DOI: XXXXX

**ARTICLE**

**Supervised Learning for Finite Element Analysis of Holes under Biaxial Load**

**Wai Tuck Chow**\* **and Jia Tai Lau**

School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore

\*Corresponding Author: Wai Tuck Chow. Email: wtchow@ntu.edu.sg

Received: 02 August 2023 Accepted: 07 March 2024

**ABSTRACT**

This paper presents a novel approach to using supervised learning with a shallow neural network to increase the efficiency of the finite element analysis of holes under biaxial load. With this approach, the number of elements in the finite element analysis can be reduced while maintaining good accuracy. The neural network will be used to predict the maximum stress for holes of different configurations such as holes in a finite-width plate (2D), multiple holes (2D), staggered holes (2D), and holes in an infinite plate (3D). The predictions are based on their respective coarse mesh with only 2 elements along the whole quarter perimeter. The result shows the prediction errors are under 5% for all the listed hole configurations. In contrast, the conventional FEM with the respective coarse mesh has errors above 20%. To achieve similar accuracy, the conventional FEM would require finer mesh with at least 6 elements along the perimeter. Furthermore, this setup is also effective in predicting the maximum stress for the 3D problem. With the aid of supervised learning, the FEM analysis of a coarse mesh with only 2 elements along the quarter perimeter can attain prediction errors of less than 2%. For the same coarse mesh, the conventional FEM has errors above 35%. To achieve similar accuracy for the 3D problem, the conventional FEM would require finer mesh with more than 8 elements along the perimeter. This result shows that supervised learning has great potential to enhance the efficiency of finite element analysis with fewer elements while attaining satisfactory results.

**KEYWORDS**

Supervised learning; finite element analysis; stress concentration

|  |
| --- |
|  |

# **1 Introduction**

The use of finite element analysis (FEA) to solve problems of engineering is popular in many engineering industries due to its ability to represent complex geometry and capture local stress concentration effects. Furthermore, due to the increasing computational power available and the decreasing costs of FEA software, the use of FEA has grown significantly over the years. Yet, modeling a large segment of a component can still require high computational costs. To keep the computational costs practical, coarser mesh and linear elements are often used. As a result, the accuracy of the model is poor, and it is insufficient for fatigue analysis to be carried out. To achieve the accuracy needed, the sub-modeling of the FEA model is performed in high-stress areas. Sub-modeling is the process of creating a solid model of the local geometry at the region of interest within the global model. Subsequently, the local solid model must undergo mesh refinement together with a transfer of boundary conditions from the global model. Unfortunately, sub-modeling is usually performed manually. Therefore, sub-modeling often is a labor-intensive process especially when the part has many features that require many sub-models. As a result, performing these complex 3-D design iterations with local geometry changes or topology optimizations often requires significant time and resources. Moreover, even with sub-modeling, the choice of 3D elements is often linear due to computational limitations. Hence, a huge amount of effort and resources can be saved if machine learning techniques could be applied to increase the accuracy of the finite element analysis of coarse 3D mesh and reduce the need for sub-modeling.

Machine learning is the usage of models to learn from data and make assessments without being explicitly programmed. The goal is to create a trained neural network model based on the pattern of the training data and be able to make predictions involving different sets of data with similar patterns. Improvement in machine learning techniques over the years has made it an effective tool to be used across many different industries such as finance, social research, and medicine. Improvements can be categorized into two areas, software and hardware. For instance, the algorithm using the Scaled Conjugate Gradient (SCG) is a more robust variant of the original conjugate gradient method (CG). It is less computationally complex and faster. SCG is used as a backpropagation algorithm to minimize errors in the neural network. Moller [1] demonstrated that SCG can converge to a minimum point where the CG can not due to the Hessian matrix for global error function not being positive definite. In addition, SCG does not include any user-dependent parameters that are crucial for the success of the algorithm, unlike CG. Furthermore, he showed that the calculation complexity of one iteration of SCG can be more than 3 times less than that of the CG method. Meanwhile, the Levenberg-Marquardt method (LM) is developed to solve non-linear least squares problems. Gavin et al. [2-4] demonstrated that LM is more robust than the Gauss-Newton algorithm in minimization tasks where it can converge to the solution even when the starting point is very far off from the minimum. At the same time, it also converges to the minimum point faster than the gradient descent method. Finally, regularization techniques have been developed to curb the problem of overfitting the neural network (NN) to training data sets. One such method is the Bayesian Regularization (BR). Kayri [5] demonstrated that the BR algorithm with tangent sigmoid transfer function in the framework of a single hidden layer feed-forward neural network is effective in predicting nonlinear relations and non-continuous data such as social data. He also showed that using 2 neurons in such a framework has the highest reliability and robustness.

As for hardware advancements, neural processing units (NPU) are increasingly adopted for machine learning tasks which are traditionally performed by the central processing unit (CPU) or graphics processing units (GPU). NPUs are dedicated processing units optimized for power and area efficiency for matrix math. For instance, the 700MHz NPU used by Google, known as a Tensor Processing Unit (TPU) has a matrix multiplication unit with over 65,000 arithmetic logic units and can process 92 trillion 8-bit operations per second [6]. In addition, Norman et al. [6] expect that rapid improvements will be made in NPU due to its design simplicity. Hence, with the rapid advancement in machine learning algorithms and NPU hardware, there seems to be a huge potential in integrating machine learning with finite element analysis.

Hashash et al. [7] utilized neural network to define the non-linear material constitutive models in the FEA analysis. Manevitz et al. [8] used neural network time series methodology to determine the location for finite element mesh refinement at optimum interval. Chow [9] demonstrated that by using supervised learning algorithms, the errors of coarse mesh can be significantly reduced for holes on a finite-width plate under a uniaxial load. However, more study is required to evaluate if supervised learning can be effectively applied to a more diverse set of problems.

The main objective of this paper is to provide the optimal approach to effectively utilize supervised learning to optimize the FEA of holes under biaxial load in a diverse set of problems. This study also evaluates the extent of reduction in errors by using supervised learning. In this paper, SCG, LM, and BR algorithms were used to train the NN. In addition, pure linear and tangent sigmoid transfer functions will be evaluated. The displacement nodal solutions of the nearest 6 nodes to the quarter-hole’s edge were used as the training parameters. The input data set used to train the NN will consist of a small set of 20 different course mesh of a hole in an infinite-width plate. There will only be 2 elements along the quarter-hole perimeter in the coarse mesh. An analytical solution of a hole under biaxial load will be used as the output target data. Subsequently, the NN will be used to predict the maximum stress of holes under a more diverse range of problems such as holes on a finite-width plate, multiple holes, staggered holes, and holes on the 3D plate.

# **2 Neural Networks**

An artificial neural network consists of neurons, weights, and biases. Using an iterative minimization procedure based on the different back-propagation techniques, the weights are adjusted to reduce the error [10,11]. In addition to weights, biases are included in a neural network to introduce flexibility and better generalization to the neural network. Bias is an extra input to neurons that may allow the neuron to be activated even when inputs are 0. The inclusion of biases is essential in neural networks to prevent overfitting to the training data set [12,13].

Many studies have been done to optimize the weight initializations to curb problems of vanishing gradients and exploding gradients [14,15]. These problems could cause weight updates that are overly small or large which stops the backpropagation from working effectively. However, they are only prevalent in multi-layered neural networks or recurrent neural networks with gradient-based propagation methods. Fortunately, this study uses only one hidden layer in a feed-forward neural network. Hence weight initialization will not be focused on mitigating these problems.

Weight initialization will instead be focused on reducing training time. The Nguyen-Widrow algorithm will be used to initialize the weights and biases for all 3 backpropagation methods. This eliminates most of the weight adjustments, hence only small adjustments are needed to be made during training. The Nguyen-Widrow algorithm has been demonstrated to be able to accelerate the training process and testing accuracy [16-19]. Since the Nguyen-Widrow method requires the layer’s transfer function to have a finite input range, it will only be used when the layer has a tangent sigmoid transfer function. For layers with a pure linear transfer function, weights and biases will be initialized randomly with values between -1 and 1.

The following three sections present the three different back-propagation algorithms used in the study. These three backpropagation methods were chosen as they are the most common methods used in MATLAB machine learning.

# **2.1 Scaled Conjugate Gradient (SCG)**

The Conjugate Gradient (CG) method is implemented as an iterative algorithm in the minimization problem. It can be regarded as being between the gradient descent algorithm and Newton’s method. The gradient descent algorithm requires the calculation of first order derivative which is the gradient to find the minimum of the cost function. It has a slow convergence rate where subsequent steps often contradict previous steps. On the other hand, Newton’s method requires the calculation of second order derivative which has high computational cost. CG method accelerates the convergence rate of the steepest descent by performing the search along conjugate directions such that subsequent steps never contradict previous steps. It also has lower computational costs compared to Newton’s method. The Scaled Conjugate Gradient (SCG) method which is faster and more robust will be used [1].

Consider a quadratic function:

|  |  |
| --- | --- |
|  | (1) |

where is a symmetric positive-definite matrix. Eq. (1) can model the cost function at a given point expressed by Taylor expression, where will be the weight vector. The minimization is an iterative process in which a local minimum in the neighborhood of the initial point is found through subsequent steps of step size and step direction . The residual is the negative gradient of at point . The iteration only stops when the corresponding residual is sufficiently small, signifying the minimum point where the gradient is 0.

The algorithm is started by choosing the initial weight vector , then setting the initial residual and initial step direction . The step size at iteration is computed as follows:

|  |  |
| --- | --- |
|  | (2) |

The weight vector and residual are then updated:

|  |  |
| --- | --- |
|  | (3) |

|  |  |
| --- | --- |
|  | (4) |

The direction vector of the next step, is determined by the residual , then subtracting away any vector that would counter-act the previous direction:

|  |  |
| --- | --- |
|  | (5) |

such that is conjugated to . Subsequently, the weight vector for the next iteration can be obtained like in Eq. (3).

CG method will fail and converge to nonstationary point if matrix is not positive definite. This problem can be resolved by introducing a scalar in CG. This is known as SCG method where the term is estimated as shown in the following equation:

|  |  |
| --- | --- |
|  | (6) |

where and are adjusted at each iteration under the condition that and . The initial value is 0 and is adjusted at each iteration looking at the sign of . This ensures that matrix is always positive definite such that a unique minimizer exists. As the scaling factor, is raised to keep positive, the step size decreases. Lundun et al. [20] demonstrated that this technique reduces the error significantly in a minimization problem.

# **2.2 Levenberg-Marquardt (LM)**

Similar to SCG, the Levenberg-Marquardt (LM) method is an iterative process such that the final weight is obtained through iterations with step . LM method uses a damping factor to decide the step size, with large values of resulting in a more robust update like the gradient descent method in the first iterations and small values of result in a fast update like the Gauss-Newton algorithm in later stages. This avoids the drawbacks of both the Gauss-Newton algorithm and gradient descent method whereby it can converge from any initial state, and thereafter rapid convergence near the vicinity of the minimum error. Many variants of the LM method have been developed [21-23].

Consider the cost function:

|  |  |
| --- | --- |
|  | (7) |

where is the number of datum pairs ; is the hypothesis curve; is the weight vector. The objective is to minimize . The weight vector of each successive step is given by:

|  |  |
| --- | --- |
|  | (8) |

where is the step at iteration; is the corresponding weight vector. To derive , an approximation of the value of the hypothesis function at the next step is made by linearization:

|  |  |
| --- | --- |
|  | (9) |

where is the gradient of with respect to. The cost function at the point can then be approximated as follows:

|  |  |
| --- | --- |
|  | (10) |

The minimum point of the cost function has a gradient of 0. By taking the derivative of and setting the results to zero, can be obtained as follows:

|  |  |
| --- | --- |
|  | (11) |

Eq. (11) is known as Gauss-Newton update. By introducing a damping factor into the denominator, the damped version of this step is now as follows [24,25]:

|  |  |
| --- | --- |
|  | (12) |

where is an identity matrix. The damping factor is adjusted at each iteration such that step size can be controlled. When any iteration results in the worst approximation ( then is increased. Otherwise, as the approximation improves, is decreased. Eq. (12) shows that when is 0, the update follows a Gauss-Newton update. When is infinite, loses its importance and the update follows a gradient descent update. The iteration stops when the value of either the step or the cost function reduces to below a predefined value.

