UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Università degli Studi di Padova

DEPARTMENT OF INDUSTRIAL ENGINEERING - DII

MSc. Degree in Aerospace Engineering

Structural analysis using a Neural Network approach of a cylindrical shell subjected to a pressure load

in collaboration with

RWTH Rheinisch-Westfälische Technische Hochschule Aachen
SLA Institute of Structural Mechanics and Lightweight Design

Student

Enrico Cagnato

Mat: **2015545**

Advisor:

Prof. Mirco Zaccariotto

Co-Advisors:

Univ.-Prof. Dr.-Ing. Kai-Uwe Schröder
Dipl.-Ing. Thomas A. Schervan

ACADEMIC YEAR 2022/2023

Thesis defended on 11 July 2023
in front of a Board of Examiners composed by:

Dott. Stefano Casotto
Dott. Giacomo Colombatti
Ing. Sebastiano Chiodini
Ing. Andrea Valmorbida
Prof. Mirco Zaccariotto (chairman)

Structural analysis using a Neural Network approach of a cylindrical shell subjected to a pressure load
Master Thesis. Università degli Studi di Padova

© 2023 Enrico Cagnato. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: enrico.cagnato@gmail.com

Dedicato a
tutti coloro che lavorano per lasciare questo mondo migliore di come l'hanno trovato,
e alla mia famiglia.

Dedicated to
all those who work to leave this world better than they found it,
and to my family.

Abstract

This thesis work is going to study how the use of artificial intelligence, and more specifically machine learning applied through the implementation of neural networks, can be applied to structural mechanics analyses carried out on different types of cantilever beam and shell structures. The use of this type of new technology to exploit artificial intelligence makes it possible to greatly reduce the calculation time required for the conventional analysis of these structures that would be required with finite element analysis, while still managing to maintain a high level of reliability in terms of predicting results such as stress and displacement of structures in response to applied loads. In order to be able to effectively apply machine learning with the implementation of a neural network for the study of a shell-type structure, the transfer matrix method is used, thanks also to the exploitation of representations of the applied load by means of Fourier series.

One would like to thank the RWTH Aachen Polytechnic University for the opportunity provided, the German thesis advisor Dipl.-Ing. Thomas A. Schervan for his guidance, collaboration and revision during the thesis work carried out in Germany, the Head of the SLA RWTH Univ.-Institute of Structural Mechanics and Lightweight Aerospace Structures, Prof. Dr.-Ing. Kai-Uwe Schröder, and for the collaboration and final revision of the thesis work the Italian thesis advisor Prof. Mirco Zaccariotto.

Italian version:

Questo lavoro di tesi va a studiare come l'utilizzo di intelligenza artificiale, e più specificatamente del machine learning, sia applicato mediante l'implementazione di reti neurali e possa essere applicato alle analisi di meccanica strutturale effettuate su diversi tipi di struttura, in particolar modo una cantilever beam e una struttura di tipo shell. L'utilizzo di questo tipo di nuove tecnologie, per sfruttare l'intelligenza artificiale, consente di andare a ridurre in modo molto importante i tempi di calcolo necessari per l'analisi convenzionale di queste strutture che si avrebbero con analisi agli elementi finiti, pur riuscendo a mantenere un livello di alta affidabilità in termini di predizione dei risultati quali stress e displacement delle strutture in risposta a carichi applicati. Per poter applicare in modo efficace il machine learning con l'implementazione di una rete neurale per lo studio di una struttura di tipo shell, viene utilizzato il transfer matrix method, grazie anche all'impiego di rappresentazioni del carico applicato mediante serie di Fourier.

Si ringrazia l'università politecnica di Aquisgrana RWTH per l'opportunità fornita, il relatore tedesco Dipl.-Ing. Thomas A. Schervan per la guida, la collaborazione e la revisione durante il lavoro di tesi svolto in Germania, il Direttore dell'istituto di meccanica strutturale e strutture leggere aerospaziali dell'università RWTH Univ.-Prof. Dr.-Ing. Kai-Uwe Schröder, e per la collaborazione e la revisione finale del lavoro di tesi il relatore italiano Prof. Mirco Zaccariotto.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this first chapter, there is a brief introduction to the background of the thesis project. This thesis was carried out during an Erasmus international mobility period at the RWTH Aachen Polytechnic University, in collaboration with the Institute for Structural Mechanics and Design of Lightweight Aerospace Structures, under the guidance of Univ.-Prof. Dr.-Ing. Kai-Uwe Schröder, with German co-advisor Dipl.-Ing. Thomas A. Schervan and Italian advisor Prof. Mirco Zaccariotto.

1.1 Objective

Structural calculations of large structures made by means of F.E.M. finite element analysis, especially considering shell-type structures, usually require very considerable computational time as the complexity of the structure itself increases. In everyday life, the space sector uses many structures of this type, just think of the propellant tanks for launchers or vehicles suitable for space transport, the fuselage of an aircraft, or the modules of the International Space Station and the cargo vehicles that supply it with raw materials essential to the lives of the astronauts and the experiments they conduct.

It is therefore necessary to reduce the calculation time for structural analyses of this type, while maintaining a good reliability of the results, in order to reduce the design time and costs of these structures used in aerospace and increasingly also in industry. To this end, therefore, neural networks were implemented to predict the behaviour of a cantilever beam and a shell-type structure under applied loads, respectively, to be included in a larger context of a prediction tool consisting of neural networks to be used for structural calculations within the institute for structural mechanics and design of lightweight aerospace structures at the RWTH Polytechnic University.

1.2 State of the art

With regard to the state of the art in the use of artificial intelligence in structural analyses, it is very common to use neural networks of a convolutional type that analyse images, pictures or videos of structures in search of possible damage or cracks. For this reason, I chose to follow the path of constructing neural networks

of a different type, pertaining to the field of deep learning, then using them in the prediction of stress and displacements of the structures analysed.

1.3 Overview

In the development of the work of this thesis, therefore, one will proceed with an introduction and explanation regarding what machine learning is and how it is used in modern engineering. Next, the beam theory for bending and normal force behaviour is explained, as this theory forms the basis of the cantilever beam study. One proceeds with the description of the implementation and the results obtained regarding the study of a cantilever beam clamped constrained is subjected to a concentrated force acting on it that leads it to bending. This first part is necessary to introduce the study of neural networks in structural mechanics. In the second part of this thesis, one continues with the description and explanation of the transfer matrix method and the behaviour of shell-type structures. This method will in fact be the one that allows the implementation of the neural network architecture used for the study of a shell-type structure with a cylindrical shape, of very wide use as previously explained in aerospace, subjected to an internal pressure load that is described using a Fourier series. A possible analysis of the inverse problem of the shell is presented. One will conclude with an analysis of the results and possible future developments.

Chapter 2

Machine Learning in structural mechanics

In this chapter, one presents how machine learning is used in engineering today, describing the functioning of the main types of artificial intelligence used, starting from the origins of the introduction of this new technology to its most recent uses.

2.1 Machine Learning basics and origins

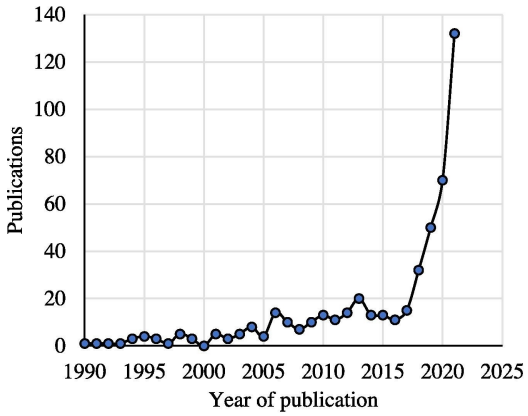
Nowadays, the artificial intelligence is really important and widespread for a lot of applications, regarding engineering, medicine, biology and a lot of other scientific areas. Inside Artificial Intelligence, commonly known as AI, there a branch that is gaining success and importance: this is Machine Learning. Machine Learning, as it is easy understandable from the name itself, is a special technique that belongs, as previously said, to artificial intelligence, that consist in give to the computer itself the capability of understand hidden relations behind a provided problem, link together data provided to the machine, and then improve its knowledge without the need of having a person physically programming the computer, giving predictions using algorithms and available data-set. Machine Learning is often used in self-driving cars, image and details of images and videos recognition, speech recognition and production, alerts for cars and traffic using the navigator, medical diagnosis, security for electronic devices, and it is used also in sports with technologies such semi-automatic offside and the goal line technology used during soccer matches etc.

To find the origin of Machine Learning as a theoretical concept, it is necessary to look in the past till the year 1943 for the conception, and to 1959 to find the original coining of the word; despite that, the very first use, especially in engineering design, and precisely also for structural mechanics, has to be referred to 1989, with the work of Adeli and Yeh, that studied how to teach a computer how to improve itself in the study of structures for civil engineering, using a steel beam design problem. [1]

Looking closer to the Machine Learning, there three main classes can be spot inside, that can be summarized as: unsupervised learning, supervised learning, and reinforcement learning. The first class of learning process is based on training an algorithm using an unlabelled data-set; for the supervised learning process instead, the learning process of the algorithm is done by using a labelled data-set, making this

method perfect for classification problems and regression problems, aiming to link a set of features given as input to an output (one or a set); the third class of learning process is the reinforcement learning method, in which the algorithm is trained using a try and error process. The second class, the supervised learning process, is of particular interest in structural engineering, and giving some examples, a typical classification problem that uses this method is the damage and cracks detection, then, a typical regression problem can be the prediction of strengths and stresses in a structure. Structural engineering is using more and more Machine Learning, differentiating using different methods for every purpose that has to be fulfilled in order to study a lot of applications. The most known are for sure artificial neural network (ANN), support vector machine (SVM), random forest (RF), boosting algorithm (BA), decision tree (DT), regression analyses (RA), and many others.

