Spacing function on a background mesh after interpolating the spacing of Figure 4(b).

considered in this work involve inviscid compressible flows in three dimensions and the design parameters are the flow conditions, namely the free-stream Mach number and the angle of attack.

For the strategy based on sources, the output consists on the position (three coordinates), the spacing and the radius of the global set of sources. For the second approach, based on a background mesh, the output is simply the spacing at the nodes of the background mesh. It is worth noting that the use of a background mesh implies a reduction of the number of outputs by a factor of five, when compared to the strategy based on sources.

In general, the values of the spacing, in both approaches, and the radius, in the first approach, vary by more than two orders of magnitude. To facilitate the

training of the NN, these outputs are scaled logarithmically. The scaling not only prevents a bias towards larger values but also prevents the prediction of unrealistic negative values.

The type of NN employed in this work is a standard multi-layer perceptron and extensively described in the literature [19, 20]. In terms of the implementation, TensorFlow 2.7.0 [21] is employed to construct the NNs. To minimise the influence of the random initialisation of the weights, each training is performed five times by performing a variation of the initial guess used in the optimisation. The maximum number of iterations allowed for the optimisation is 500, and the process is stopped either when this number of iterations is reached or when the objective cost function does not decrease during 50 consecutive iterations.

Through preliminary numerical experimentation on the influence of the activation function on the accuracy of the NN, the sigmoid activation function was chosen as it tended to produce more accurate results compared to other classical activation functions. Therefore, for each NN produced, the sigmoid function was employed for all the hidden layers, with a linear function being used on the output layer. Respectively, these activation functions are given by

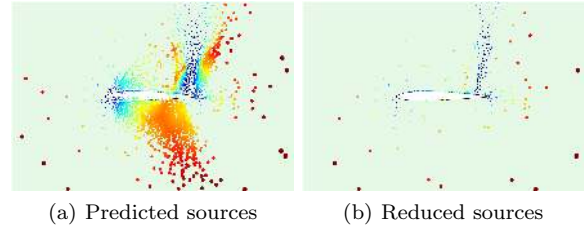
$$S(x) = \frac{1}{1 + e^{-x}} \quad \text{and} \quad L(x) = x. \quad (4)$$

To train the NNs, the cost function used is the mean square error (MSE), with the optimisation function used to minimise the cost being the ADAM optimiser [22], with a learning rate of 0.001.

As usual in this context, the hyperparameters of the NN are tuned to ensure that the best architecture is employed. In [16] the authors demonstrate that even performing a fine tuning of the hyperparameters and repeating the training five times, the resulting approach is more efficient than the usual practice in industry of generating an overrefined mesh to perform the simulations for varying flow conditions using a fixed grid.

The design of the NNs considered in this work requires selecting an appropriate number of layers, number of neurons per layer and activation functions. For each numerical example in Section 7, the number of layers  $N_l$  and the number of neurons in each layer  $N_n$  is varied in the pursuit of finding the optimal hyperparameter configuration. The hyperparameter variation is defined by a grid using the ranges  $N_l = [1, 2, \dots, 5, 6]$  and  $N_n = [25, 50, \dots, 225, 250]$ .

The accuracy of the predictions is measured using the statistical  $R^2$  measure [23]. To better analyse the results, when the approach using sources is considered, the  $R^2$  measure is reported independently for the five



**Figure 8:** Predicted global sources for an unseen case and the resulting sources after removing redundant sources.

source characteristics (i.e., the three coordinates of the source, the spacing and the radius).

### 6.1 Spacing prediction using sources

After the NN is trained, it is used to predict the characteristics of the global sources for cases not seen during the training stage. It is possible to directly use the predicted global sources to define the mesh spacing function that is required to generate a near-optimal mesh. However, due to the use of a global set of sources, it is expected that the predicted sources for a new case contain redundant information. For this reason an extra step is required with this technique to minimise the number of queries that the mesh generation requires to define the spacing at a given point. The process, described in detail in [16], involves removing sources with an associated spacing function that can be described by other sources. In addition, an attempt is made to reduce the number of sources by merging point sources into line sources when possible.