# **2****.3 Bayesian Regularization (BR)**

To reduce overfitting, this method uses regularization parameters and to regulate between reducing errors and weight size. Mackay [26] presented an extensive work on the BR method. He showed that the objective function to be minimized is a linear combination of the sum of squares of errors and the sum of squares of network weights . The objective function is as follows:

|  |  |
| --- | --- |
|  | (13) |

When , the training emphasizes on reducing errors. Alternatively, when , the training will emphasize on reducing weight size. The goal of Bthe R method is to have a changing after every iteration depending on the regularizing parameters, such that at the end of the training, the resulting network has good generalization qualities. Foresee et al. [27] presented the detailed derivation of and . They contributed to the BR algorithm by applying Gauss-Newton approximation to the Hessian matrix of which is required to calculate and . This step reduces the calculation complexity of the BR algorithm.

The application of Bayesian Regularization (BR) within the framework of the LM algorithm will be used [27]. At each iteration, one step of the LM algorithm will be used to minimize . Thereafter, the regularization parameters and can be calculated and compared. As and change for each successive iteration, changes, and the minimum point will move. The iteration will stop when does not significantly change in subsequent iterations resulting in the convergence of the minimum point.

# **3 Methods**

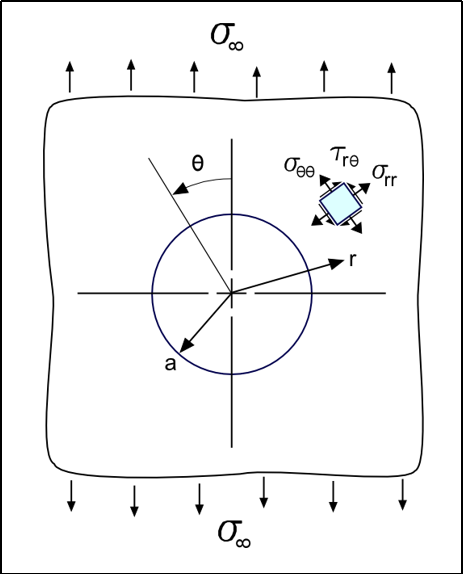
This section will first present the analytical stresses and displacement of a hole in an infinite-width plate under uniaxial load. Next, it will introduce the approach to how the analytical solution can be integrated with supervised learning for both uniaxial load and biaxial load.

# **3.1 Stress and Displacement around a Hole under uniaxial load**

In Fig. 1**,** a hole in an infinite-width plate is subjected to uniaxial tensile load where the tensile load only exists in one primary axis. The stress field solution under uniaxial load [28] can be integrated to obtain the displacement field as follows:

|  |  |
| --- | --- |
|  | (14) |
|  |  |
|  |  |
|  | (15) |

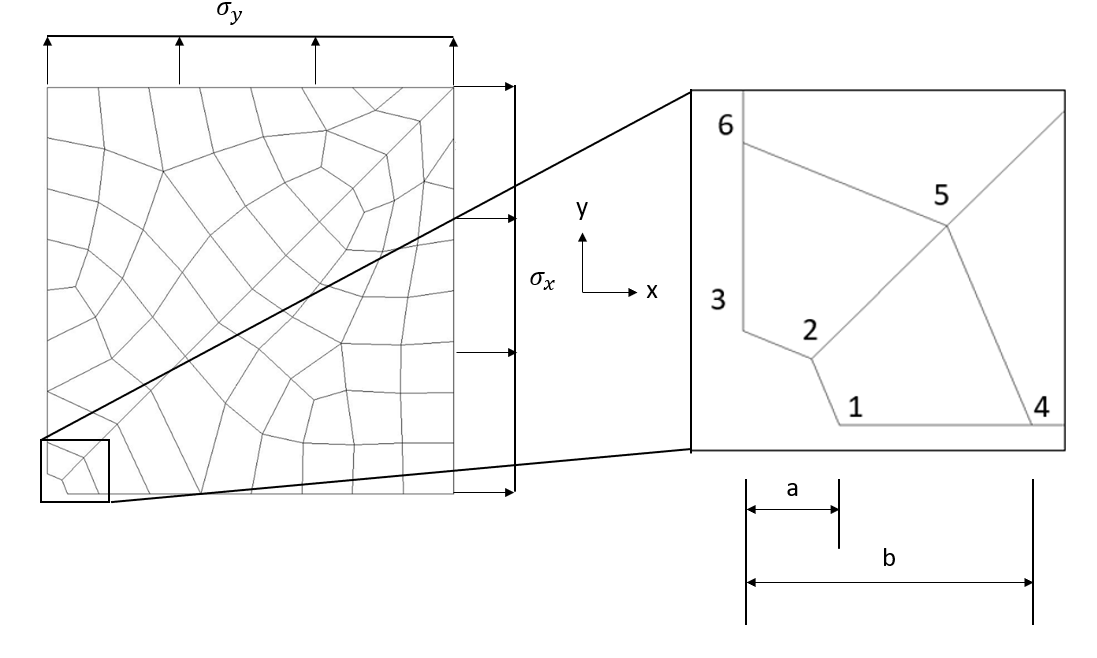
where is the applied stress, is the radius of the hole, and are the polar coordinates of a point in the element.



**Figure 1:** Figure of a hole in an infinite plate under uniaxial tensile load

# **3.2 Approach to apply Supervised Training for Uniaxial load**

The input data set used to train the NN will consist of a set of 20 course mesh of a hole in an infinite-width plate. An analytical solution of a hole under biaxial load will be used as the output target data. A coarse mesh of the quarter-hole is shown in Figure 2. The infinite-width effect is approximated using half width = 20 x hole radius. Ansys 2D linear element, Plane 182 [29], is used. There are only 2 elements along the perimeter of the quarter-hole in the coarse mesh. The displacement nodal solution of the nearest 6 nodes to the quarter-hole’s edge is used as the training parameters. The 6 nodes are shown in Figure 2. The transfer function of the first layer will be either tangent sigmoid or pure linear, while the transfer function of the output layer will be set to pure linear.



**Figure 2:** Coarse mesh of a quarter-hole with 2 elements along the perimeter.

The material property is based on Aluminum, where and . The variations of parameters are shown in Table 1. The mesh is subjected to unit stress load, *psi* and There is a total of 20 mesh variations. Figure 3 shows a few samples of the different coarse mesh variations. To train the model to be sensitive to the magnitude change in displacement, the training is done using a simple linear scale of 70% and 150% on the training parameters which are the displacements of the 6 nodes.

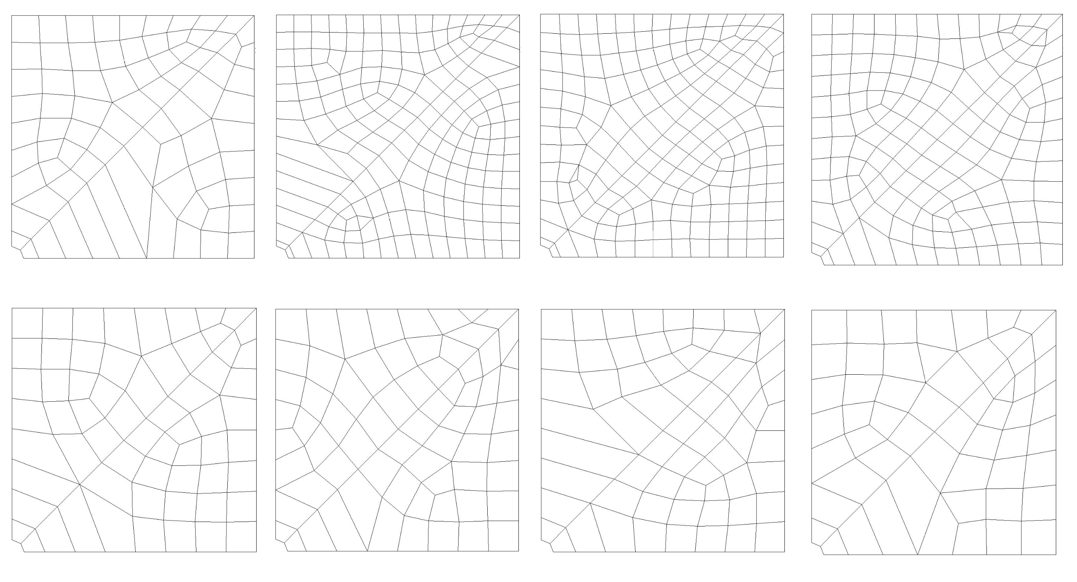
|  |  |
| --- | --- |
|  | (16) |

where *i* is node = 1 to 6 and .

**Table 1:** Variation of parameters for uniaxial load

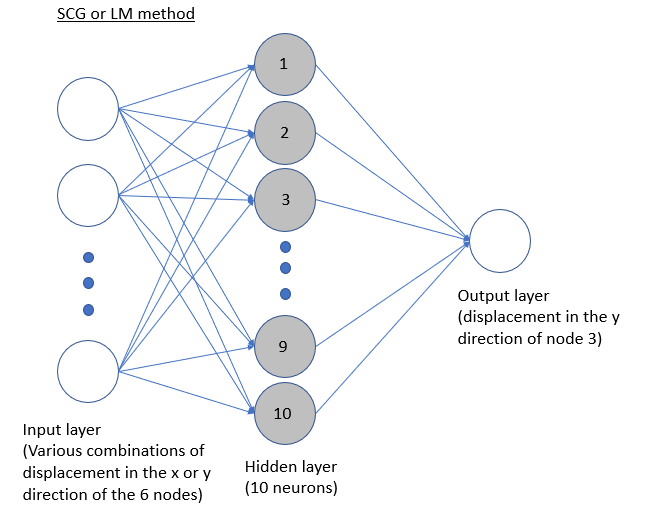
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Total Sampling Size\* | Parameters | Min | Max | Interval |
| 40 |  | 1.5 | 3 | 0.5 |
|  | 0.6 | 1.4 | 0.2 |

\*The total sampling size is obtained after applying a linear scale of 70% and 150% as shown in Eq. (16), resulting in a total of 2 load variations

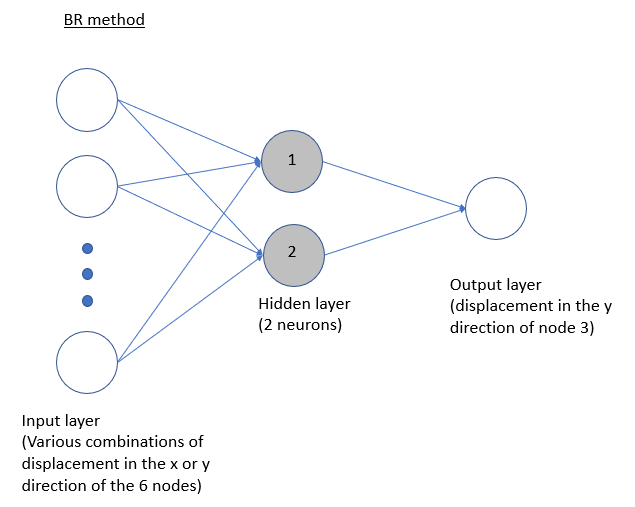


**Figure 3:** Samples of mesh variations

The neural network consists of a single hidden layer as shown in Figure 4. The effectiveness of the neural network in predicting the maximum stress of a single hole in a finite-width plate will be investigated using different combinations of training parameters based on the displacements of the 6 nodes shown in Figure 2. Two approaches to conduct supervised training will be used, one for uniaxial and the other for biaxial load.