Machine learning used in structural engineering can be useful in saving time that normally in a structural analysis and using a conventional design process can be fundamental, giving affordable results in a reasonable time, allowing to modify then the initial input also adding some little modifications. This fact led to a first use of Machine Learning in past, around the '90, when the computation capability of computer machines was absolutely not comparable with the one that every computer, also the weakest one, has nowadays available. In this field, one of the most used method has become the implementation of neural networks, of which many past references can be found starting from 1989[1] as previously said, and from 1991 with another interesting work conducted by Hajela and Berke [12]. The fact that in the first period machine learning is still at its very first phase can be proved also counting the number of publications of those years, that was not high, due to computational limits but also due to simplicity of the first algorithms implemented, as clearly visible in Fig. 2.1 The number of Machine Learning methods used at the state of the art is big, and many of them can be applied also to structural mechanics, and this is the topic of the next section.



(a)

Figure 2.1. Yearly distribution of articles related to ML applications in structural engineering considering all the available algorithms. [27]

2.2 Machine Learning algorithms

A lot of algorithms and new technologies arise every year using artificial intelligence with all its sub-classes and techniques (see the composition in Fig. 2.3); here are presented the most widespread methods to use machine learning and its theoretical peculiarities in structural design. In recent time, the use of Python and some common python libraries, open-source developments and already existing data-sets has increased the use of machine learning also for common users, without specific knowledge in computer science. The performances of the learning method and the initial data-set provided to the algorithms are important evaluating the precision and the accuracy of the machine learning model. The sequence then in building machine learning models can be divided in three processes: creating databases, learning, and evaluating models.

Initial data are split in two big data-sets: the training data-set, that contains the vast majority of data, use to train the algorithm, and the testing data-set, whose values are not submitted to the machine learning method during the train, but just after it, resulting than as completely unknown for artificial intelligence. Sometimes, to improve the results of some algorithms, it is very important to scale input data in a range that goes from 0 to 1, normalizing them. [5] The steps for the creation of the training data-set are summarized in Fig. 2.2

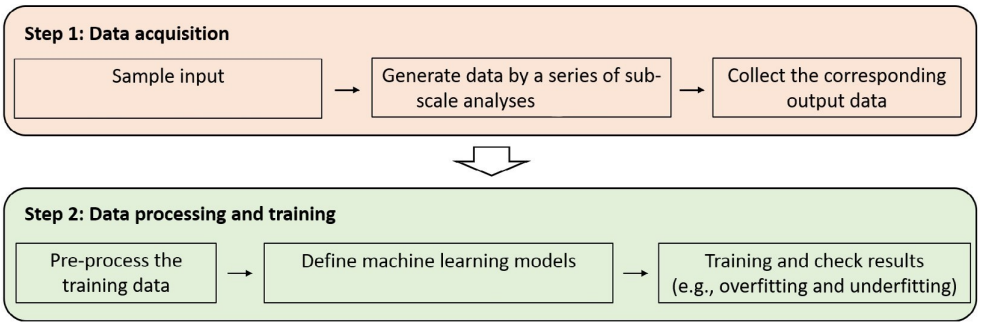


Figure 2.2. Steps for the creation of the training data-set. Image source: [14]

Then, comparing all the available algorithms is important to choose the one that best suits the needs of the studied problem. Once chosen, the training of the algorithm starts using the training data-set. After the training, the testing data-set is used to test the performance and the predictions that the algorithm is now capable to provide. As performance indicator, a loss function is used to measure the real difference between the predicted value of a variable and its true value. Select the correct loss function has consequences in the evaluation of performance; the most used and representative loss functions for regression problems are the Root Mean Square Error (MSE) and Mean Absolute Error (MAE).

Mean squared error (MSE) is a loss function used to solve regression problems, calculated as the average of the squared differences between the actual and predicted values:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.1}$$

where:

- n - the number of data points.
- y - the actual value of the data point. Also known as true value.
- $\hat{y}$ - the predicted value of the data point. This value is returned by model.

Mean absolute error (MAE) is a loss function as well used to solve regression problems, calculated as the average of the absolute differences between the actual and predicted values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.2}$$

where:

- n - the number of data points.
- y - the actual value of the data point. Also known as true value.
- $\hat{y}$ - the predicted value of the data point. This value is returned by model.

The commonly used machine learning algorithms for structural engineering are designed for specific purposes, presenting different strength points and weaknesses. An idea of the majority of the nowadays used algorithms can be seen in Fig. 2.4

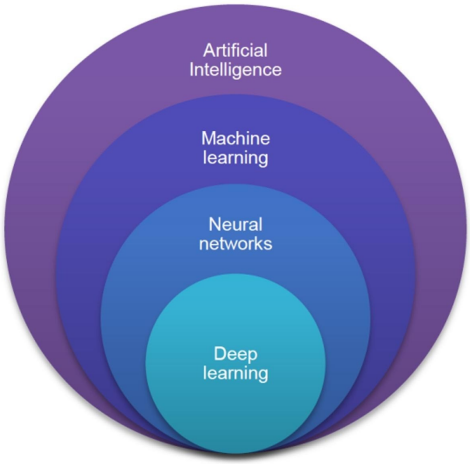


Figure 2.3. Schematic view of the broad range area of the artificial intelligence. Image source: [25]

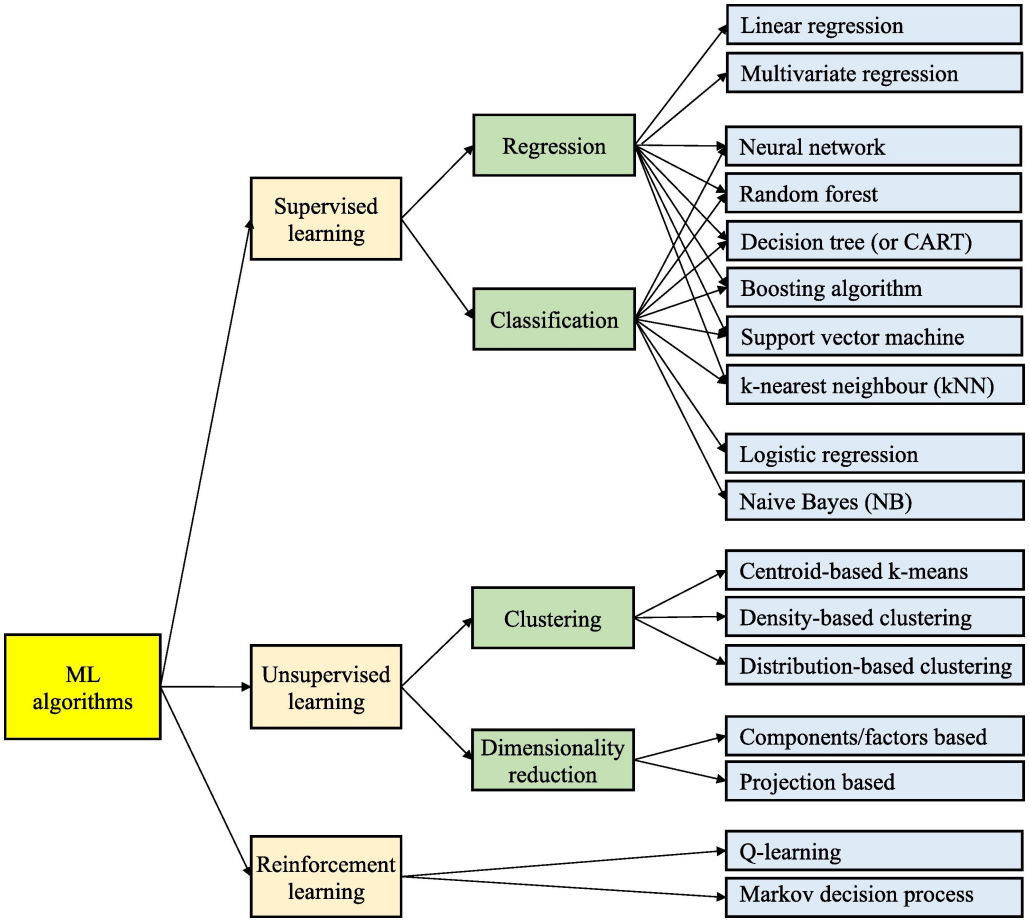


Figure 2.4. Machine Learning algorithms grouped by learning type. Image source: [27]

Regression analysis is a predictive modelling technique based on statistics that allow to study the relationship between an independent variable (predictor) and a dependent variable (target). This method is applied in the supervised learning algorithm, to predict, based on the values of the input variables, the output values. Many regression models have been developed, differentiating one to each other in the number and/or type of variables, and the shape of the regression lines used. The most used regression analysis methods for structural mechanics are:[27]