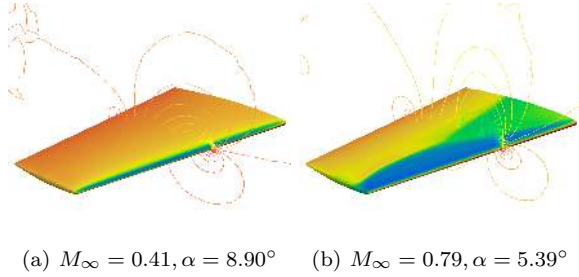
Figure 8 shows the result of the process used to reduce sources for a predicted global set of sources.

### 6.2 Spacing prediction using a background mesh

In this case, once the NN is trained it can be used to predict the spacing at the nodes of the background mesh. With this approach, there is no need to perform any further processing of the predicted data and it can be directly used by a mesh generator to obtain the near-optimal mesh for an unseen case.

## 7. NUMERICAL EXAMPLES

This section presents a numerical example to demonstrate the potential of both approaches and to compare their performance. The example involves the prediction of near-optimal meshes for three dimensional inviscid compressible flow simulations over a wing for varying flow conditions. A second numerical example is presented to show the ability of the approach



**Figure 9:** Pressure coefficient,  $C_p$ , for the ONERA M6 wing and for two flow conditions.

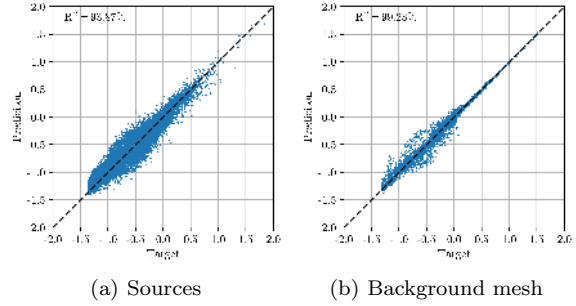
best performing for the first example in a more realistic scenario involving the inviscid compressible flow around a full aircraft configuration. In both examples the variation of the flow conditions considered induce a significant variation of the solution and include subsonic and transonic flows. All the CFD simulations used in this work were conducted using the in-house flow solver FLITE [24].

### 7.1 Near-optimal mesh predictions on the ONERA M6 wing

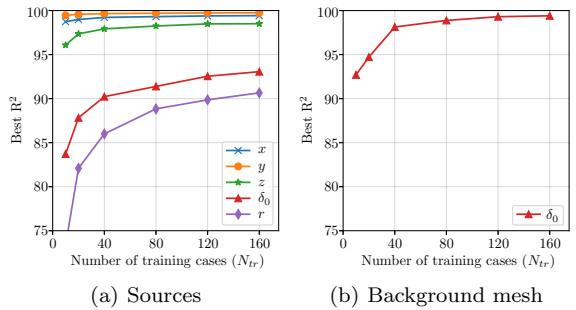
The ONERA M6 wing [25] is considered for this example and flow conditions are described by the free-stream Mach number,  $M_\infty$ , and the angle of attack,  $\alpha$ . The range used for the two design parameters,  $M_\infty \in [0.3, 0.9]$  and  $\alpha \in [0^\circ, 12^\circ]$ , leads to subsonic and transonic flows. Therefore, the mesh requirements for different cases are substantially different, posing a challenge in the prediction of the near-optimal mesh for a given set of parameters.

The variation of the solution that is induced by the variation of the parameters is illustrated in Figure 9, showing the pressure coefficient,  $C_p$ , for two flow conditions. For the subsonic case, with  $M_\infty = 0.41$  and  $\alpha = 8.90^\circ$ , the solution requires refinement only near the leading and trailing edges. In contrast, for the transonic case, with  $M_\infty = 0.79$  and  $\alpha = 5.39^\circ$ , the mesh should be refined to also capture the  $\lambda$ -shock on the top surface. The simulations were conducted using tetrahedral meshes with approximately 1.3M elements and 230K nodes.