1. Schematic of NN for SCG or LM method



1. Schematic of NN for BR method

**Figure 4:** Neural network with 1 output

10 neurons will be used for the SCG and LM method, while 2 neurons will be used for the BR method as the BR method requires fewer neurons to perform effectively as demonstrated by Kayri [5]. Numerical experimental was conducted and the result shows the accuracy improvement starts to taper off as the number of neurons for SCG and LM methods approaches 10. This could be because there are only 10 input parameters from the 6 nodes (2 DOF X 6 nodes with 2 less due to symmetric boundary conditions). The target consists of the analytical solution of only of node 3 obtained by Eq. (14). For the supervised training, both the input and target data are normalized to a magnitude near 1. As a result, the displacement on the 6 nodes is normalized with the of node 3 obtained from coarse mesh.

|  |  |
| --- | --- |
|  | (17) |

where *i* is node = 1 to 6 and . A normalized displacement, will be predicted by the neural network. Considering the stress concentration factor of 3 (Kt of a hole in an infinite-width plate), the predicted maximum stress, can be obtained from the predicted normalized displacement as follows:

|  |  |
| --- | --- |
|  | (18) |

where is the analytical radial displacement at .

# **3.3 Stress and Displacement around a Hole under biaxial load**

A diagram of a square with arrows

Description automatically generated

**Figure 5** Figure of a hole in an infinite plate under biaxial tensile load

Since deformation is small, assuming that the response is linear, the principle of superposition can be applied to the uniaxial load solution [28]. In Fig. 5, the analytical stress for hole under applied biaxial stress, and , with no applied shear stress will thus be as follows:

|  |  |
| --- | --- |
|  | (19) |

|  |  |
| --- | --- |
|  | (20) |

|  |  |
| --- | --- |
|  | (21) |

where

By applying superposition on plane stress displacements under uniaxial load given by Eqs. (14) and (15), the radial and tangential displacement of the hole under biaxial load and plane stress conditions are as follows:

|  |  |
| --- | --- |
|  | (22) |

|  |  |
| --- | --- |
|  | (23) |

where

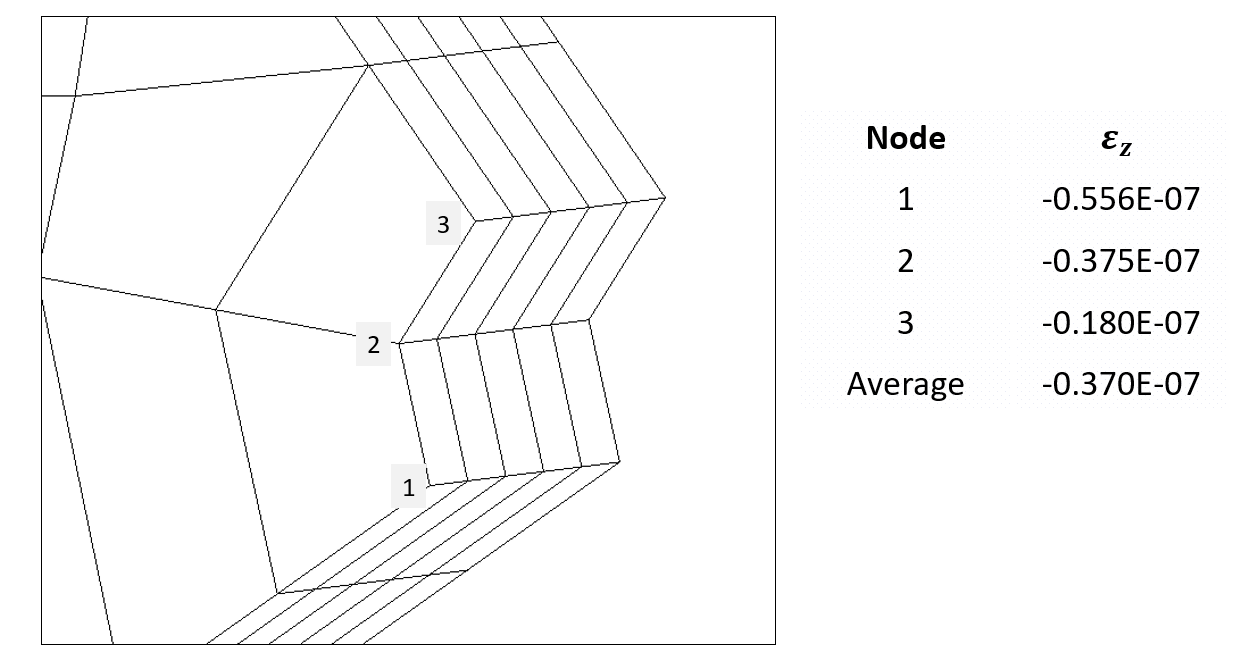
Aside from plane stress conditions, this paper also evaluates the effectiveness of a 2D coarse mesh training data set generated under general plane strain conditions. Under general plane strain conditions, the strain in the z-direction is not zero. Under Hooke’s law with general plane strain condition, the polar strains are defined as a function of as follows:

|  |  |
| --- | --- |
|  | (24) |
|  | (25) |
|  |  |

Using Eqs. (19) and (20) to convert the polar stresses in Eqs. (24) and (25) to applied stresses in the cartesian coordinate system, then by integrating the strains, the displacements in terms of applied stresses and are derived:

|  |  |
| --- | --- |
|  | (26) |
|  |  |
|  | (27) |

Notice that Eq. is a function of the z-component strain . Hence, successful prediction of 3the D mesh by a neural network trained using a 2D coarse mesh data set (under general plane strain condition) depends on the user input parameter .



**Figure 6:** from a 3D coarse mesh

To obtain an average will be taken from the z-component strain of the 3 nodes from a coarse 3D mesh as shown in Fig. 6. The resulting value will be used to generate the 2D coarse mesh under general plane strain conditions from APDL [29], as well as to obtain the analytical solution. With the 2D coarse mesh and the analytical solution, the neural network can then be trained to predict stresses for the case of a 3D model.

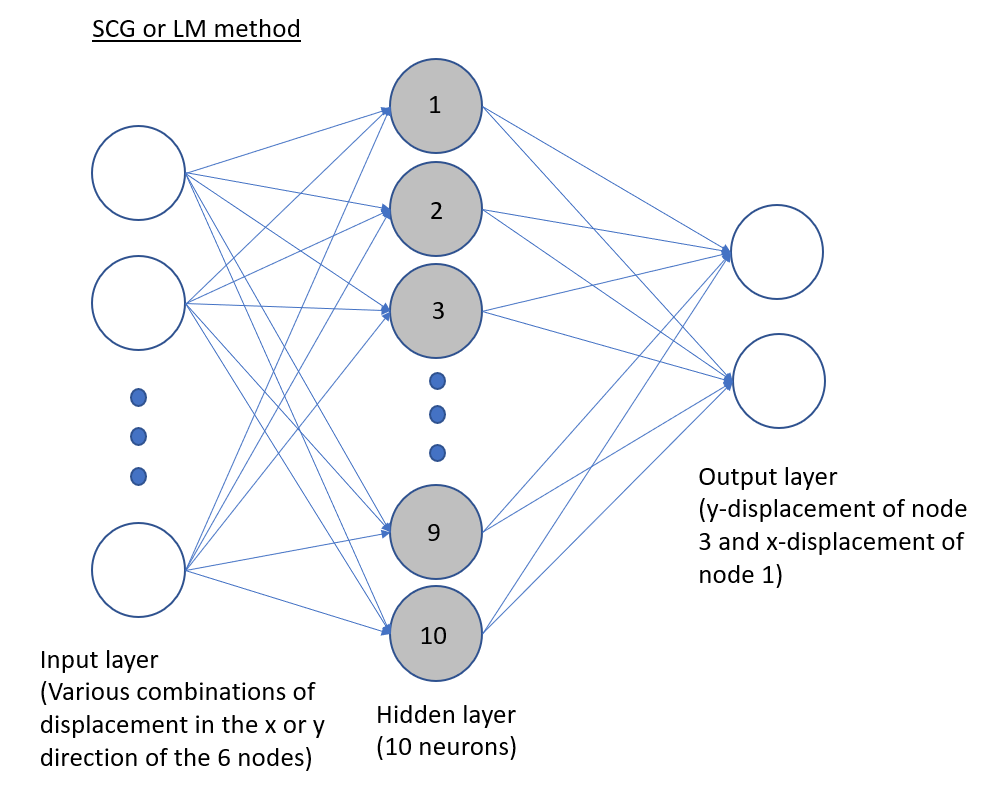
# **3.4 Approach to apply Supervised Training for Biaxial load**

The neural network is trained with the same set of 2D coarse mesh used in the uniaxial load case. Fig. 7 shows the neural network models used for the analysis. In Figure 2, is fixed while is varied such that ranges from -1 to 1. Different intervals of are computed to investigate the minimum sampling size required to obtain a reasonably good prediction. shows the variation of parameters for each sampling size.

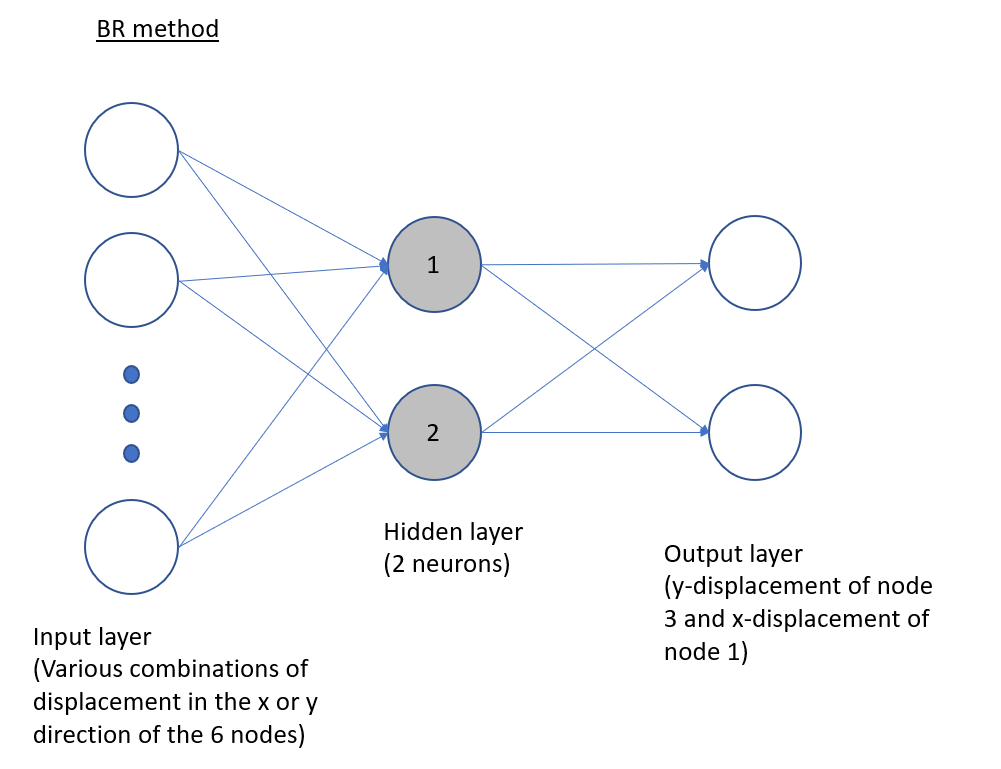
**Table 2:** Variation of parameters and biaxial loads

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Total Sampling Size \* | Parameters | Min | Max | Interval |
|  |  | 1.5 | 3 | 0.5 |
|  | 0.6 | 1.4 | 0.2 |
| 200 |  | -1 | 1 | 0.5 |
| 440 |  | -1 | 1 | 0.2 |
| 840 |  | -1 | 1 | 0.1 |

\*The total sampling size is obtained after applying a linear scale of 70% and 150% as shown in Eq. (16), resulting in the doubling of load variations for each case of interval.