- Linear Regression: the simplest regression model, in which a linear function best fits output and input variables. Minimizing the loss function, such as MSE and MAE previously defined, the coefficients of the linear equation are determined. With only one input variable, the model is called simple linear regression. With multiple input variables, it is called multiple linear regression;

- Multivariate regression: it is an extension of multiple linear regression when dealing with problems with multiple output variables. "Multivariate" refers to multiple output variables, while "multi" refers to multiple input variables. The advantage of this approach is that it helps to understand the relationship between input and output variables;

- Polynomial regression: the difference between this model and the linear regression model is the shape of the regression line. The line of best fit in a polynomial is a curve for which the power of the input variables is greater than one (a polynomial function). Overflow can occur in the model if the input variables are fitted using higher order polynomials to obtain a smaller error. Therefore, it is useful to plot the model to ensure reasonable results;

- Lasso regression: Least absolute shrinkage and selection operator (LASSO) regression is a common version of linear regression and is used when the input variables are highly correlated. In this case, using linear regression techniques can lead to over-fitting. Therefore, this model is proposed to reduce over-fitting by adding a regularization term to the cost function during training. This keeps the weight of the model as low as possible. The regularization term used in this model is the absolute value of the weights; [8]

- Ridge regression: Similar to lasso regression, this form is also a normal version of linear regression. However, the regularization term used here is the squared value of the weights, not the one used in lasso regression. The goal of this modelling technique is to try to remove weight from less important features; [8]

- Logistic regression: This regression technique was developed for classification problems when the output variable is binary or discrete in nature (e.g. true/false, 1/0, yes/no, etc.). In this model, the relationship between input and output variables is represented by a logistic function (also known as a sigmoid function). The method is based on the concept of probability. This means that the model outputs a binary value of 0 (when the estimated probability is less than 50%) or 1 (when the estimated probability is greater than 50%) instead of a numeric value.

Neural networks are one of the most used machine learning methods, especially in structure engineering problems, where they are also called Artificial Neural Network. This model works exactly as a biological neuron in the human brain, doing the same job. The first artificial neural network was created in 1958, by RosenBlatt, who created a model called Perceptron, using the capable of performing pattern recognition tasks after training.[21] It can be defined as a function

$$\phi(\vec{X}) = y \tag{2.3}$$

that calculates

$$b + \sum x_i w_i \tag{2.4}$$

where w_i are the weights to learn and x_i are the input elements of $\vec{X}$.

The output of a neuron is the result of its activation function ϕ, which is a Heaviside step function (threshold function) that returns 1 if $y > 1$ and 0 otherwise. During the training process each weight is adjusted, in a supervised manner, with the desired output known by the training data-set. Given the training sample, the new weights are modified by

$$w_i^{t+1} = w_i^t + \alpha(\bar{y} - y)x_i \tag{2.5}$$

with a learning rate α that controls the size of the update, and $\bar{y}$ is the expected output for the input vector $\vec{X}$. Iterating this process for a given number of e epoch for the training data-set, Perceptron tends to find solutions to linearly separable binary problems. [22]

Thanks to the continue improving of computers, ANN is the most used machine learning algorithm today, including some variants such as:

- feed-forward neural network, the first and simplest type, that transmits information only form input node to output node, in a forward direction;
- improved by multi-layer perceptron, an improved version of feed-forward neural network that uses also one or more hidden layer between input and output;
- radial basis function neural network, with a specific activation function;
- convolutional neural network (CNN), specifically developed for image recognition using deep learning, implemented especially in image clustering, division, and recognition;
- recurrent neural network improved by long short term memory, designed to interpret temporal or serial information;
- adaptive neuro-fuzzy inference system, combines the learning capabilities of the adaptive neural network with the inference of the fuzzy inference system;

- physical neural network (PINN), in which a physical process is trained rather than a mathematical process, training directly the physical transformation of the machine to carry out the required calculations.

The convolutional neural network is a method very used in structural mechanics for crack detection based on images, detecting damages on structure very quickly. Each of the layers of a convolutional neural network can convert the information from the previous layer into more complex information, passing them to the next layers. Connected to it, a team at Google achieved another impressive algorithm by creating a new ML method called AutoML-Zero that scales without human intervention. .[19]

2.2.1 Artificial Neural Networks

The Artificial neural networks are made up of artificial neurons, structures like the human brain, called nodes. These nodes are arranged in three layers, and they are connected one to each other (see Fig. 2.5 (a)). The input layer received the data x, and then the predictions of the network are then presented in y in the output layer (see Fig. 2.5 (b)). More hidden layers consisting of data processing units are present between the input and output layers: they can be one or more, and they do the fundamental computational operations on the input data. When an Artificial neural network is designed with two or more hidden layers, it is called Deep Learning and is a specific sub-field of Neural network-based machine learning algorithms.[23]

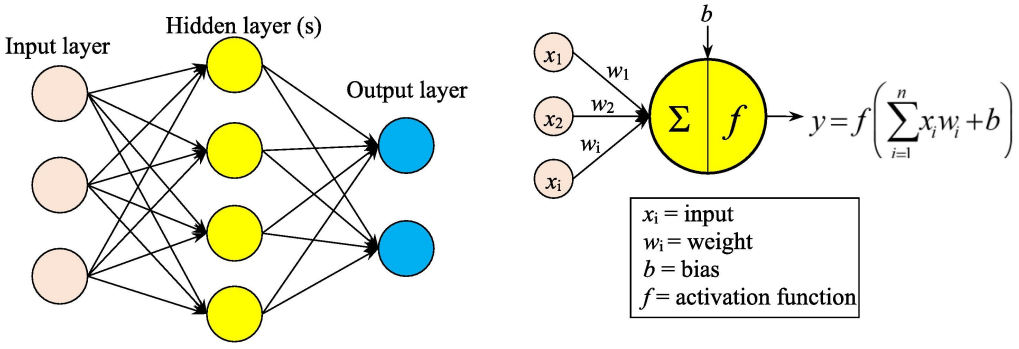


Figure 2.5. Example of an ANN. Image source: [27]

To every neuron a weight w is assigned controlling how it behaves, so that the corresponding weight w_i is multiplied by the data x_i that passes to the input layer. Then, a transfer function is used to calculate the weighted sum of the inputs, to which the bias b is added to reduce at minimum the difference between the desired value and the one obtained from the algorithm, by adjusting incrementally this bias parameter b. To control if the node has to send data to the output layer or not, the value of the transfer function is passed through the activation function f. First,

training an artificial neural network, some random weights values are assigned to the nodes, and then they are adjusted thanks to the use of an error function, that check the difference of predicted and desired value as soon as the activation function passes to the output layer the predicted value y in order to minimize the difference just described: this is the back-propagation process. To end this process of training, the developer set a convergence condition. A typical scheme of artificial neural network flow is presented in Fig. 2.6. To control and optimize the learning process and the training process, there are some so called hyper parameters that have to be chosen when the model of the neural network is defined. These hyper parameters are correlated to the structure and architecture of the neural network itself, and are for example the number of hidden layers used, the number of nodes per hidden layers, and the type of activation functions, and correlated to the training variables for example the leaning rate, the number of epochs (iterations that the network will do to try to converge), momentum, and batch size.

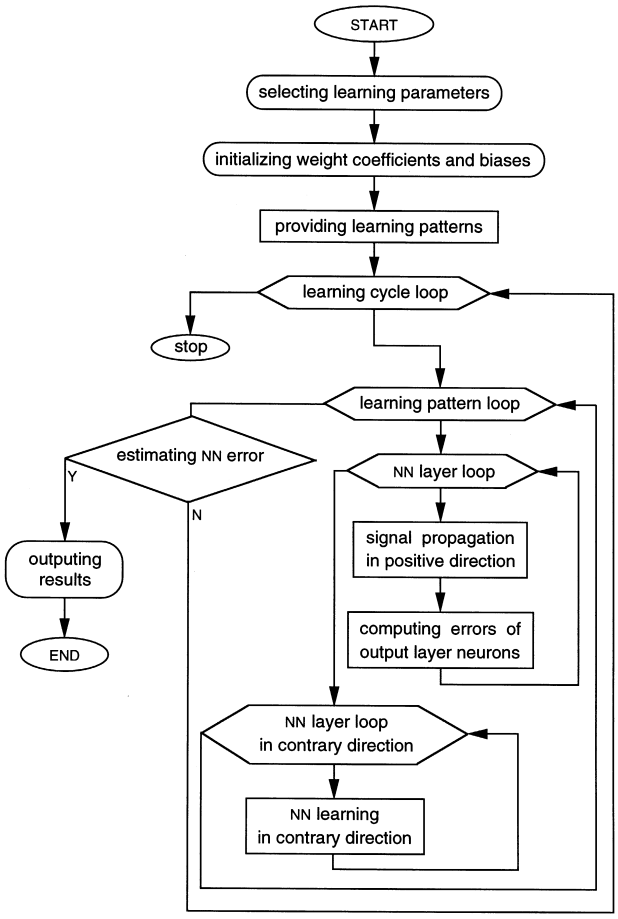


Figure 2.6. Some frequently used activation functions. Image source: [11]

Increasing the number of hidden layers and hidden nodes can increase for sure the precision and accuracy of the prediction, but the risk is to make the computational

time explode. The activation function is used to account for the non-linearity of the models themselves. Some examples of activation functions can be seen in the Fig. 2.7.