For the purpose of this study, training and testing data sets were generated by employing a Halton sampling [26] in the parametric space. The training set comprises  $N_{tr} = 160$  cases, whereas the test set is made of  $N_{tst} = 90$  cases. To ensure that the conclusions are not biased by an incorrect use of the NN for extrapolation, the range of values used to generate the test set is slightly smaller than the range used to generate the training data.



**Figure 10:** The regression plots for the spacing,  $\delta_0$ , for the approach using sources and the approach using a background mesh.



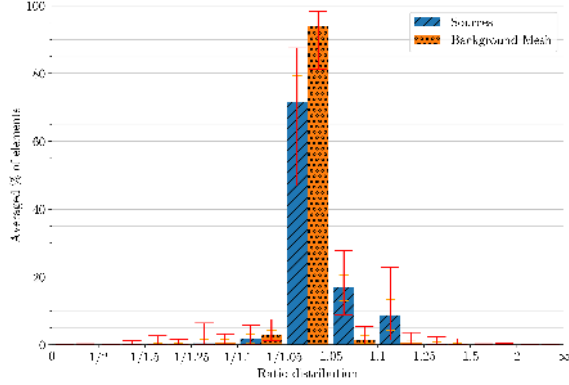
**Figure 11:** ONERA M6: Minimum  $R^2$  for the predicted outputs as a function of the number of training cases for the two methods.

The approach using sources, required between 2,142 and 5,593 sources to represent the spacing of each training case. When combined, the resulting number of global sources is 19,345. This means that the number of outputs of the NN to be trained is almost 100K. For the second approach a background mesh with 14,179 nodes is employed, meaning that the NN to be trained has almost seven times less outputs when compared to the approach that uses sources.

After tuning the NN that best predicts the spacing, both approaches can be compared. For one of the 90 unseen cases Figure 10 shows the regression plot for the spacing for both approaches. The results indicate a better performance of the approach using a background mesh for this particular unseen case.

To better compare the accuracy, the minimum  $R^2$  for each of the 90 unseen test cases is taken and compared in Figure 11, for an increasing number of training cases. The results show that the strategy that uses sources leads to a very accurate prediction of the location of the sources. However, predicting the spacing and the radius of influence seems much more challenging. To achieve an  $R^2$  of 90 in all the outputs,



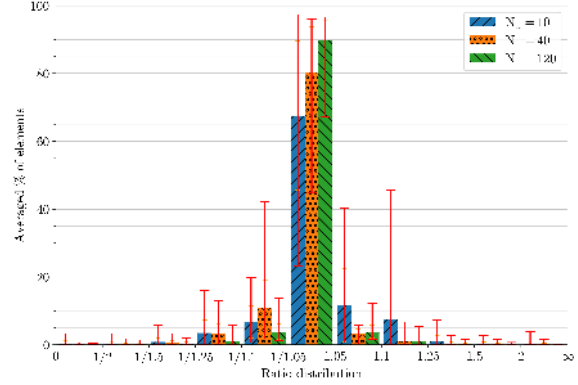


**Figure 12:** ONERA M6: Histogram of the ratio between the predicted and target spacing for the two strategies.

the whole training data set, with 160 cases, must be considered. In contrast, for the strategy that uses a background mesh 10 training cases are enough to provide an  $R^2$  above 90. By comparing the results, it is clear that the approach that uses a background mesh is significantly more efficient as with 10 training cases the results are as accurate as with the approach that uses sources employing 160 training cases. It is also worth remarking that the approach that uses sources requires the training of multiple NN, whereas only one NN is to be trained by the approach proposed here. In this example the tuning and training of the NN for the proposed approach is almost four times faster than the approach using sources.

To further analyse the performance of the two approaches, the predicted spacing function through the domain is compared against the target spacing function for the two methods. At the centroid of each element of a target mesh, and for all test cases, the spacing induced by the two strategies is compared to the target spacing. Figure 12 shows a histogram of the ratio between the predicted and target spacing for both methods. The results correspond to both approaches using all the available training data. Red bars are used to depict the minimum and maximum values for each bin in the histogram and the standard deviation from the mean is represented by the orange bars. A value of the ratio of spacings between  $1/1.05$  and  $1.05$  is considered accurate enough to generate a mesh that is capable of resolving all the required flow features. Values higher than  $1.05$  where the predicted spacing is larger than the target spacing and, analogously, values below  $1/1.05$  indicate regions where the NN prediction will induce more refinement than required.