1. Schematic of NN for SCG or LM method



1. Schematic of NN for BR method

**Figure 7:** Neural network with 2 outputs

The target data consists of analytical solutions of of node 3 and of node 1 obtained by Eqs. (22) for plane stress condition or Eqs. and (27) for general plane strain condition. This is indicated by the 2 outputs shown in 7. The input and target data are normalized as follows:

|  |  |
| --- | --- |
|  | (28) |
|  |  |

where *i* is node = 1 to 6 and . The following part shows the computation of predicted stress from predicted displacements. The displacements under uniaxial load, and is related to the displacements under biaxial load, and by the following:

|  |  |  |
| --- | --- | --- |
|  |  | (29) |
|  |  |  |

When substituting and in Eq. (29) for the predicted displacements, the stresses, and will be the predicted equivalent applied stresses for a hole in an infinite-width plate. and are the analytical displacements under uniaxial load. Subsequently, by applying Cramer’s rule, the predicted applied stresses are:

|  |  |  |
| --- | --- | --- |
|  |  | (30) |

|  |  |  |
| --- | --- | --- |
|  |  | (31) |

Finally, by using Eq. (20), the predicted maximum stress which is the tangential stress at node 1 can be obtained.

# **4 Results**

In this study, Matlab’s library [30], *fitnet*, is used for the neural network supervised training with ‘trainscg’ (SCG), ‘trainbr’ (BR), and ‘trainlm’ (LM) as the backpropagation method. For SCG and LM methods, 70% of the training sample size is used for the training of neural networks, 15% for the validation of the neural network model, and the remaining 15% for the testing of the accuracy of the model. For the BR method, which is a self-validating algorithm, 85% of the training sample size is used for training and 15% for testing. At the hidden layer, either tangent sigmoid or pure linear transfer function will be used. Their effectiveness will be evaluated against each other. In addition, uniaxial and biaxial approaches are compared.

For the prediction of 2D mesh, plane stress condition is used to generate the 2D coarse mesh training data set. For the prediction of 3D mesh, the general plane strain condition is used to generate the 2D coarse mesh training data set [9].

# **4.1 Infinite-width Hole: Neural Network Option**

The accuracy of the test result based on the displacement at all 6 nodes as training parameters is shown in Table 3. BR method with tangent sigmoid transfer function and LM method with pure linear transfer function offer low prediction errors for both approaches.

**Table 3:** Root Mean Squared Error (RMSE) of the test sample using different neural models and transfer functions. Training is done using and of all 6 nodes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE (%) under Uniaxial approach\* | | RMSE (%) under Biaxial approach\*\* | |
| Tangent Sigmoid | Pure Linear | Tangent Sigmoid | Pure Linear |
| SCG | 45.87 | 34.52 | 15.54 | 13.08 |
| BR | 2.32 | 60.98 | 1.65 | 19.23 |
| LM | 6.26 | 2.80 | 2.10 | 1.52 |

\* Based on 20 mesh variations and 2 load variations, training sample size of 40;

\*\* Based on 20 mesh variations and 10 load variations, training sample size of 200;

Maximum error with 300 trials

To further improve the result, the study evaluates the displace field on the element attached to the maximum stress location at Node 1. In this element, Node 2 and Node 5 have non-zero . The training result for this option is shown in Table 4. This step is important because using fewer training parameters reduces the tendency for the neural network model to overfit the training model, which might result in lower prediction error when applying the NN model to a separate set of problems. Table 3 and Table 4 show that with the BR method, the tangent sigmoid transfer function offers lower errors than the pure linear transfer function. The opposite is true for SCG and LM methods. This is due to the synergy between the BR method and the tangent sigmoid transfer function as demonstrated by Kayri [5]. Fig. 8 demonstrates the verification of obtained neural network models for the biaxial approach. Fig. 8a and 8b show that the training of NN under biaxial load by both cases offers prediction errors that are unbiased and centered at 0, with the maximum absolute errors of less than 4%.

**Table 4:** Root Mean Squared Error (RMSE) of the test sample using different neural models and transfer functions. Training is done using and of Node 2 and Node 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE (%) under Uniaxial approach\* | | RMSE (%) under Biaxial approach\*\* | |
| Tangent Sigmoid | Pure Linear | Tangent Sigmoid | Pure Linear |
| SCG | 22.56 | 19.96 | 14.55 | 7.93 |
| BR | 2.52 | 51.19 | 2.16 | 13.76 |
| LM | 9.47 | 5.42 | 2.28 | 2.23 |

\* Based on 20 mesh variations and 2 load variations, training sample size of 40;

\*\* Based on 20 mesh variations and 10 load variations, training sample size of 200;

Maximum error with 300 trials

A graph with different colored bars

Description automatically generatedA graph of error histogram with red green and blue bars

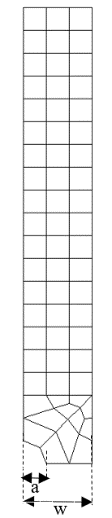
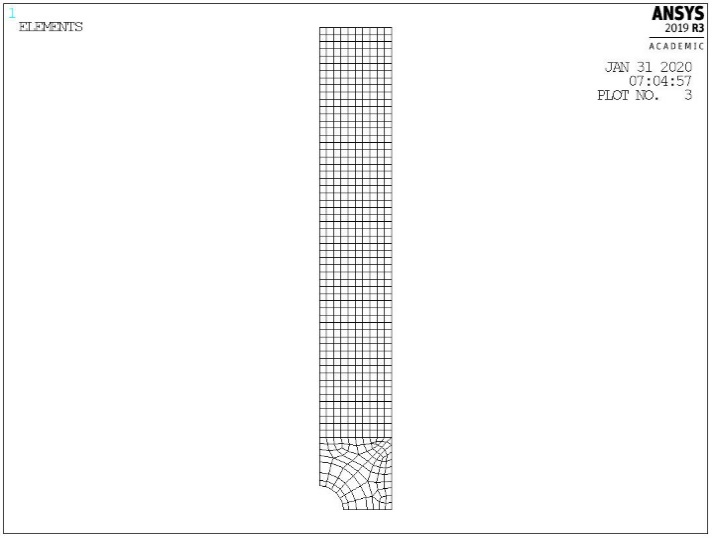
Description automatically generated

|  |  |
| --- | --- |
| 1. and of all 6 nodes as training parameters | 1. and of Node 2 and 5 only as training parameters |

**Figure 8:** Error histogram of training using based on biaxial approach using Levenberg-Marquardt method with pure linear transfer function.

# **4.2 Finite-width Single Hole**

The neural network model that is trained based on the hole in an infinite-width plate problem will be used to predict the maximum stresses on a hole in a finite-width plate problem. In the infinite-width model, the side wall is far from the hole such that there is no interaction between the stress concentration of the hole with the free edge on the side. However, with the finite-width geometry, the hole is closer to the free edge on the side. As a result of the interaction between the hole and the free edge, the stress concentration of the hole would increase. The analysis of the finite-width model is to evaluate if supervised learning can be used to support the finite element method to calculate stress accurately with an alternate geometry with coarse mesh size. Tests are conducted for a range of (width to hole radius) of 3, 4, 5, and 6. The coarse mesh for this problem is shown in Fig. 9a.

****

|  |  |
| --- | --- |
| 1. Coarse mesh with 2 elements   along the perimeter | 1. Fine mesh with 6 elements   along the perimeter |

**Figure 9:** Coarse and fine mesh of a quarter-hole with finite-width of .

The mesh is constructed by linear elements and only has 2 elements at the quarter-hole perimeter. The result is compared against the Roark’s solution [31] given as follows:

|  |  |
| --- | --- |
|  | (32) |

where

|  |  |
| --- | --- |
|  | (33) |

|  |  |
| --- | --- |
|  | (34) |

The result in Table 5 shows the prediction errors using the neural network model based on both and of all 6 nodes are high. In this case, the tangent sigmoid transfer function performs worse than the pure linear transfer function even with the BR method as shown in Table 5. The poor performance of the tangent sigmoid transfer function could be attributed to the fact that the tangent sigmoid transfer function has properties that are more suitable for classification problems than regression problems. This is indicated by the shape of the graph of a sigmoid function where it levels off to an asymptote at both ends. In this study, using the trained neural network to predict maximum stresses for a different set of problems can be regarded as a regression problem. Hence, the pure linear transfer function may perform better as an approximator. However, the prediction errors using the pure linear transfer function are still high. Therefore, the neural network model trained using the infinite hole mesh cannot be applied to the finite hole mesh under this setup.

**Table 5:** Prediction Error (%) using different neural models and transfer functions. Training done using and of all 6 nodes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Absolute Error (%) under Uniaxial approach\* | | Absolute Error (%) under Biaxial approach\*\* | |
| Tangent Sigmoid | Pure Linear | Tangent Sigmoid | Pure Linear |
| SCG | 87.40 | 77.51 | 81.51 | 19.80 |
| BR | 34.97 | 18.39 | 18.27 | 13.62 |
| LM | 74.79 | 21.40 | 112.30 | 16.45 |

\* Based on 20 mesh variations and 2 load variations, training sample size of 40;

\*\* Based on 20 mesh variations and 10 load variations, training sample size of 200;

Maximum error with 300 trials, conducted for a range of *w/a*: 3, 4, 5 and 6

To evaluate if the trained model can be further improved by focusing the training on just the element with the highest stress, the training is based on and of Node 2 and Node 5. shows that using just and of Node 2 and Node 5 indeed reduced the prediction error under pure linear transfer function as using fewer parameters reduces the tendency to overfit to training model. With only 20 mesh variations and a sample size of 40, the uniaxial method is unable to provide a satisfactory prediction for the finite hole. On the other hand, with the same 20 mesh variations but with a sample size of 200, the biaxial approach can offer low prediction errors under the LM method and BR method with pure linear transfer function.

**Table 6:** Prediction Error (%) using different neural models and transfer functions. Training is done using and of Node 2 and Node 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Absolute Error (%) under Uniaxial approach\* | | Absolute Error (%) under Biaxial approach\*\* | |
| Tangent Sigmoid | Pure Linear | Tangent Sigmoid | Pure Linear |
| SCG | 66.17 | 27.40 | 57.70 | 6.39 |
| BR | 55.81 | 13.96 | 4.26 | 3.98 |
| LM | 86.82 | 13.02 | 325.27 | 3.55 |

\* Based on 20 mesh variations and 2 load variations, training sample size of 40;

\*\* Based on 20 mesh variations and 10 load variations, training sample size of 200;

Maximum error with 300 trials, conducted for a range of *w/a*: 3, 4, 5 and 6

In Table 7, the stress concentration, , has a maximum error of 26.2% for the conventional FEA analysis. However, with the aid of the neural network, the maximum error is reduced to just 3.55%. To obtain similar accuracy without supervised learning, a much finer mesh is required. A fine mesh with 6 elements along the perimeter, as shown in 9b, would be necessary.