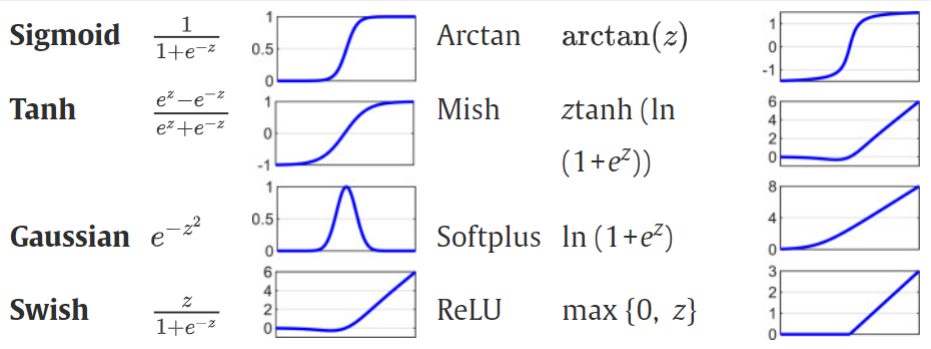


Figure 2.7. Some frequently used activation functions. Image source: [11]

The learning rate determines the velocity with the artificial neural network can update its parameters. Using a higher learning rate will speed up learning, but it may not converge. Epoch controls the number of iterations of the training data set, while using momentum to choose the direction of the next move from the previous one, which helps prevent spurious oscillations.[27]

2.2.2 Python libraries for Machine Learning

Nowadays neural networks are very accessible and easy to use also for people not so expert in programming also thanks to cloud platforms and services provided by companies like Google, Facebook, Microsoft, creating many models. A very common programming language for machine learning is Python, that is really used thanks to the big amount of libraries and algorithms already implemented present in the web. This is one of the reasons why Python has been chosen also as the language used for this thesis work. The development of many open source libraries and ready to use applications (for example TensorFlow and Keras, developed by Google, and PyTorch, developed by Facebook), has resulted in vast use of Python for machine learning also in the field of structural engineering.
TensorFlow was first developed by google in 2015, and then also a second improved version is available since 2017. It is an open source library, currently considered the best platform for developing machine learning models, especially deep neural networks, that integrates several algorithms.
Keras is also an open source library for developing and evaluating deep neural networks as TensorFlow, and was developed by Chollet [46] in 2015 as an interface to the TensorFlow library. [5] Keras contains many functions for creating NN networks and many tools for working with images and text data, and it can run on TensorFlow and Theano platforms.

PyTorch was first released by Facebook in 2016. It is very popular in deep learning use as Keras and TensorFlow.[8] It has a powerful framework for creating computational graphics and provides many tools and libraries for machine learning.
Scikit-Learn is a library created in 2007 based on both Python and the NumPy and SciPy scientific digital libraries. Scikit-Learn provides a wide range of functions for classification, regression, clustering and dimension reduction algorithms. It is popular especially for data mining and analysis of data.
Pandas is the most popular Python library for data analysis in machine learning, developed in 2008. It contains advanced data structures and can handle various data types, including raster, table, and time series data. It is suitable for large datasets handling.
Spark MLlib library has been developed to make easier the big data processing.
Theano is a platform developed by the University of Montreal as Symbolic Mathematics Assistant. It is a powerful library of scientific calculations similar to Spark MLlib, designed to quickly compute many large-scale computation-intensive problems. Many deep learning libraries, as Keras, are built on it.
NumPy is a very useful general-purpose library for manipulating arrays, also in multiple dimensions (as matrices). NumPy stands for "Numerical Python", which builds on top of the old "Numeric" library. It has become one of the core and popular libraries for machine learning because it contains many mathematical functions for working with large multidimensional matrices and arrays. It is especially useful in basic algebra, stochastic simulations, and Fourier transforms.
SciPy stands for Scientific Python, and is a library built on top of the NumPy library, used for scientific computing. It has many optimization, linear algebra, fast Fourier transforms, and signal processing mathematical functions. It is fast in computing power and scientific functions implementations.
Matplotlib is a complete plotting library for Python visualization, this as well is built on top of NumPy. Matplotlib helps to understand common patterns and distributions of data, providing a wide range of plotting functions. [27]

Even if Machine Learning is so widespread and used also in engineering field, there still some not negligible challenges and limitations that still haven't been studied. A summary of all the challenges, encountered also during the development of the neural networks used in this thesis, can be seen in Fig. 2.8 It is interesting also to take into account the limitations of Machine Learning, in order to understand that this method of solving engineering problems has still a lot of potential for the future, so it can become even more powerful in the near future. A summary of machine learning limitations can be seen in Fig. 2.9.

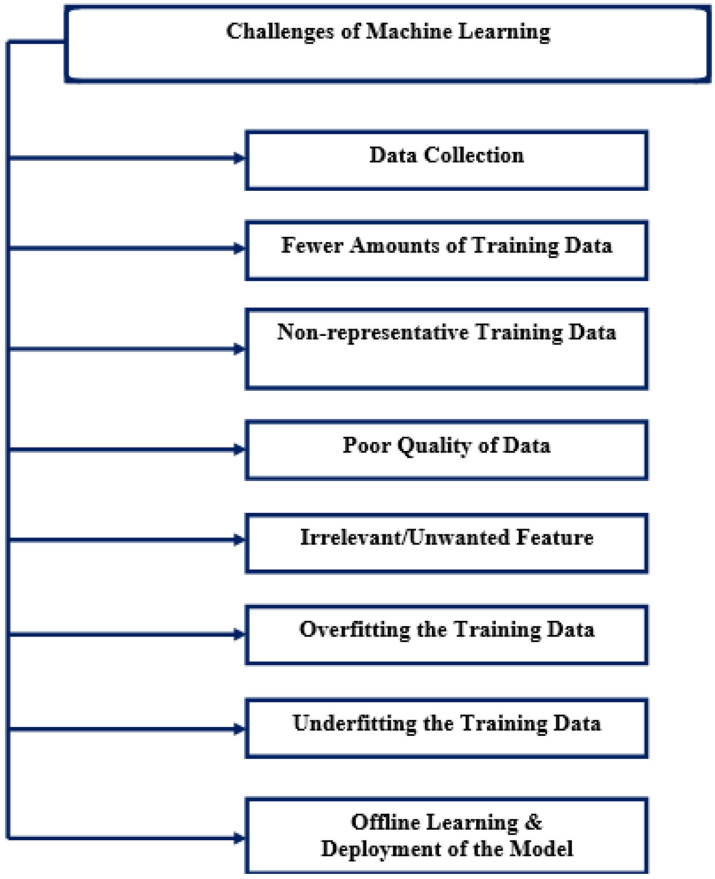


Figure 2.8. Challenges of machine learning. Image source: [17]

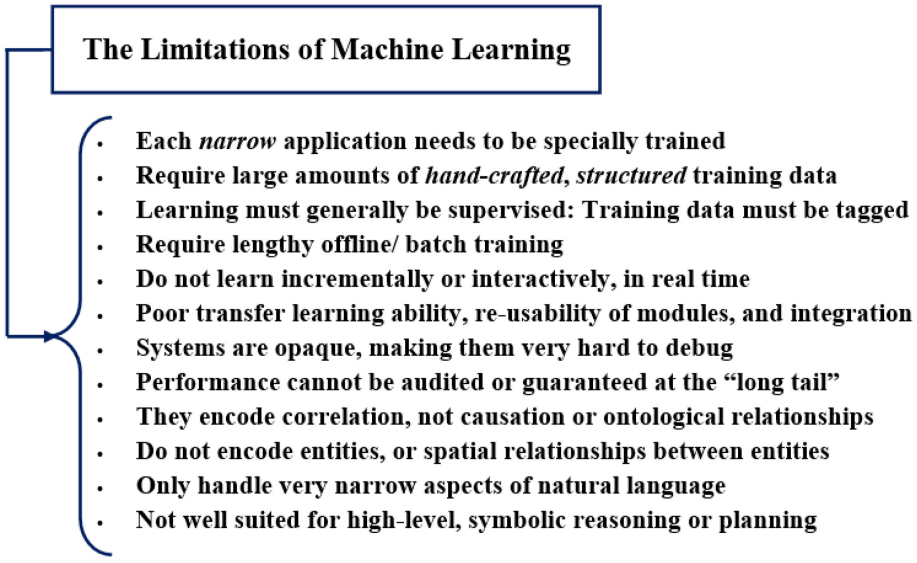


Figure 2.9. The limitations of machine learning. Image source: [17]

Chapter 3

Beam theory to study a cantilever clamped beam

This chapter deals with the theory of the beam, which is necessary to then develop the study of stresses and displacements using the neural network that is subsequently implemented. In particular, the main focus is on normal force loads, simple beam bending and semi-inverse hypothesis.