The results in Figure 12 clearly illustrate the superiority of the strategy proposed in this work, by using a background mesh. The strategy based on sources



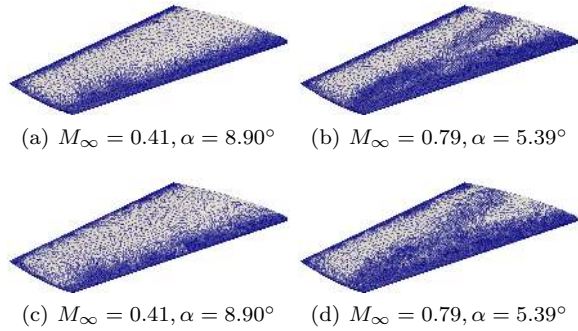
**Figure 13:** ONERA M6: Histogram of the ratio between the predicted and target spacing for the strategy using a background mesh for an increasing number of training cases.

provides approximately 70% of the elements with an appropriate spacing whereas the approach based on a background grid accurately predicts the spacing for almost 95% of the elements. In addition, the worst performing case for the approach using sources is less accurate than the worst case for the approach that uses a background mesh. Finally, it is worth mentioning that when the background mesh approach is less accurate, it tends to produce a smaller spacing, which is preferred to a larger spacing, as this will ensure that all solution features are resolved with the predicted near-optimal mesh. This tendency to over-refine can be explained by the conservative interpolation scheme that has been introduced in Section 5.1.

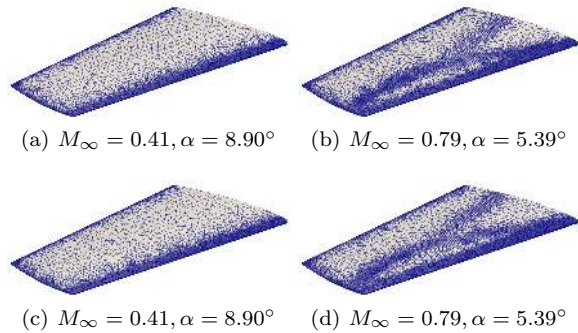
Given the high accuracy observed in Figure 11 for the approach that uses a background mesh with very few training case, Figure 13 shows the histogram of the ratio of between predicted and target spacing for an increasing number of training cases. The results corroborate the conclusions obtained from the  $R^2$  measure in Figure 11 and show that with a significantly smaller number of training cases, the approach using a background mesh not only produces an  $R^2$  comparable to the approach with sources with all training data, but also the predicted spacing is as accurate.

To illustrate the potential of the strategies being compared, the trained NNs are used to predict the spacing function for unseen cases and near-optimal meshes are generated and compared to the target meshes. It is worth remarking that the approach that uses sources undergoes the extra processing step to reduce and merge sources as mentioned in Section 6.1.

Figure 14 shows two target meshes and the near-optimal mesh prediction obtained with the strategy based on sources for two test cases not seen during the training of the NN. The comparison between tar-



**Figure 14:** Target (top row) and predicted (bottom row) meshes using the strategy based on sources.

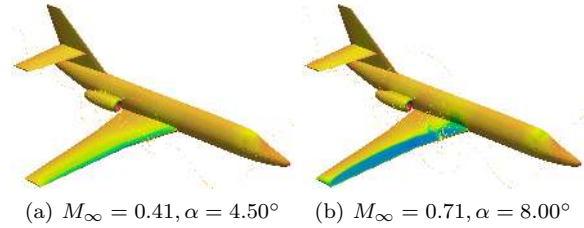


**Figure 15:** Target (top row) and predicted (bottom row) meshes using the strategy based on a background mesh.

get and predicted meshes using the strategy based on a background mesh is shown in Figure 15. It is worth noting that the target meshes for the two strategies considered are slightly different due to the different definition of the target spacing function.