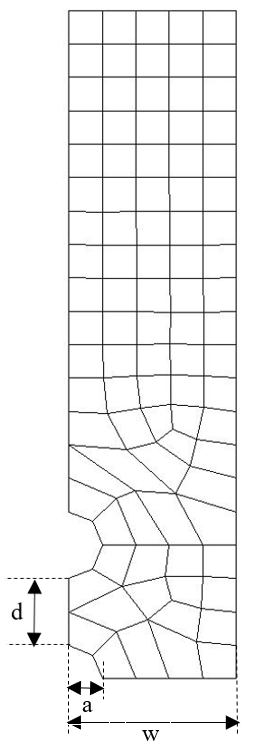
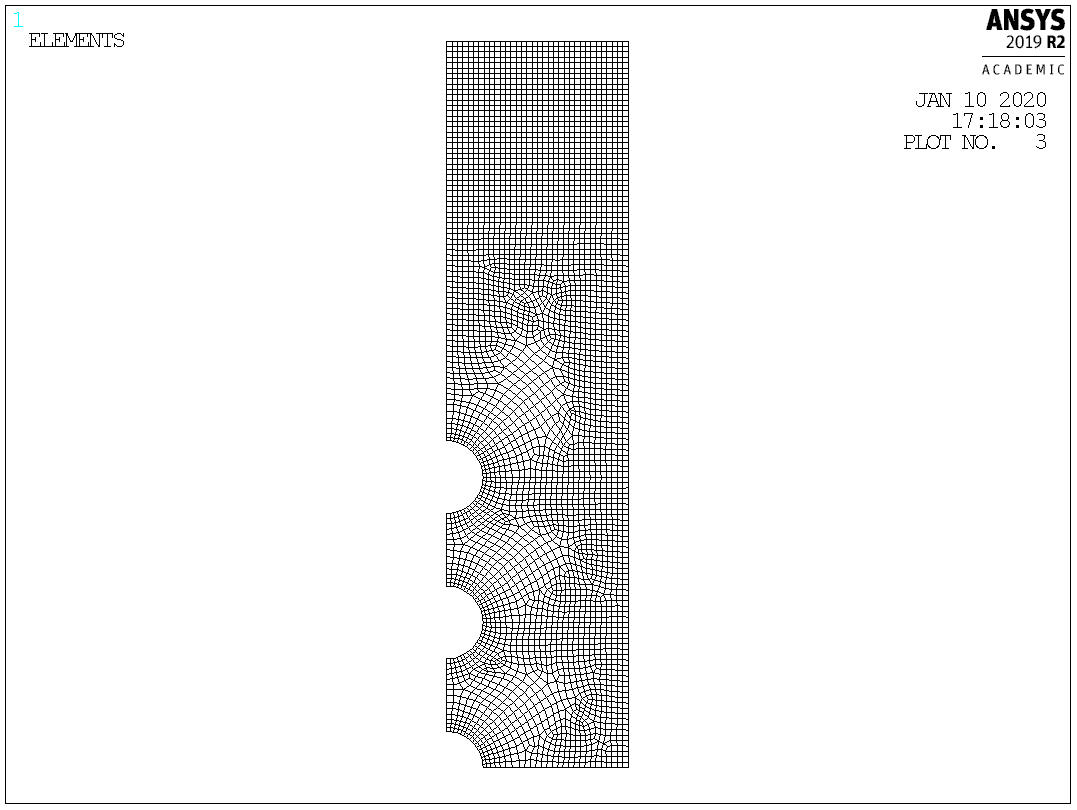
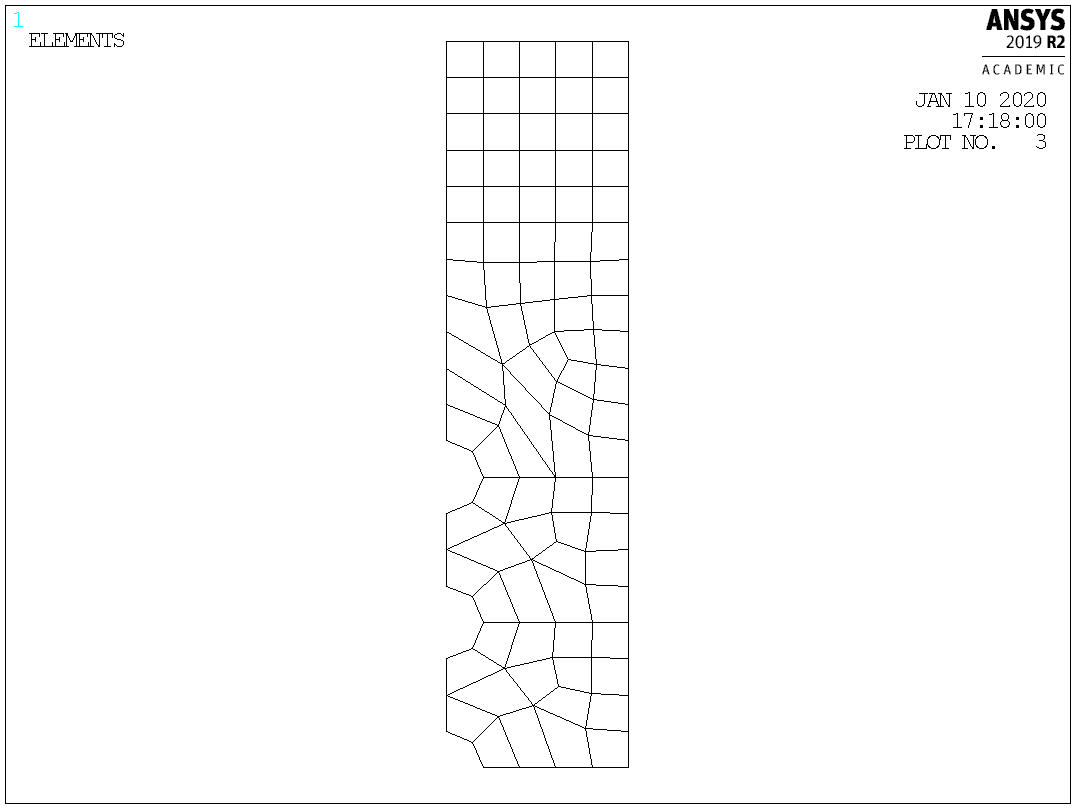
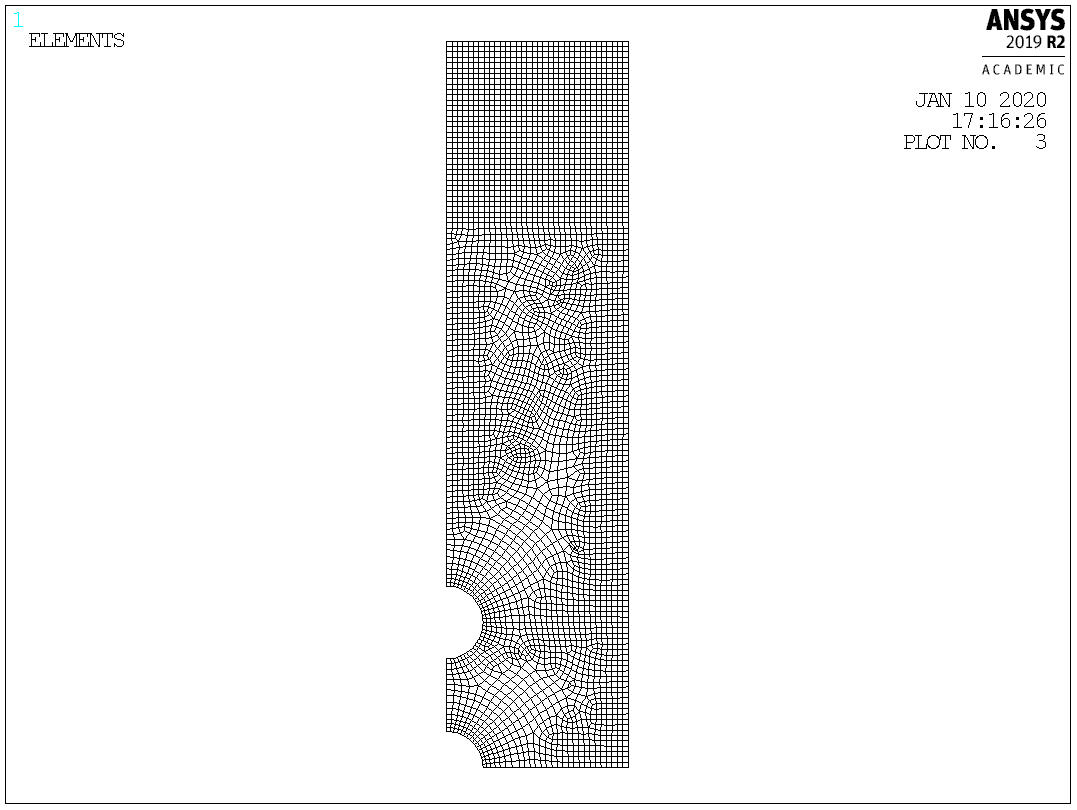
**Table 7:** A sensitivity study on the mesh size where *n* is the number of elements on the hole perimeter.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *w/a=*3 | | *w/a=*4 | | *w/a=*5 | | *w/a=*6 | |
|  | *n* | Kt | Error | Kt | Error | Kt | Error | Kt | Error |
| Roark |  | 3.460 |  | 3.230 |  | 3.135 |  | 3.088 |  |
| Biaxial Approach\* | 2 | 3.340 | -3.47% | 3.115 | -3.55% | 3.112 | -0.73% | 3.031 | -0.85% |
| FEM | 2 | 2.552 | -26.2% | 2.420 | -25.1% | 2.404 | -23.3% | 2.377 | -23.0% |
| FEM | 4 | 3.281 | -5.2% | 3.072 | -4.9% | 2.984 | -4.8% | 2.940 | -4.8% |
| FEM | 6 | 3.388 | -2.08% | 3.188 | -1.30% | 3.099 | -1.15% | 3.057 | -1.85% |

\* Using and of Node 2 and 5 as training parameters, the Levenberg-Marquart algorithm, pure linear function, based on the maximum error with 300 trials, using the biaxial approach.

Both the LM and BR methods offer comparable results. Since the LM method has lower computational costs, the LM method would be the preferred method. The LM method will thus be applied on mesh with very different configurations such as multiple holes and staggered holes on finite-width plates for further tests. The biaxial approach will be used since it offers lower error than the uniaxial approach and has the advantage of being sensitive to biaxial loads.

# **4.3 Multiple holes on finite-width plate**



|  |  |
| --- | --- |
| 1. Coarse mesh with 2 elements   along the perimeter (Plane 182),  Left: 3 holes; Right: 5 holes | 1. Fine mesh with 16 elements   along the perimeter (Plane 183)  Left: 3 holes; Right: 5 holes |

**Figure 10:** Coarse and fine mesh of multiple holes with finite-width of , *d* = 2

The holes are separated by distance, ***d***, from edge to edge of holes as shown in Figure 10. To evaluate the accuracy of the prediction, the Kt values are obtained from the fine mesh using quadratic element Plane 183 as shown in Figure 10b [9]. These Kt values are compared to the Kt values predicted by the neural network using displacements from coarse mesh shown in Figure 10a. The variations in the mesh are shown in Table 8, with a fixed *w/a =* 5.

**Table 8:** Variations in finite multiple hole mesh

|  |  |  |  |
| --- | --- | --- | --- |
|  | Min | Max | Interval |
|  | 2 | 4 | 1 |

Table 9 investigates the minimum number of training sampling sizes needed to obtain good predictions for mesh with different configurations using biaxial-trained NN. To allow the NN to fit more loosely to the single hole in an infinite plate problem, and of Node 2 and 5 are used as training parameters such that the NN can effectively predict the stresses for multiple holes on finite plate problems. Table 9 shows that a sampling size of 440 lines offers significant improvement over 200 lines while the sampling size of 840 lines does not offer much improvement. The biaxial approach will be used to illustrate the effects of different training sampling sizes.