The first study approach of this thesis involves the analysis of a cantilever beam, with the aim of understanding and studying the operation of a simple neural network in the analysis of a structural analysis problem of a linear type. The analytical study of this problem is done using the *Elastic line* theory. In order to better understand and apply the elastic line theory, one proceeds with a brief recapitulation of the theory behind the analytical study of beams. The beam is a solid generated by a plane figure S characterised by variations in shape and size that are always modest, also known as the orthogonal section or cross-section, which generates the beam moving in the surrounding space while remaining perpendicular to the trajectory s described by its barycentre G. The length of the beam itself must be at least one order of magnitude grater than the dimensions of its section, allowing to consider the beam as a one-dimensional object.[13]

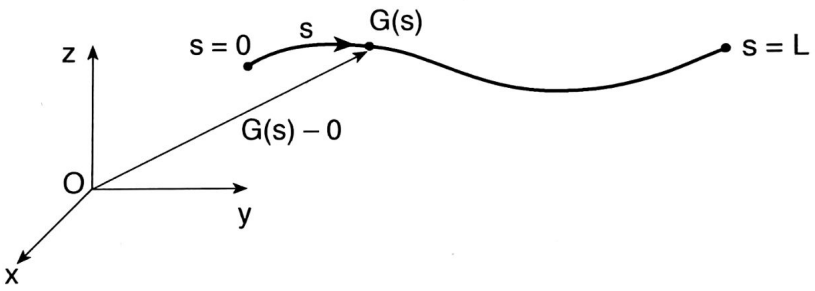


Figure 3.1. Beam represented with the trajectory s. Image source: [13]

.

Since the beam is therefore by definition a one-dimensional object, it is useful to describe its basic characteristics by referring only to its axis, thinking of it as a fairly regular curve in space defined $s \to [G(s) - O]$. The parameterisation employed then measures the length of the curve from a reference point. By calculating the derivative of the position vector $[G(s) - O]$ with respect to the trajectory s, it obtains the tangent versor $\mathbf{t}$ to the axis of the beam:

$$\mathbf{t}(s) = \frac{d[G(s) - O]}{ds} = [G(s) - O]' \tag{3.1}$$

The demonstration that $\mathbf{t}$ in this case represents a versor is directly derived from the fact that s is the abscissa measuring the length of the curve and therefore holds:

$$\begin{aligned} |\mathbf{t}(s)| &= 1 \\ \mathbf{t}(s) \times \mathbf{t}(s) &= 1 \end{aligned} \tag{3.2}$$

Then, deriving this last equation, it is possible to get to:

$$\mathbf{t}'(s) \cdot \mathbf{t}(s) + \mathbf{t}(s) \cdot \mathbf{t}'(s) = 2\ \mathbf{t}'(s) \cdot \mathbf{t}(s) = 0 \tag{3.3}$$

for which $\mathbf{t}'$ is orthogonal to $\mathbf{t}$.

It is then possible to define the curvature of the beam $\kappa(s)$ as the modulus of $\mathbf{t}'$, the radius of curvature $\rho(s) = 1/\kappa(s)$ as the inverse of the curvature and the principal normal $\mathbf{n}$ as the versor of $\mathbf{t}'$. Thus, for the Eq. 3.3, $\mathbf{n}$ is perpendicular to $\mathbf{t}$. The radius of curvature ρ is the circumference radius that better approximates locally the beam axis; $\mathbf{n}$ point toward the centre of this circumference. The curvature is the index that measures how much the behaviour is far from the straightness: for straight beams $\mathbf{t}(s) = constant$, and $\kappa = 0$. For straight beams, the z-axis is chosen coincident with the beam axis, for which $s = z$.

3.1 Beam theory in the plane

A beam in the plane is represented by the equation $y = y(x)$.

The differential quantity ds can be expressed with the equation:

$$ds = \sqrt{(dx)^2 + (dy)^2} \tag{3.4}$$

dividing it for the differential quantity dx, it can be written as:

$$\frac{\mathrm{ds}}{\mathrm{d}x} = \sqrt{1 + y'(x)^2} \tag{3.5}$$

Working with these quantities, the versor $\mathbf{t}$ tangential to the trajectory and its first derivative are expressed as function of both $x(s)$ and $y(s)$ as in Eq. 3.6:

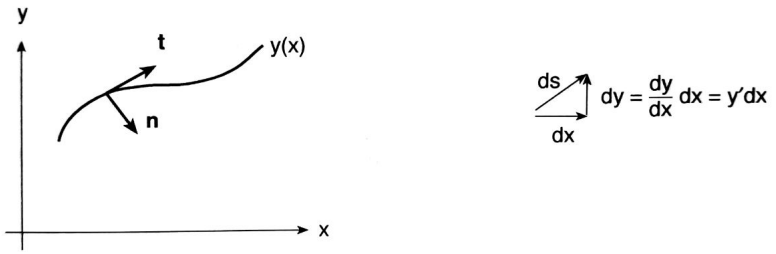


Figure 3.2. Beam represented in the plane x, y, z. Image source: [13]

.

$$\mathbf{t} = \frac{d}{ds}\begin{Bmatrix} x(s) \\ y(s) \end{Bmatrix} = \begin{Bmatrix} \frac{dx}{ds} \\ \frac{dy}{dx}\ \frac{dx}{ds} \end{Bmatrix} = \frac{dx}{ds}\begin{Bmatrix} 1 \\ y'(x) \end{Bmatrix} = \frac{1}{\sqrt{1+y'(x)^2}}\begin{Bmatrix} 1 \\ y'(x) \end{Bmatrix} \tag{3.6}$$

$$\mathbf{t}' = \frac{\mathrm{dt}}{\mathrm{d}x}\frac{\mathrm{d}x}{\mathrm{ds}} = \ldots = \frac{-y''(x)}{[1+y'(x)^2]^{3/2}}\frac{1}{\sqrt{1+y'(x)^2}}\begin{Bmatrix} y'(x) \\ -1 \end{Bmatrix} \tag{3.7}$$

From these equations, the curvature κ and the normal versor $\mathbf{n}$ are directly presented here:

$$\begin{aligned} \kappa &= \frac{-y''(x)}{[1+y'(x)^2]^{3/2}} \\ \mathbf{n} &= \frac{1}{\sqrt{1+y'(x)^2}}\begin{Bmatrix} y'(x) \\ -1 \end{Bmatrix} \end{aligned} \tag{3.8}$$

3.2 The deformable beam and the de Saint Venant problem

By considering a beam as a deformable object, it is possible to solve the elastic problem in a complete and exact manner, i.e. by studying the behaviour of the beam itself in the field of behaviour indicated by the elastic field. This problem is called the de Saint Venant problem, and with it, the volume of the beam is considered, removing the simplification whereby the beam was only considered by referring to its axis, and its deformability is also considered. The solution of de Saint Venant's problem leads to a solution that is acceptable from an engineering point of view by deriving an exact solution.
The form of the solid object being studied with de Saint Venant's problem is a rectilinear cylinder with an arbitrary base, representing a beam with constant cross-section A. The reference system of the body is chosen such that the axis represented with the versor z is collinear to the axis of the beam, while the axes of the system represented with the versors x and y are principal and barycentric, Fig. 3.3

By making a cut perpendicular to the axis of the beam, coincident as said with the z-axis, the section of the beam is obtained, and acting on a generic point of it, the tension vector can be identified: $\mathbf{t}_z = \{\tau_{zx}, \tau_{zy}, \sigma_z\}^T$.

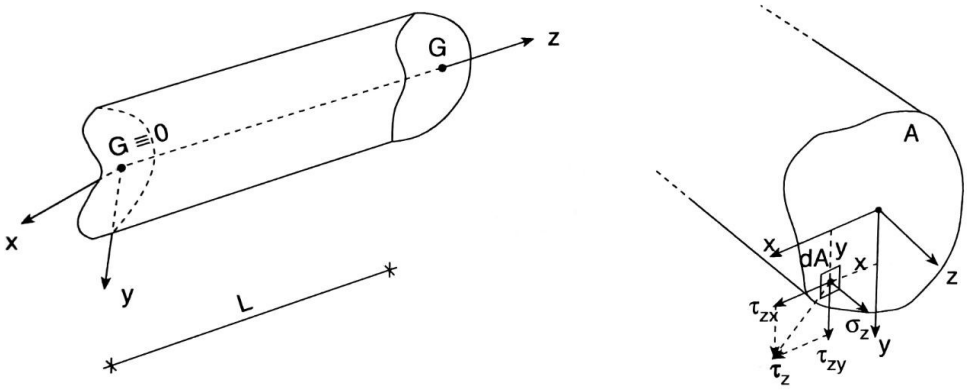


Figure 3.3. Beam represented in a system of reference x, y, z. Image source: [13]

On the beam's sections, the resultants functions of the tension vector components can be written as:

$$
\begin{aligned}
\mathrm{N} &= \int_{\mathrm{A}} \sigma_z \ \mathrm{dA} && \text{normal force} \\
\mathrm{T}_x &= \int_{\mathrm{A}} \tau_{zx} \ \mathrm{dA} && \text{shear force along } x \\
\mathrm{T}_y &= \int_{\mathrm{A}} \tau_{zy} \ \mathrm{dA} && \text{shear force along } y \\
\mathrm{M}_x &= \int_{\mathrm{A}} y \ \sigma_z \ \mathrm{dA} && \text{bending moment around } x \\
\mathrm{M}_y &= -\int_{\mathrm{A}} x \ \sigma_z \ \mathrm{dA} && \text{bending moment around } y \\
\mathrm{M}_t &= \int_{\mathrm{A}} (x \ \tau_{zy} - y \ \tau_{zx}) \ \mathrm{dA} && \text{torque}
\end{aligned}
\tag{3.9}
$$

The solution of the de Saint Venant problem provides the tension vector components as functions of the resultants, thus inverting these equations.
Analyzing the tensions, they are properties of the point of application itself, and therefore they are dependent from x, y, z; resultant instead are not functions of the application point of a force, because they are properties of a section, so they

depends only by z.