The results visually show superior accuracy of the proposed approach, based on a background mesh. Not only the meshes obtained with the predicted spacing functions resemble the target more than the meshes predicted with sources, but also the spacing gradation is visually smoother with the approach based on a background mesh.

Further numerical experiments, not reported here for brevity demonstrate that the CFD calculations on the near-optimal predicted meshes result in accurate CFD simulations. More precisely, the aerodynamic quantities of interest (e.g., lift and drag) are obtained with the required accuracy for the aerospace industry.



**Figure 16:** Falcon aircraft: Pressure coefficient,  $C_p$ , for two different flow conditions.

## 7.2 Near-optimal mesh predictions on the Falcon aircraft

After demonstrating the superiority of the approach based on a background mesh, this section considers an example with a more complex and realistic geometry to show the potential of this approach.

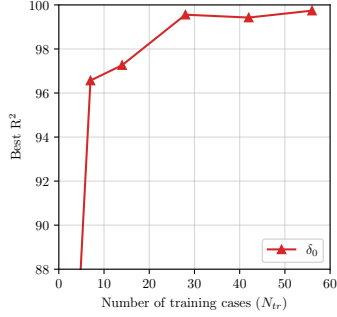
Halton sequencing of the two input parameters is used to generate a training dataset consisting of  $N_{tr} = 56$  training cases and  $N_{tst} = 14$  testing cases. The range used for the parameters is  $M_\infty \in [0.35, 0.8]$  and  $\alpha \in [-4^\circ, 10^\circ]$ , leading, again, to subsonic and transonic flow regimes.

For each training and test case, the CFD solution is obtained using FLITE [24] on an unstructured tetrahedral mesh consisting of 6M elements and 1M nodes. The distribution of the pressure coefficient for two test cases is shown in Figure 16. The Figure shows the different flow features that are induced by a change in the design parameters.

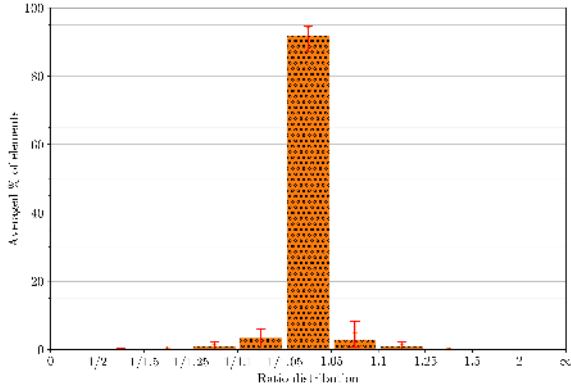
To represent the spacing function, the spacing is first determined at each node of the mesh where the solution was computed. A coarse unstructured background mesh is then generated, containing approximately 30K tetrahedral elements, and the spacing is interpolated to the background mesh using the technique described in Section 5.1.

A NN is then trained and the hyperparameters are tuned, following the procedure described in the previous example. After the training is performed, the spacing is predicted for the 14 unseen test cases and the accuracy of the predictions is evaluated using the  $R^2$  measure. Figure 17 shows the minimum  $R^2$ , as a function of the number of training cases. The results show that, even for this more complex example, the behaviour is almost identical to the one observed for the previous geometry. With less than 10 training cases the predicted spacing achieves an excellent accuracy, with the value of  $R^2$  above 96%. If the total set of available training cases is considered, the value of  $R^2$  reaches almost 100.

To further assess the accuracy of the predictions, the



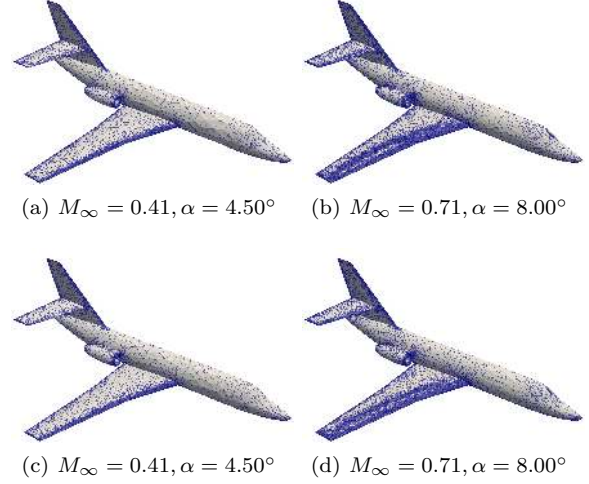
**Figure 17:** Minimum  $R^2$  for the characteristics as a function of the number of training cases.