**Table 9:** Prediction errors (%) using different sampling sizes for the case of 3 holes, *w/a =* 5

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediction error (%) under Biaxial approach\* | | |
|  | = 2 | = 3 | = 4 |
| sampling size of 200 | -13.47 | -4.88 | -3.02 |
| sampling size of 440 | 0.48 | 3.01 | 2.07 |
| sampling size of 840 | 0.37 | 2.89 | 1.97 |

\* Using and of Node 2 and 5 as training parameters, the Levenberg-Marquart algorithm, a pure linear function, based on a maximum absolute error with 300 trials

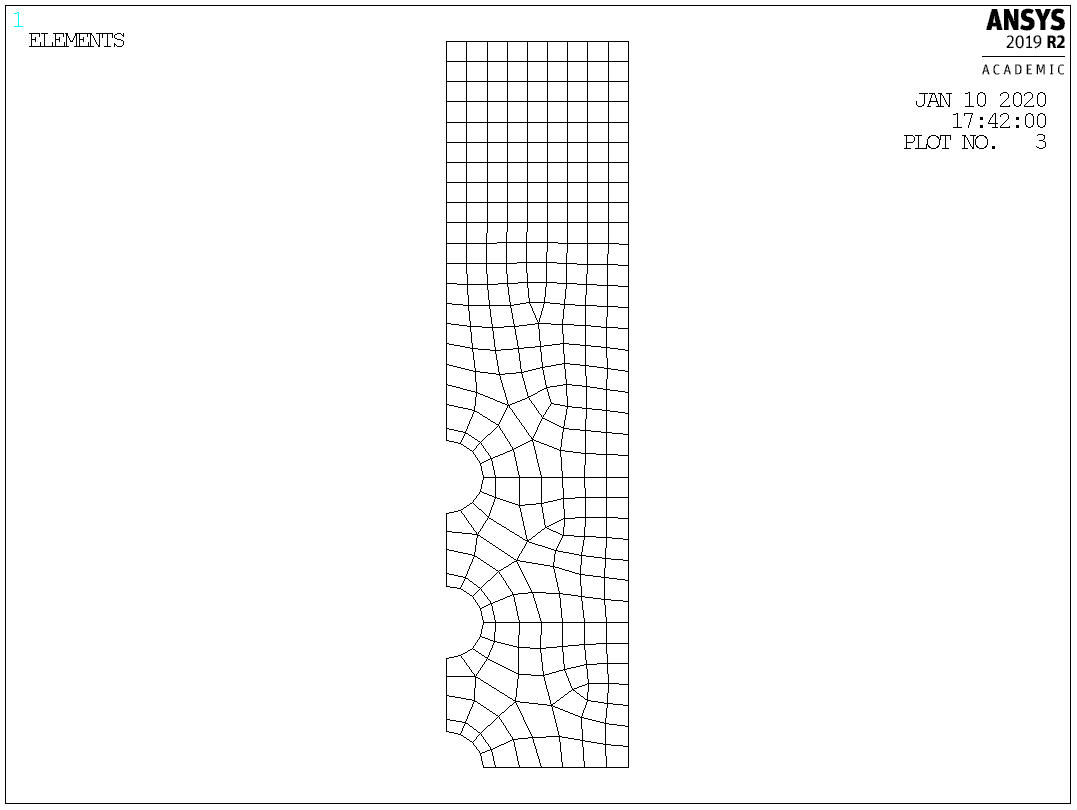
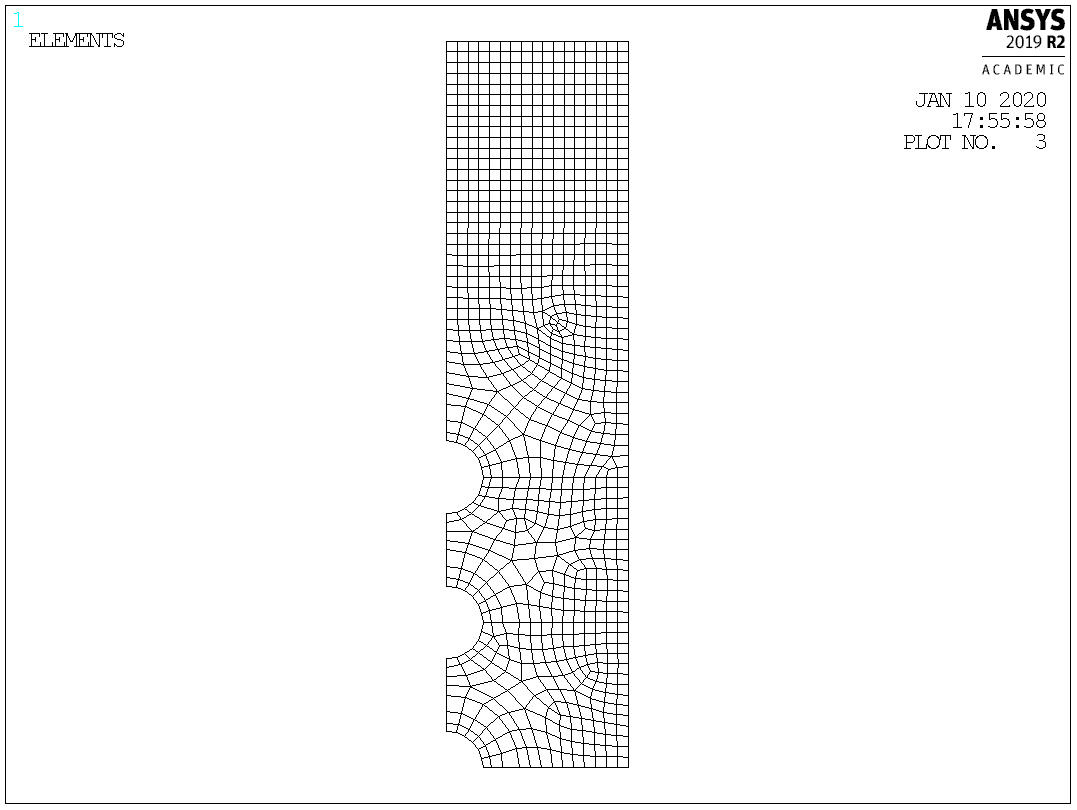
To ensure that the NN is also effective in predicting similar problems, the prediction error of a quarter finite plate with 3 holes as shown in Figure 10 is studied. Table 10 shows that errors are generally higher than those in Table 9. However, the difference is not significant.

**Table 10:** Prediction errors (%) for case of 5 holes, *w/a =* 5

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediction error (%) under Biaxial approach\* | | |
|  | = 2 | = 3 | = 4 |
| sampling size of 440 | -0.74 | 3.27 | 2.14 |

\* Using and of Node 2 and 5 as training parameters, the Levenberg-Marquart algorithm, a pure linear function, based on a maximum absolute error with 300 trials

A sensitivity study will be conducted for the case of 5 holes since it has higher prediction errors. Table 11 shows that even for 5 holes, to achieve similar accuracy, a fine mesh with 6 elements along the perimeter as shown in Figure 1b would be necessary. The prediction errors in Table 11 are low and comparable to that of the single hole in finite-width plate problem. This is because of the absence of any stagger of the holes which may introduce additional shear stresses. Therefore, the stress field is still similar to that of a single-hole problem. As a result, the trained NN model will still be able to make satisfactory predictions in this problem. Furthermore, a *w/a* of 5 is used in this problem. With higher *w/a*, the prediction error is expected to be low as it has higher similarity with the infinite-width problem that is used to train the NN.



|  |  |
| --- | --- |
| 1. Coarse mesh with 4 elements along the perimeter (Plane 182) | 1. Coarse mesh with 6 elements along the perimeter (Plane 182) |

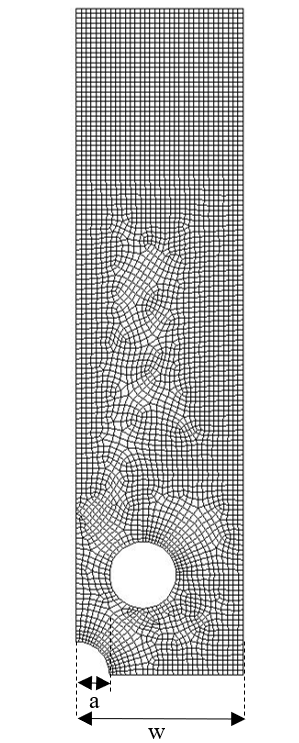
**Figure 11:** Various fine mesh of 5 holes

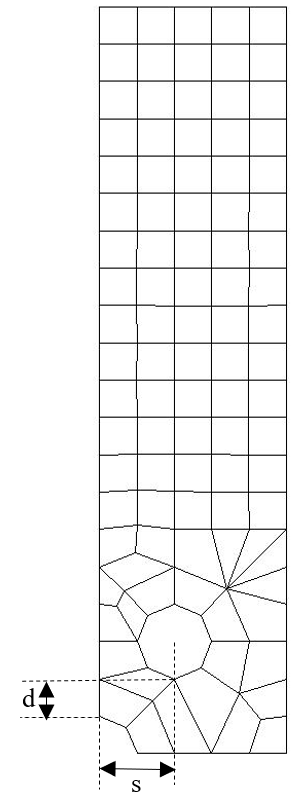
**Table 11:** A sensitivity study on the mesh size for 5 holes where *n* is the number of elements on the hole perimeter. (based on fine mesh shown in Figure 10b)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *d/a =* 2 | | *d/a =*3 | | *d/a =* 4 | |
|  | *n* | Kt | Error | Kt | Error | Kt | Error |
| FEM (Quadratic element, Plane 183) | 16 | 2.512 |  | 2.693 |  | 2.827 |  |
| Biaxial Approach\*  (Linear element, Plane 182) | 2 | 2.493 | -0.74% | 2.781 | 3.27% | 2.887 | 2.14% |
| FEM (Linear element, Plane 182) | 2 | 1.940 | -22.77% | 2.132 | -20.85% | 2.227 | -21.25% |
| FEM (Linear element, Plane 182) | 4 | 2.344 | -6.67% | 2.528 | -6.15% | 2.661 | -5.90% |
| FEM (Linear element, Plane 182) | 6 | 2.469 | -1.68% | 2.664 | -1.11% | 2.796 | -1.11% |

\* Using and of Node 2 and 5 as training parameters, training sampling size of 440, Levenberg-Marquart algorithm, pure linear function, based on maximum absolute error with 300 trials

# **4.4 Staggered holes on finite-width plate**





|  |  |
| --- | --- |
| 1. Coarse mesh with 2 elements   along the perimeter (Plane 182) | 1. Fine mesh with 16 elements   along the perimeter (Plane 183) |

**Figure 12:** Coarse and fine mesh of staggered holes with finite-width of ; *d* = 1; = 2

There is a stagger of *s* = 2\*radius of the hole. From 12a, only the elements adjacent to the quarter-hole are relevant. The mesh distortions away from the quarter-hole are inconsequential. Like the previous section, the Kt values of fine mesh shown in Figure 12b are compared to the Kt values predicted by the neural network using displacements from coarse mesh. The variations of the mesh are shown in Table 12 with a fixed *w/a* = 5.

**Table 12:** Variations in finite multiple-hole mesh

|  |  |  |  |
| --- | --- | --- | --- |
|  | Min | Max | Interval |
|  | 1 | 3 | 1 |

Since node 5 is in the vicinity of the other hole, its stresses may be affected by the stress field of the other hole. This may increase discrepancies from that of a single hole. Table 13 evaluates if prediction error can be reduced by removing node 5 from the training parameters. Indeed, prediction errors are lowered by using and of only Node 2. Using fewer training parameters also reduces the tendency for NN to overfit the training setup.

**Table 13:** Prediction error (%) using a different set of training parameters for the case of *s/a* = 2, *w/a* = 5

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediction error (%) under Biaxial Approach\* | | |
|  | = 1 | = 2 | = 3 |
| and of Node 2 and 5 | 8.12 | 6.81 | 4.38 |
| and of Node 2 | 4.61 | 4.07 | 3.11 |

\*Based on the training sampling size of 440, Levenberg-Marquart algorithm, pure linear function, maximum absolute error over 300 trials