The material of the beam, solving the de Saint Venant problem, is assumed as linearly elastic, isotropic and homogeneous. The boundary conditions of the beam in this case are not important, thus there is no need to place any particular constraint.

Regarding the loads, no volume forces are present here, and just the edges sections of the beam are loaded; the lateral area of the beam is assumed as unloaded, considering $\mathbf{n}$ as the outgoing surface normal:

$$\mathbf{t}_n = \{\sigma_n,\ \tau_{nz},\ \tau_{nm},\}^T = \mathbf{0}$$
$$\mathbf{n} = \{\alpha_x,\ \alpha_y,\ 0\}^T$$

Taking the normal $\mathbf{n}$, one has that τ_{zn} is the component of the tension vector referring to the shear orthogonal to the z-axis, i.e. to the section of the beam itself, and the property holds for it that:

$$\begin{aligned} \tau_{nz} = 0 \qquad , \qquad \tau_{zn} = 0 \\ \tau_z = \{\tau_{zn},\ \tau_{zm}\}^T = \{0,\ \tau_{zm}\}^T \end{aligned} \tag{3.10}$$

From Eq. 3.10, it is clear that on the perimeter of the section of the beam, the tangential tension vector τ_z is parallel to the edge of the section itself.

An important postulate thought up and described by de Saint Venant forms the basis of the problem itself, simplifying its solution. The postulate says: "*If different stress distributions having the same results are applied to the base sections of the beam, beyond a certain distance from the bases called the extinction distance, the stress state is essentially identical.* [13] This is of course valid from an engineering point of view, but not mathematically.
This postulate is very important, since it guarantees that the solution of de Saint Venant's problem for a given stress distribution is valid for any other stress distribution than the above defined extinction zone, as long as the other stress distribution is characterised by the same results as the first; it is possible to say that this is a direct consequence of the fact that the stress distribution on the base sections of the beams only influences the areas of the beam that are very close to the bases, not influencing the stress state beyond the extinction distance, which is therefore only a function of the results.
For isotropic beams, the extinction distance is generally satisfactorily approximated by assuming it to be equal to the maximum characteristic dimension of the transversal beam section, as can be seen in Fig. 3.4 . Therefore, the greater the length of the beam, the greater will also be the part of the length covered by the solution of de Saint Venant's problem, minimising the extinction zones close to the bases. [13]

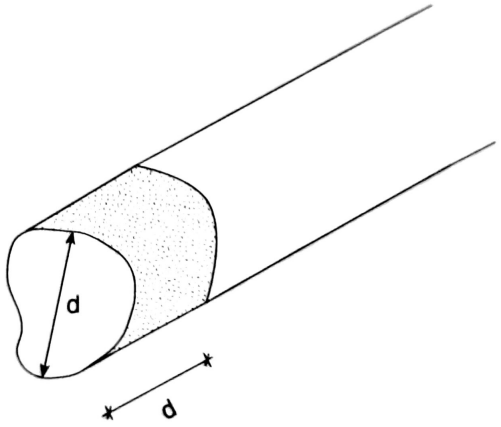


Figure 3.4. The extinction zone in an isotropic material beam. Image source: [13]

.

Referring to simple examples, if one considers a beam with a length equal to 5 times the diameter $L = 5\ d$, de Saint Venant's solution is valid in the central part of the beam, which is: $L - 0.2\ L - 0.2\ L = 0.6\ L$; on the other hand, if one considers a beam with a length equal to 20 times the diameter $L = 20\ d$, the solution of de Saint Venant is valid in the central part of the beam which results as: $L - 0.05\ L - 0.05\ L = 0.9\ L$.

de Saint Venant's solution loses its validity, however, when considering cases in which the beams are non-compact or anisotropic: in this case, the approximation of the extinction distance with the length of the beam's maximum diameter is no longer valid, since the distribution of stresses on the beam's base section influences a much larger section of the beam, making the entire beam a function of it. For this type of beam, the extinction length may even exceed the length of the beam itself.[13] In this thesis work, a beam consisting of isotropic material is therefore considered.

3.3 Semi-inverse hypothesis for the solution of the beam problem

For the solution of de Saint Venant's problem of the beam, another hypothesis is very important, called the *Semi-Inverse Hypothesis*, which consists of looking for the solution to the problem in the form: [13]

$$\sigma_x = \sigma_y = \tau_{xy} = 0 \tag{3.11}$$

Analysing the behaviour resulting from this hypothesis, it can be seen that it emphasises the centrality of the behaviour given by longitudinal stresses with respect to transverse stresses, assuming that by taking cuts of the beam parallel to the

z-axis, the vector of stresses previously defined is directed along the axis itself, thus not transmitting transverse stresses.

Defining the tensor of stresses $\mathbf{T}$ at a generic point of the de Saint Venant solid, considering applied the semi-inverse hypothesis that cancels all the components of the tensor itself, except for the components linked to $\mathbf{t_z}$ which is unaltered, it is valid:

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & \tau_{zx} \\ 0 & 0 & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix} \tag{3.12}$$

in which the the main stresses are:

$$\begin{gathered} \sigma_1 = \frac{\sigma_2 - \sqrt{\sigma_2^2 + 4\tau_z^2}}{2} \leq \sigma_2 = 0 \leq \sigma_3 = \frac{\sigma_2 + \sqrt{\sigma_z^2 + 4\tau_z^2}}{2} \\ \tau_z^2 = \tau_{xz}^2 + \tau_{yz^2}^2 \end{gathered} \tag{3.13}$$

so, from these Eq. 3.13, at each point there is a plane state of stress.
The maximum tangential stress is:

$$\tau_{\text{man}} = \frac{\sigma_3 - \sigma_1}{2} = \frac{\sqrt{\sigma_2^2 + 4\tau_2^2}}{2} \tag{3.14}$$

As far as the deformational state is concerned, one obtains, again as a consequence of the semi-inverse hypothesis:

$$\begin{gathered} \varepsilon_z = \frac{\sigma_2}{\mathrm{E}} \qquad \varepsilon_x = \varepsilon_y = -v\ \varepsilon_z = -v\ \frac{\sigma}{\mathrm{E}} \\ \gamma_{xy} = 0 \qquad \gamma_{zx} = \frac{\tau_{zx}}{\mathrm{G}} \qquad \gamma_{zy} = \frac{\tau_{xy}}{\mathrm{G}} \end{gathered} \tag{3.15}$$

The elastic strain and complementary energy densities are therefore:

$$\begin{gathered} \mathscr{E}_{de}(\mathbf{E}) = \frac{1}{2}\left[\mathrm{E}\ \varepsilon_z^2 + \mathrm{G}\left(\gamma_{zx}^2 + \gamma_{zy}^2\right)\right] \\ \mathscr{E}_{ce}(\mathbf{T}) = \frac{1}{2}\left(\frac{\sigma_z^2}{\mathrm{E}} + \frac{\tau_{xx}^2 + \tau_z^2}{\mathrm{G}}\right) \end{gathered} \tag{3.16}$$

The solution of de Saint Venant's problem therefore depends on the results presented in Eq. 3.9 applied to the base sections of the beam, with the two cuts T_x and T_y being mechanically equal, as are the bending moments M_x and M_y, from which it follows that the cases to be studied can be divided into four different cases, for which the principle of superposition of effects will then apply: normal force,

shear-bending, bending, torsion.[13] In what follows, the cases of normal force and bending will be presented, as they are decisive in solving the problem of the beam in the case being studied in this thesis.

3.4 Normal force applied on the beam base sections: stresses, strains, energies and displacements.

In this section, normal force applied on the beam base is considered, computing stresses, strains, energies and displacements. For the study of this case, it is assumed that there are no tangential stresses and that the normal force N is equally distributed, uniformly, over the whole section of the beam, defining its tensional state as follows:

$$\sigma_z = \frac{\mathrm{N}}{\mathrm{A}} \qquad \tau_{zx} = \tau_{zy} = 0 \tag{3.17}$$

in which A is the area of the section of the beam. Here σ_z is constant over the beam section, and thanks to the fact that also N is constant over the length of the beam, σ_z can be assumed constant on the entire length of the beam: it is not depend from x, y, z.
It is immediate to demonstrate how these relations of Eq. 3.17, when inserted into Eq. 3.9, give as the only result only the normal force N.

For what concern the strains, considering the Eq. 3.15 and Eq. 3.17, the tensor of the strains is:

$$\varepsilon_z = \frac{\mathrm{N}}{\mathrm{E\ A}} \qquad \varepsilon_x = \varepsilon_y = -v\ \frac{\mathrm{N}}{\mathrm{E\ A}} \qquad \gamma_{xy} = 0 \qquad \gamma_{zx} = 0 \qquad \gamma_{zy} = 0 \tag{3.18}$$

The theory is valid for small deformations, so this is true if the normal force is small and Young's modulus E and beam cross-sectional area A are sufficiently large.