**Figure 18:** Falcon aircraft: Histogram of the ratio between the predicted and target spacing.

ratio between predicted and target spacing is evaluated to quantify the performance of NN in producing new meshes for unseen flight conditions. Figure 18 shows the histogram of the ratio between predicted and target spacing at the nodes of the background mesh. The minimum and maximum values for each bin in the histogram are depicted with red error bars, whereas the orange bar represents the standard deviation from the mean. A value of the ratio between  $1/1.05$  and  $1.05$  is considered sufficiently accurate to produce a mesh able to capture the targeted flow features. The histogram confirms the accuracy of the predictions, with the middle bin containing more than 90% of the elements.

The trained NNs are next used to predict the spacing for the background mesh, from which its subsequent near-optimal mesh is generated and compared with the corresponding target meshes. Figure 19 displays the target and ML-produced meshes for the two unseen examples outline in Figure 16. The results clearly show the ability of the proposed technique, based on a background mesh, to automatically produce meshes that are locally refined near the relevant regions. For



**Figure 19:** Falcon aircraft: Target (top row) and predicted (bottom row) meshes for two flow conditions.

the subsonic case, the NN has appropriately refined the leading and trailing edges of the main wing, the vertical and horizontal stabiliser, as well as the entry and exit of the jet engine. Similarly, for the transonic case, those features are also appropriately captured, but in addition, the NN has successfully predicted the presence and location of a shock along the main wing and consequently appropriately refined this region.

## 8. CONCLUDING REMARKS

A novel technique to predict the required spacing for a simulation has been presented. The approach is based on the use of background mesh and a NN to predict the required spacing at each node of the background mesh. Using available data from previous simulations, the required spacing to capture a given solution is computed on the available mesh. Then, a method to interpolate the spacing onto the background mesh is devised. The approach is conservative and avoids the problems that a naïve interpolation will induce. Once the available data is processed, a NN is trained where the inputs are design parameters (i.e., flow conditions in the example considered here) and the output is the required spacing at the background mesh. When the spacing is available, a standard mesh generator can be used to obtain the near-optimal mesh suitable for a new simulation.

The strategy has been compared to a recently proposed approach in which a NN is used to predict the position, strength and radius of influence of a set of sources. The results show that the proposed approach is much more efficient. First, it requires significantly less training data to provide the same accuracy. Sec-

ond, the NNs to be trained are significantly smaller due to the need to only predict the spacing at the nodes of the background mesh. In addition, it does not require a complex processing of the available data to create a set of global sources.

The proposed approach has been applied to two examples, relevant to the aerospace industry. Flow conditions were considered as the design parameters and three dimensional examples showed the potential of the proposed approach in dealing with large scale problems.

Future work will include the extension of this approach to deal with geometric parameters and the ability to predict anisotropy in the near-optimal meshes.