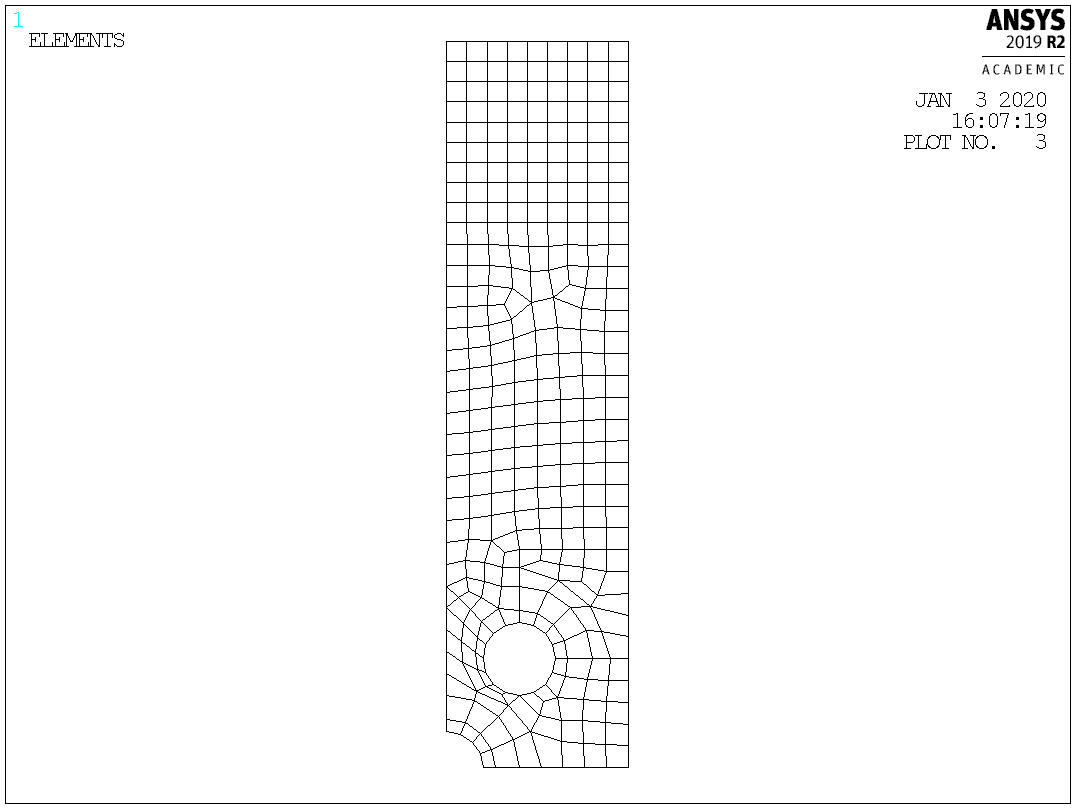
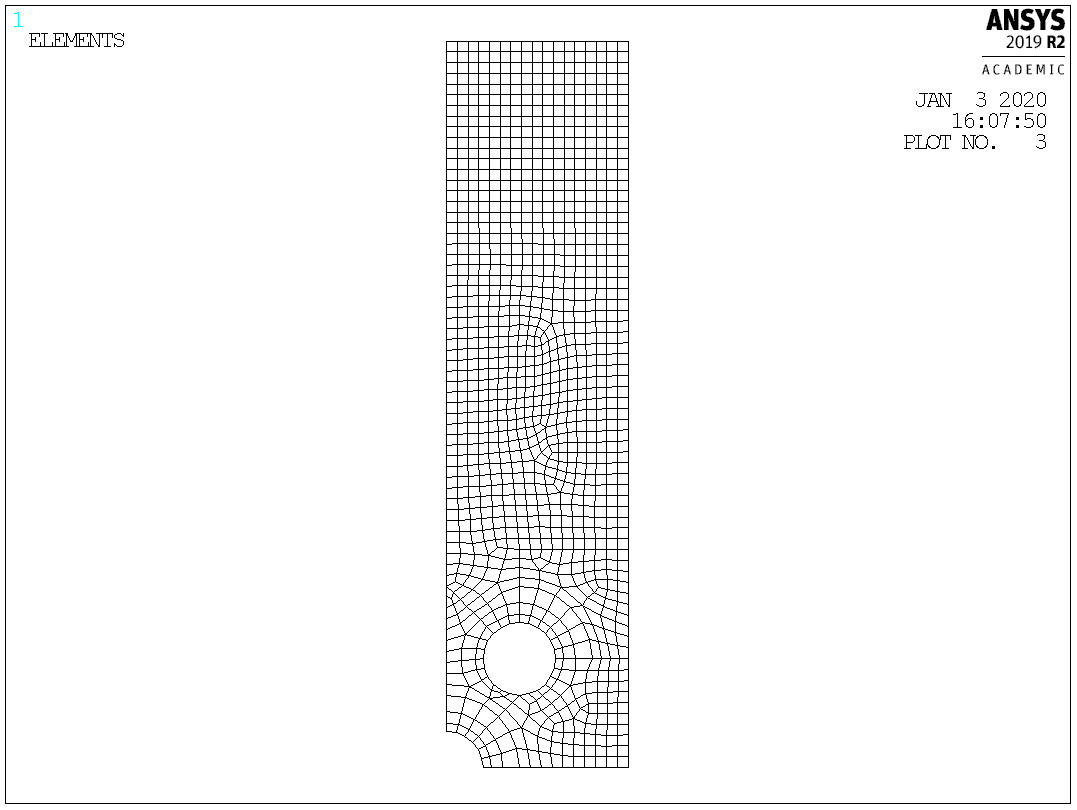
Similar to Table 9 in Section 4.5, Table 14 shows that a sampling size of 440 lines offers a significant improvement over 200 lines while 840 lines offer little improvement.

**Table 14:** Prediction error (%) using different training sampling size for the case of *s/a* = 2, *w/a* = 5

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prediction error (%) under Biaxial Approach\* | | |
|  | = 1 | = 2 | = 3 |
| sampling size of 200 | 31.95 | 16.96 | 5.35 |
| sampling size of 440 | 4.61 | 4.07 | 3.11 |
| sampling size of 840 | 4.55 | 3.97 | 2.98 |

\*Using and of Node 2 as training parameters, Levenberg-Marquart algorithm, pure linear function, maximum absolute error over 300 trials

Table 15 shows that for *s/a* = 2, to achieve similar accuracy, a fine mesh with 6 elements along the perimeter shown in Figure 3b would be necessary. The prediction errors are slightly higher than those for holes without stagger (Table 11) but are still satisfactory. This is due to the extra shear effect introduced by the staggered hole.



|  |  |  |
| --- | --- | --- |
| 1. Coarse mesh with 4 elements along the perimeter (Plane 182) | 1. Coarse mesh with 6 elements along the perimeter (Plane 182) |  |

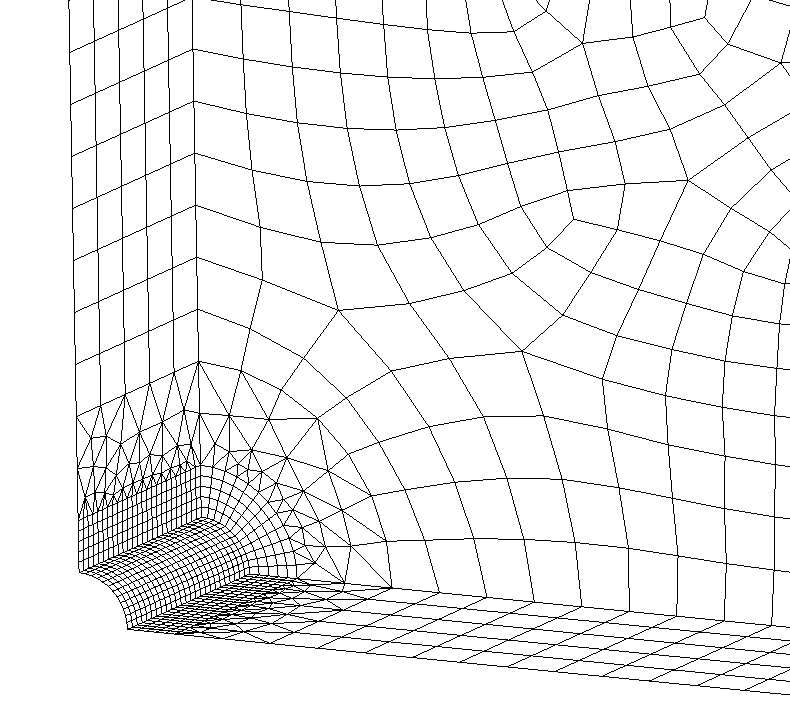
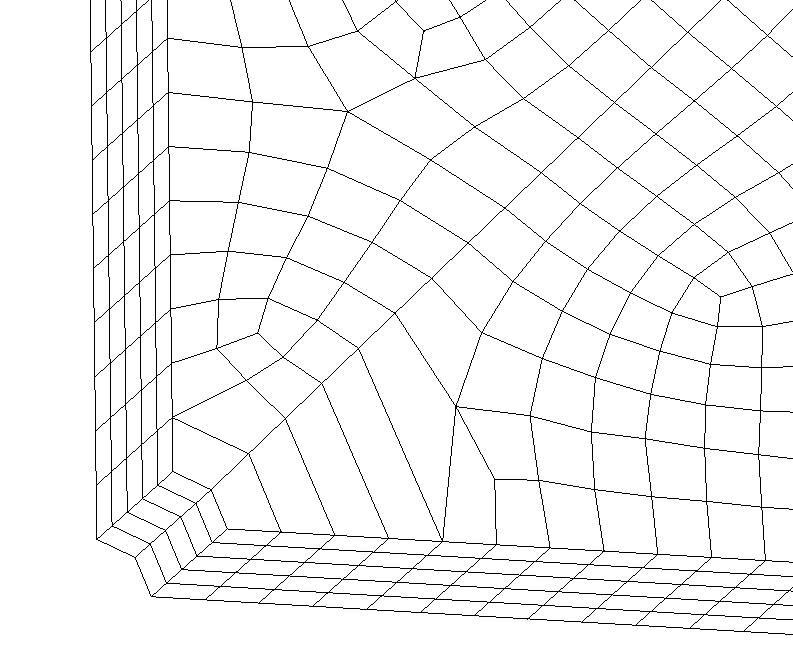
**Figure 13:** Various fine mesh of staggered holes with *s/a* = 2, *w/a* = 5.

**Table 15:** A sensitivity study on the mesh size for *s/a* = 2 where *n* is the number of elements on the hole perimeter. (compared to fine mesh shown in 12b)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *d/a =* 1 | | *d/a =* 2 | | *d/a =* 3 | |
|  | *n* | Kt | Error | Kt | Error | Kt | Error |
| FEM (Quadratic element,  Plane 183) | 16 | 3.465 |  | 2.988 |  | 2.878 |  |
| Biaxial Approach\*  (Linear element,  Plane 182) | 2 | 3.625 | 4.61% | 3.110 | 4.07% | 2.968 | 3.11% |
| FEM (Linear element,  Plane 182) | 2 | 2.773 | -19.98% | 2.386 | -20.14% | 2.289 | -20.47% |
| FEM (Linear element,  Plane 182) | 4 | 3.394 | -2.06% | 2.857 | -4.40% | 2.739 | -4.84% |
| FEM (Linear element,  Plane 182) | 6 | 3.512 | 1.36% | 2.988 | 0.01% | 2.864 | -0.47% |

\* Using and of Node 2 as training parameters, training sampling size of 440, Levenberg-Marquart algorithm, pure linear function, based on maximum absolute error with 300 trials.

# **4.5 3D single hole in the infinite plate using biaxial load trained NN**

|  |  |
| --- | --- |
| 1. Coarse mesh with 2 elements   along the perimeter (Solid 185) | 1. Fine mesh with 12 elements   along the perimeter (Solid 186) |

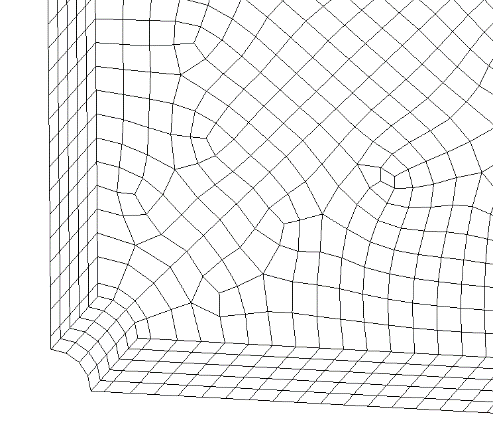
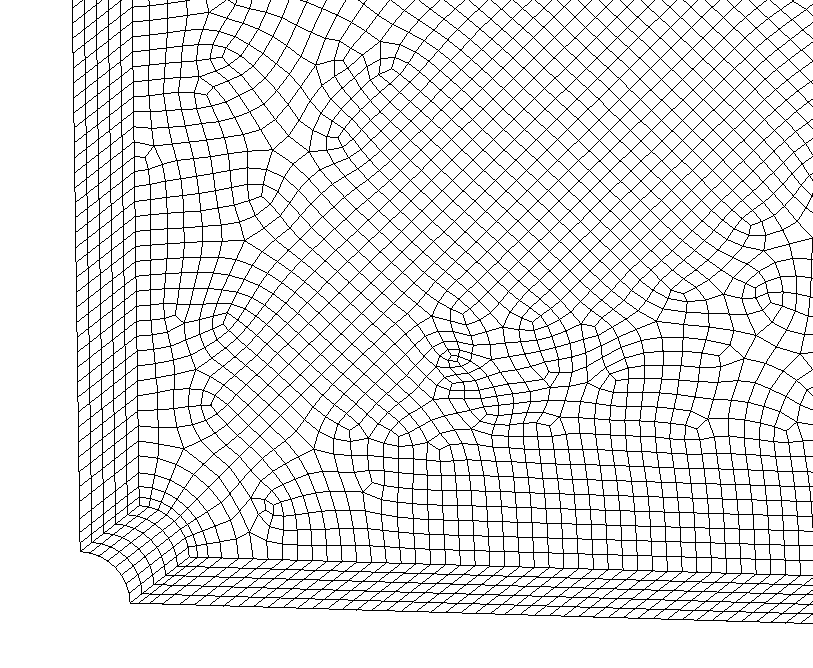
**Figure 14:** Coarse and fine mesh of 3D hole in infinite plate (*w/a =* 20) with thickness, *t* where *t/a =* 10

The 3D mesh model is a quarter-hole in an infinite plate that is bisected in half along the z-axis. The coarse mesh is constructed with linear element Solid 185. Since this is the problem of predicting the stresses of a single hole in an infinite-width plate using a biaxial load trained NN like in Section 4.3, and of all 6 nodes of the coarse mesh will be used as training parameters. The predicted stresses are compared to that of the fine mesh shown in Figure 114b which is constructed with the quadratic element Solid 186 [10]. Table 16 shows the variation of thickness tested.