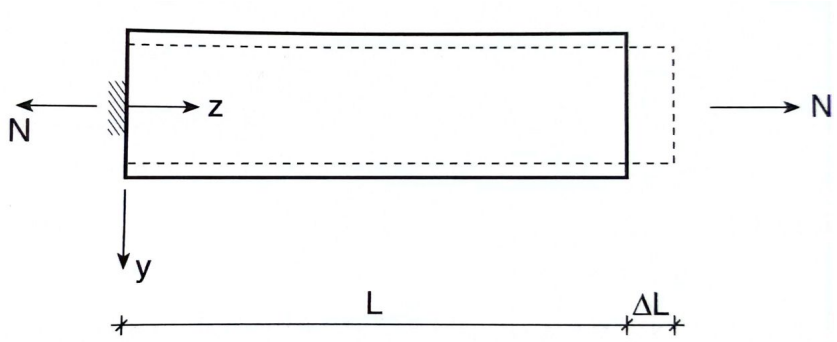


Figure 3.5. The deformed by normal force beam. Image source: [13]

If one takes a single infinitesimal piece of beam, of length dz, one immediately notices that transversally there will be an homothetic contraction, due to the Poisson effect. As for the deformation ε_z, it is constant at each point of the section, which will therefore undergo the same displacement along the z axis, resulting in a new section, displaced along the axis by a certain amount, still flat and parallel to the initial one.[13]
From the constancy of the deformation ε_z, it also follows that the change in length ΔL can be directly calculated from the relation $\varepsilon_z = \Delta\mathrm{L}/\mathrm{L}$, and is therefore valid:

$$\Delta\mathrm{L} = \frac{\mathrm{N\ L}}{\mathrm{E\ A}} \tag{3.19}$$

Two important quantities can be recognised in this relationship: EA/L is called the *axial stiffness* of the beam and represents the force required to have a unit elongation; the reciprocal of this quantity, called *axial deformability*, represents the elongation of the beam that corresponds to a unit force.

The elastic deformation and complementary energies of the entire beam are respectively:

$$\begin{aligned} e_{de} &= \int_v \mathscr{E}_{de}\ \mathrm{dV} = \frac{1}{2}\frac{\mathrm{EA}}{\mathrm{L}}\Delta\mathrm{L}^2 \\ e_{ce} &= \int_v \mathscr{E}_{ce}\ \mathrm{dV} = \frac{1}{2}\frac{\mathrm{L}}{\mathrm{EA}}\mathrm{N}^2 \end{aligned} \tag{3.20}$$

In terms of displacement, the solution to the problem of the normal force applied to the beam is:

$$u = -v \frac{\mathrm{N}}{\mathrm{EA}} x$$

$$v = -v \frac{\mathrm{N}}{\mathrm{EA}} y \tag{3.21}$$

$$w = \frac{\mathrm{N}}{\mathrm{EA}} z$$

3.5 Simple bending applied to beam base sections: stresses, deformations, energies and displacements

In the case of simple bending of the beam, it is assumed that the moment vector is directed following a principal direction of inertia of the beam section. If the beam is subjected to two equal and opposite moments applied at the ends, a deformation is obtained in which, due to symmetry, the centre section of the beam placed at the middle of its total length does not move along the z-axis, and therefore remains flat (Fig. 3.6); in general, however, the shape of the section may vary, since the consideration just made does not apply to the x, y directions.[13]

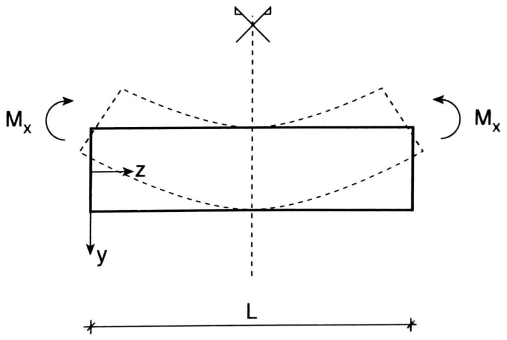


Figure 3.6. Image source: [13]

Then, it is possible to add to the beam a rigid motion, and thanks to the fact that the rigid motions do not modify the deformation, the section in the middle of the beam can still be considered plane. A very useful motion is the translation and rotation of the beam to make the deformed section $z = 0$ and the initial one coincident, as in Fig. 3.7.

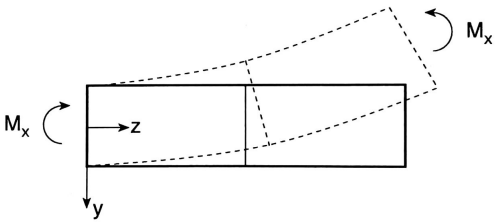


Figure 3.7. Image source: [13]

If cuts are made along the longitudinal direction of the beam, obtaining several sections, by adding appropriate rigid motions to these, the various sections of the beam can be brought back into alignment satisfying the constraint conditions by adding translation and rotation motions to the sections while keeping them flat, as in Fig. 3.8. This therefore makes it possible to state that each section can undergo rigid rotational and translational motions while always remaining flat. From this it follows that the deformation of a beam segment appears as in Fig. 3.9, in which the left and right sections remain flat by meeting at an ideal point C, called the centre of curvature. The beam segment thus obtained and the axis, which is initially straight, become an arc of a circle of radius R, which is nothing other than the radius of curvature. [13]

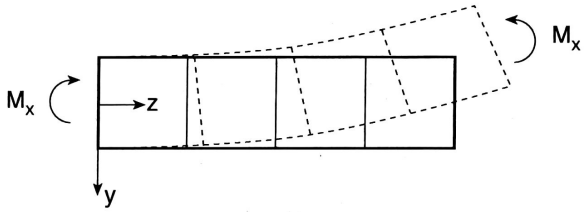


Figure 3.8. Many cuts on the longitudinal direction of the beam. Image source: [13]

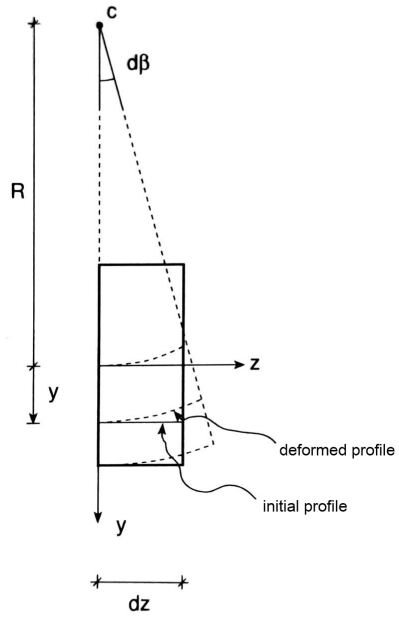


Figure 3.9. Deformation of a beam segment. Image source: [13]

From the definition of curvature $\kappa = 1/\mathrm{R}$, in the plane perpendicular to x, the normal tension can be defined as:

$$\sigma_z = \mathrm{E}\ \kappa_x\ y \tag{3.22}$$

The tangential tensions τ_{zx} and τ_{zy} are zero, from which it follows:

$$T_x = T_y = M_t = 0 \tag{3.23}$$

For the normal force N and the moments M_x and M_y, considering the x and y directions as the principal directions of inertia, the formulas apply:

$$\begin{aligned} \mathrm{N} &= \int_{\mathrm{A}} \sigma_z \ \mathrm{dA} = \mathrm{E}\ \kappa_x \int_{\mathrm{A}} y \ \mathrm{dA} = \mathrm{E}\ \kappa_x\ S_x = 0, \\ \mathrm{M}_x &= \int_{\mathrm{A}} y\ \sigma_z \ \mathrm{dA} = \mathrm{E}\ \kappa_x \int_{\mathrm{A}} y^2 \ \mathrm{dA} = \mathrm{E}\ \kappa_x\ \mathrm{I}_x, \\ \mathrm{M}_y &= -\int_{\mathrm{A}} x\ \sigma_z \ \mathrm{dA} = \mathrm{E}\ \kappa_x \int_{\mathrm{A}} x\ y \ \mathrm{dA} = \mathrm{E}\ \kappa_x\ \mathrm{I}_{xy} = 0. \end{aligned} \tag{3.24}$$

Then, from Eq. 3.24, the curvature formula is obtained:

$$\kappa_{\mathrm{x}} = \frac{1}{\mathrm{R}} = \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x} \tag{3.25}$$

one derives Navier's formula:

$$\sigma_z = \frac{\mathrm{M}_x}{\mathrm{I}_x} y \tag{3.26}$$

which makes it possible to determine the tensional state as a function of the beam's internal actions. The curvature formula is proof that in order to flex beams that are not very stiff, characterised by a Young's modulus of a small value, a small bending moment is sufficient, whereas if the Young's modulus value is high (which means that the beam is very stiff), a large value of bending moment is required, since the beam is stiff, with the risk of causing it to break.

The most important maximum normal tensile and compressive stresses are:

$$\begin{aligned} \sigma^t_{\max} &= \frac{\mathrm{M}_x}{\mathrm{W}^t}, \\ \sigma^c_{\max} &= -\frac{\mathrm{M}_x}{\mathrm{W}^c} \end{aligned} \tag{3.27}$$

where W^t and W^c are the modules of strength:

$$\begin{aligned} \mathrm{W}^t &= \frac{\mathrm{I}_x}{y_{\max}}, \\ \mathrm{W}^c &= -\frac{\mathrm{I}_x}{y_{\min}}. \end{aligned} \tag{3.28}$$

From these formulae, it is evident that the sections of best resistance to bending are those that have a large modulus of resistance W: at the same maximum tension, the beam can withstand a greater moment since, for the same bending moment, the maximum tension is smaller for sections that have large modules of resistance, as can be seen in Eq. 3.28.