## References

- [1] Dawes W., Dhanasekaran P., Demargne A., Kellar W., Savill A. “Reducing bottlenecks in the CAD-to-mesh-to-solution cycle time to allow CFD to participate in design.” *Journal of Turbomachinery*, vol. 123, no. 3, 552–557, 2001
- [2] Slotnick J.P., Khodadoust A., Alonso J., Darmofal D., Gropp W., Lurie E., Mavriplis D.J. “CFD vision 2030 study: a path to revolutionary computational aerosciences.” Tech. rep., 2014
- [3] Karman S.L., Wyman N., Steinbrenner J.P. “Mesh generation challenges: A commercial software perspective.” *23rd AIAA Computational Fluid Dynamics Conference*, p. 3790. 2017
- [4] Thompson J.F., Soni B.K., Weatherill N.P. *Handbook of grid generation*. CRC press, 1998
- [5] Löhner R. *Applied computational fluid dynamics techniques: an introduction based on finite element methods*. John Wiley & Sons, 2008
- [6] Peraire J., Peiro J., Morgan K. “Adaptive remeshing for three-dimensional compressible flow computations.” *Journal of Computational Physics*, vol. 103, no. 2, 269–285, 1992
- [7] George P.L., Borouchaki H., Alauzet F., Laug P., Loseille A., Marcum D., Maréchal L. “Mesh Generation and Mesh Adaptivity: Theory and Techniques.” E. Stein, R. de Borst, T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics Second Edition*, vol. Part 1 Fundamentals, chap. 7. John Wiley & Sons, Ltd., Chichester, 2017
- [8] Dyck D., Lowther D., McFee S. “Determining an approximate finite element mesh density using neural network techniques.” *IEEE transactions on magnetics*, vol. 28, no. 2, 1767–1770, 1992
- [9] Chedid R., Najjar N. “Automatic finite-element mesh generation using artificial neural networks-Part I: Prediction of mesh density.” *IEEE Transactions on Magnetics*, vol. 32, no. 5, 5173–5178, 1996
- [10] Alfonzetti S., Coco S., Cavalieri S., Malgeri M. “Automatic mesh generation by the let-it-grow neural network.” *IEEE transactions on magnetics*, vol. 32, no. 3, 1349–1352, 1996
- [11] Chen G., Fidkowski K. “Output-based error estimation and mesh adaptation using convolutional neural networks: Application to a scalar advection-diffusion problem.” *AIAA Scitech 2020 Forum*, p. 1143. 2020
- [12] Bohn J., Feischl M. “Recurrent neural networks as optimal mesh refinement strategies.” *Computers & Mathematics with Applications*, vol. 97, 61–76, 2021
- [13] Yang J., Dzanic T., Petersen B., Kudo J., Mittal K., Tomov V., Camier J.S., Zhao T., Zha H., Kolev T., et al. “Reinforcement learning for adaptive mesh refinement.” *International Conference on Learning Representations. 2022*
- [14] Zhang Z., Wang Y., Jimack P.K., Wang H. “MeshingNet: A new mesh generation method based on deep learning.” *International Conference on Computational Science*, pp. 186–198. Springer, 2020
- [15] Zhang Z., Jimack P.K., Wang H. “MeshingNet3D: Efficient generation of adapted tetrahedral meshes for computational mechanics.” *Advances in Engineering Software*, vol. 157, 103021, 2021
- [16] Lock C., Hassan O., Sevilla R., Jones J. “Meshing using NNs for improving the efficiency of computer modelling.” *Engineering with computers*, vol. Under review, 2022
- [17] Zienkiewicz O.C., Zhu J.Z. “The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique.” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, 1331–1364, 1992
- [18] Zienkiewicz O.C., Zhu J.Z. “The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity.” *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, 1365–1382, 1992
- [19] Hagan M.T., Demuth H.B., Beale M. *Neural network design*. PWS Publishing Co., 1997

- [20] Balla K., Sevilla R., Hassan O., Morgan K. “An application of neural networks to the prediction of aerodynamic coefficients of aerofoils and wings.” *Applied Mathematical Modelling*, vol. 96, 456–479, 2021
- [21] Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M., et al. “TensorFlow: a system for Large-Scale machine learning.” *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283. 2016
- [22] Kingma D.P., Ba J. “ADAM: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980*, 2014
- [23] Glantz S.A., Slinker B.K. *Primer of applied regression & analysis of variance, ed*, vol. 654. McGraw-Hill, Inc., New York, 2001
- [24] Sørensen K., Hassan O., Morgan K., Weatherill N. “A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow.” *Computational mechanics*, vol. 31, no. 1-2, 101–114, 2003
- [25] Schmitt V. “Pressure distributions on the ONERA M6-wing at transonic Mach numbers, experimental data base for computer program assessment.” *AGARD AR-138*, 1979
- [26] Halton J.H. “Algorithm 247: Radical-inverse quasi-random point sequence.” *Communications of the ACM*, vol. 7, no. 12, 701–702, 1964