**Table 16:** Variations in finite multiple-hole mesh

|  |  |  |  |
| --- | --- | --- | --- |
|  | Min | Max | Interval |
|  | 6 | 18 | 4 |

Table 17 shows that to achieve similar accuracy, a much finer mesh with 8 elements along the quarter-hole perimeter is required as shown in Figure 15b. Furthermore, the mesh must be constructed by quadratic element Solid 186.



|  |  |  |
| --- | --- | --- |
| 1. Coarse mesh with 4 elements along the perimeter |  | 1. Coarse mesh with 8 elements along the perimeter |

**Figure 15:** Various 3D fine mesh of single hole in infinite-width plate

**Table 17:** A sensitivity study on the mesh size where *n* is the number of elements on the hole perimeter

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *t/a =* 6 | | *t/a =*10 | | *t/a =* 14 | | *t/a =* 18 | |
|  | *n* | Kt | Error | Kt | Error | Kt | Error | Kt | Error |
| FEM (Quadratic element, Solid 186) | 12 | 3.053 |  | 3.038 |  | 3.033 |  | 3.032 |  |
| Biaxial Approach\*  (Linear element,  Solid 185) | 2 | 3.112 | 1.93% | 3.091 | 1.74% | 3.082 | 1.62% | 3.079 | 1.55% |
| FEM (Linear element,  Solid 185) | 2 | 1.963 | -35.70% | 1.958 | -35.55% | 1.957 | -35.48% | 1.956 | -35.49% |
| FEM (Linear element,  Solid 185) | 4 | 2.586 | -15.30% | 2.573 | -15.31% | 2.569 | -15.30% | 2.568 | -15.30% |
| FEM (Linear element,  Solid 185) | 8 | 2.844 | -6.85% | 2.827 | -6.95% | 2.825 | -6.86% | 2.824 | -6.87% |
| FEM (Quadratic element,  Solid 186) | 4 | 2.95 | -3.37% | 2.935 | -3.39% | 2.93 | -3.40% | 2.93 | -3.36% |
| FEM (Quadratic element,  Solid 186) | 8 | 3.043 | -0.33% | 3.028 | -0.33% | 3.023 | -0.33% | 3.022 | -0.33% |

\* Using and of all 6 Nodes as training parameters, training sampling size of 440, Levenberg-Marquart algorithm, pure linear function, based on maximum absolute error with 300 trials

# **5 Conclusion**

The result shows that with the aid of supervised learning, an error of less than 5% can be obtained even for a coarse mesh with only 2 elements along the whole quarter perimeter. Training is done using the analytical solution for a coarse 2D mesh of a single hole in an infinite-width plate (2D). The prediction is successfully applied to different geometries such as holes in finite-width plates (2D), multiple holes without stagger (2D) and staggered holes (2D). A preliminary study done in this paper also demonstrates that the approach can be applied to 3D models with minor adjustments.

Levenberg-Marquardt method (LM) with pure linear transfer function and Bayesian Regularization (BR) with tangent sigmoid transfer function can achieve low prediction errors for the hole in infinite-width and finite-width problems. However, LM is faster and less calculational complex than BR. Thus, LM is extensively tested in this study. The result shows good prediction accuracy for the different problems.

Nevertheless, various factors can affect the prediction results. Firstly, to train neural networks to be sufficiently sensitive to biaxial loading, the interval of must be sufficiently small. An interval of 0.2 for the range of -1 to 1 is necessary. Secondly, using more training parameters increases the tendency for the NN to overfit the training data set. To reduce the prediction error for other problems such as finite-width (2D) and multiple holes (2D), the number of training parameters is reduced. This is done by using only non-zero displacements of the nodes associated with the element with the highest stress. In this case, 8 non-zero displacements from the 6 nodes are reduced to 4 non-zero displacements. Under these conditions, LM with pure linear transfer function offers maximum absolute prediction errors of 1.52% in the infinite-width problem; 3.55% in the finite-width problem; and 3.27% in the multiple holes (without stagger) problem. In contrast, conventional FEM with a coarse mesh of 2 elements at the quarter-hole perimeter has an error of 26.2%. To achieve similar accuracy by FEM without the aid of supervised learning, at least 6 elements at the quarter-hole perimeter would be necessary.

For staggered holes (2D), the prediction errors are higher due to the influence of shear stress which the neural network is not trained to model. In this case, the interaction of nearby holes may affect the prediction accuracy. Nodes chosen are encouraged to be further away from the other holes to reduce the influence of the stress field of the other adjacent holes. For example, for the staggered holes problem, removing the nodes that are in proximity of the adjacent hole from the training parameters drastically reduced the prediction errors to 4.61%. In contrast, conventional FEM with a coarse mesh of 2 elements at the quarter-hole perimeter has an error of 20.47%. To achieve similar accuracy by FEM without the aid of supervised learning, at least 6 elements at the quarter-hole perimeter would be necessary.

This paper has also demonstrated that the 3D problem consisting of a hole in an infinite-width 3D plate can be accurately predicted using supervised learning. The neural network training is performed using the 2D coarse mesh training set with generalized plane strain condition. When applied to the 3D plate that has a coarse mesh with just 2 elements at the quarter-hole perimeter, the maximum prediction error is 1.93%. In contrast, using the conventional FEM without any supervised training gives an error of 35.7%. To achieve similar accuracy by conventional FEM without supervised learning, more than 8 elements at the quarter-hole perimeter would be necessary.

Hence, it is demonstrated that supervised learning can be used effectively to aid finite element analysis such that even coarse mesh can offer satisfactory accuracy. As a result of using this approach, when performing complex 3-D design iterations with many holes, the need to perform many sub-modeling analyses (for greater accuracy) is significantly reduced.

# **Acknowledgment:** The authors thank the guidance and encouragement from Prof S.N. Atluri over the years.

# **Funding Statement:** The author(s) received no specific funding for this study.

# **Author Contributions:** The authors confirm their contribution to the paper as follows: study conception and design: WT Chow; analysis and interpretation of results: JT Lau, WT Chow. Author; draft manuscript preparation: JT Lau. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Data will be available upon request.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study

**Reference**

1. Moller. M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks,* 6(4). 525-533.
2. Gavin, H. P. (2019). *The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems*, Department of Civil and Environmental Engineering, Duke University, Durham.
3. Shirangi, M. G., Emerick, A. A. (2016). An improved TSVD-based Levenberg–Marquardt algorithm for history matching and comparison with Gauss–Newton. *Journal of Petroleum Science and Engineering*, 143, 258-271.
4. Prerana, P. S., Sehgal, P. (2015). Comparative study of GD, LM and SCG method of neural network for thyroid disease diagnosis. *International Journal of Applied Research*, *1*(10), 34-39.
5. Kayri, M. (2016). Predictive abilities of Bayesian regularization and Levenberg–Marquardt algorithms in artificial neural networks: a comparative empirical study on social data. *Mathematical and Computational Applications*, *21*(2), 20.
6. Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., et al. (2017). In-datacenter performance analysis of a tensor processing unit. *Proceedings of the 44th International Symposium on Computer Architecture (ISCA).* pp. 1-12. Toronto, Canada
7. Hashash, Y. M. A., Jung, S., Ghaboussi, J. (2004). Numerical implementation of a neural network based material model in finite element analysis. *International Journal for numerical methods in engineering*, *59*(7), 989-1005.
8. Manevitz, L., Bitar, A., Givoli, D. (2005). Neural network time series forecasting of finite-element mesh adaptation. *Neurocomputing*, *63*, 447-463.
9. Chow, W. T. (2019). Supervised Learning for Finite Element Analysis of Holes Under Tensile Load. In *International Conference on Computational & Experimental Engineering and Sciences.* pp. 1329-1339. Switzerland: Springer International Publishing.
10. Bishop, C. M. (1995). *Neural networks for pattern recognition*. UK: Oxford university press.
11. Haykin, S. (1998). *Neural networks: a comprehensive foundation*. USA: Prentice Hall PTR.
12. Almuallim, H., Dietterich, T. G. (1991). Learning With Many Irrelevant Features. In *AAAI*. 91, 547-552.
13. Gordon, D. F., Desjardins, M. (1995). Evaluation and selection of biases in machine learning. *Machine learning*, *20*, 5-22.
14. Tan, H. H., Lim, K. H. (2019). Vanishing gradient mitigation with deep learning neural network optimization. *2019 7th international conference on smart computing & communications (ICSCC).* pp. 1-4. Sarawak, Malaysia.
15. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107-116. \
16. Nguyen, D., Widrow, B. (1990). Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. *1990 IJCNN international joint conference on neural networks*, pp. 21-26. San Diego, USA.
17. Matlab. (2018). *Nguyen-Widrow layer intialization function.* USA: MathWorks.
18. Pavelka, A., Procházka, A. (2004). Algorithms for initialization of neural network weights. *In Proceedings of the 12th annual conference, MATLAB,* pp. 453-459.
19. Wayahdi, M. R., Zarlis, M., Putra, P. H. (2019). Initialization of the Nguyen-widrow and Kohonen algorithm on the backpropagation method in the classifying process of temperature data in Medan. *Journal of Physics: Conference Series,* 1235(1), 012031.
20. Lundén, J., Koivunen, V. (2007). Scaled conjugate gradient method for radar pulse modulation estimation. *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07.* pp. II-297. Honolulu, USA.
21. Lourakis, M. I. (2005). A brief description of the Levenberg-Marquardt algorithm implemented by levmar. *Foundation of Research and Technology*, *4*(1), 1-6.
22. Madsen, K., Nielsen, H. B., Tingleff, O. (2004). *Methods for non-linear least squares problems.* Denmark: Technical University of Denmark.
23. Nielsen, H. B. (1999). *Damping parameter in Marquardt's method.* Denmark: Technical University of Denmark.
24. Levenberg, K. (1944). Method for the solution of certain problems in least squares siam. *Quarterly of Applied Mathematics*, 2, 164-168.
25. Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, *11*(2), 431-441.
26. MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, *4*(3), 415-447.
27. Foresee, F. D., Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. *Proceedings of international conference on neural networks (ICNN'97),* pp. 1930-1935. Houston, USA.
28. Pilkey, W. D., Pilkey, D. F., Bi, Z. (2020). *Peterson's stress concentration factors*. USA: John Wiley & Sons.
29. Ansys. (2018). *Mechanical APDL Theory Reference*, USA: Ansys Inc.
30. Matlab. (2018). *Statistics and Machine Learning Toolbox:User's Guide*. USA: MathWorks.
31. Young, W. C., Budynas, R. G., Sadegh, A. M. (2012). *Roark's formulas for stress and strain*. USA: McGraw-Hill Education.