Since the moment M, Young's modulus E, and moment of inertia I_x referenced to x are constant along the z direction, all beam segments curve according to the same radius of curvature and the deformation of the beam axis results in an arc of a circumference of radius R. The figure Fig. 3.10 shows the transverse displacement of the beam axis, which belongs to the neutral plane and therefore does not elongate; its longitudinal displacement is zero in this case.

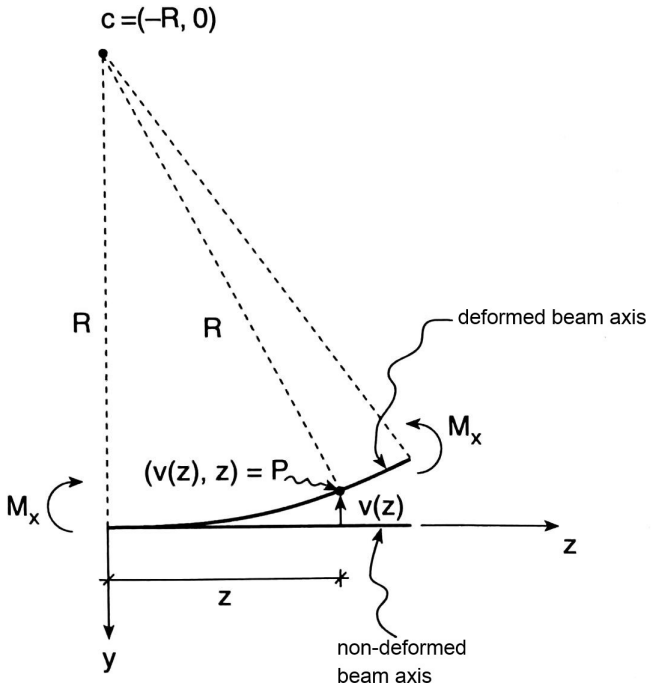


Figure 3.10. Transverse displacement of the beam axis. Image source: [13]

The expression of the analytical form of the deformed beam is retrieved assuming the distance between the centre of curvature C = (-R, 0) and a generic point P = (v(z),z) of the arch of circumference is equal to R.

$$\mathrm{R}^2 = d^2(P, C) = (v(z) + \mathrm{R})^2 + (z - 0)^2 \tag{3.29}$$

from which

$$0 = v^2(z) + 2v(z)R + z^2 \tag{3.30}$$

Here, with the assumption of small displacements for the beam, $v^2(z)$ is negligible with respect to the other terms, and the transverse displacement of the beam axis is:

$$v(z) = -\frac{z^2}{2\ \mathrm{R}} = -\frac{1}{2}\ \kappa_x\ z^2 = -\frac{1}{2}\ \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x} z^2 \tag{3.31}$$

Deriving this last equation, the inclination angle is:

$$\phi(z) = -v'(z) = \kappa_x\ z = \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x} z \tag{3.32}$$

assumed as positive if it develops in counter-clockwise verse.
With another derivation operation, the equation that links the curvature and the displacements is provided:

$$v''(z) = -\kappa_x \tag{3.33}$$

and the rotation angle of the final section is:

$$\phi_{\mathrm{L}} = \frac{\mathrm{M}_x\ \mathrm{L}}{\mathrm{E}\ \mathrm{I}_x} \tag{3.34}$$

The number E I_x/L, i.e. the moment it takes to have a unit rotation, is called the bending stiffness of the beam.

The elastic deformation and complementary energies of the entire beam are respectively:

$$\begin{aligned} e_{de} &= \int_{\mathrm{V}} \mathscr{E}_{\mathrm{de}}\ \mathrm{dV} = \frac{1}{2}\ \frac{\mathrm{E}\ \mathrm{I}_x}{\mathrm{L}} \phi_{\mathrm{L}}^2, \\ e_{ce} &= \int_{\mathrm{V}} \mathscr{E}_{\mathrm{ce}}\ \mathrm{dV} = \frac{1}{2}\ \frac{\mathrm{L}}{\mathrm{E}\ \mathrm{I}_x} \mathrm{M}_x^2 \end{aligned} \tag{3.35}$$

and are equal to the work done by the external moment:

$$\ell = \frac{1}{2} \mathrm{M}_x\ \phi_{\mathrm{L}} \tag{3.36}$$

The complete solution of the simple straight bending problem in terms of displacement is:

$$u = -v \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x}\ x\ y,$$

$$v = -\frac{1}{2} \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x} \left[z^2 - v \left(x^2 - y^2 \right) \right], \tag{3.37}$$

$$w = \frac{\mathrm{M}_x}{\mathrm{E}\ \mathrm{I}_x}\ y\ z$$

Some cases that turn out to be important solutions to the bending problem are considered through the extension of de Saint Venant's problem, by having the tension tensor maintained at each point, knowing the action moment M(z) inside the beam, and simplifying the formulae so as to refer only to the axis of the beam, introducing quantities that represent the bending behaviour of the beam itself. From now on, the subscript x is subtended in the following formulae.[13]
The tensional state is:

$$\begin{aligned} \tau_{zx} &= \tau_{zy} = 0 \\ \sigma_z(z) &= \frac{\mathrm{M}(z)}{\mathrm{I}(z)} y \end{aligned} \tag{3.38}$$

The displacement here is the transversal one $v(z)$, that is the unique possible for the axis of the beam. Since the beam is curved by the moment, the parameter that best represents the deformation undergone is the curvature $\kappa(z)$.
The constitutive bonding equation can then be introduced, the quantities of which depend only on z.

$$\kappa(z) = \frac{\mathrm{M}(z)}{\mathrm{E}(z)\ \mathrm{I}(z)} \tag{3.39}$$

Considering variable curvature, the congruence equation is:

$$v^{''}(z) = -\kappa(z) \tag{3.40}$$

Finally, considering p(z) as the transverse load, and c(z) the moments per unit length applied to the beam axis, the equilibrium equation is:

$$\mathrm{M}^{''}(z) = -p(z) - c^{'}(z) \tag{3.41}$$

In the study of the elastic problem for the bending deformable beam, the data are: the length L of the beam, the axial inertia modulus I(z) of the beam; the Young's modulus E(z) of the material, the constraints at the extremity with transverse constraining displacements v_0 and rotational Φ_0, the loads distributed along the beam p(z) and c(z) and concentrated at the extremity T_0 and M_0. The unknowns of the problem are thus: the kinematic transverse displacement v(z), the flexural deformation κ(z) (called curvature) and the static bending moment known as M(z) (the shear T can be directly derived from this).[13]

From the Eq. 3.33 and Eq. 3.39, it is obtained:

$$-\mathrm{M} = \mathrm{E}\ \mathrm{I}\ v^{''} \tag{3.42}$$

to which the boundary conditions are associated. The Eq. 3.42 is known as 2^{nd} order flexural elastic line differential equation.
By using the Eq. 3.42 in the Eq. 3.41, it is possible to obtain the differential equation of the higher-order transverse elastic deformation, also known as the differential equation of the 4^{th} order flexural elastic line:

$$(\mathrm{E}\ \mathrm{I}\ v^{''})^{''} = p + c^{'} \tag{3.43}$$

to which the static and kinematic boundary conditions are associated. If the elastic problem has $c = 0$ and the product E I is constant, it can be written as:

$$v^{''''} = \frac{p}{\mathrm{EI}} \tag{3.44}$$

with the following boundary conditions applied:

$$\begin{aligned} v &= v_0 \\ v^{'} &= -\Phi_0 \\ v^{''} &= -\frac{\mathrm{M}_0}{\mathrm{EI}} \\ v^{'''} &= -\frac{\mathrm{T}_0}{\mathrm{EI}} \end{aligned} \tag{3.45}$$

Considering the specific case studied in this thesis, the cantilever beam with concentrated force at the extreme, as can be see in Fig. 3.11, computing the transversal displacement with the second order elastic line equation, the moment has to be computed first, using this relation:

$$\mathrm{M}(z) = \mathrm{F}(-\mathrm{L} + z) \tag{3.46}$$

This results in:

$$v'' = -\frac{\mathrm{M}}{\mathrm{EI}} = \frac{\mathrm{F}}{\mathrm{EI}}(\mathrm{L} - z) \tag{3.47}$$

an integration performed twice is obtained:

$$v(z) = \frac{\mathrm{F}}{\mathrm{EI}}\left(\mathrm{L}\frac{z^2}{2} - \frac{z^3}{6}\right) + c_1 z + c_2 \tag{3.48}$$

The constants are determined with kinematic boundary conditions. At embedding ($z = 0$) the transverse displacement and rotation are zero.

$$\begin{aligned} 0 &= v(0) = c_2, \\ 0 &= v'(0) = c_1. \end{aligned} \tag{3.49}$$

The free extremity ($z = L$) can lower and rotate freely. There are therefore no further kinematic conditions to add to the conditions just written, as these already determine the integration constants. The displacement is therefore:

$$v(z) = \frac{\mathrm{F}}{\mathrm{EI}}\left(\mathrm{L}\frac{z^2}{2} - \frac{z^3}{6}\right) \tag{3.50}$$

and the maximum displacement, for $z = L$, is:

$$v(z) = \frac{1}{3}\frac{\mathrm{FL}^3}{\mathrm{EI}} \tag{3.51}$$

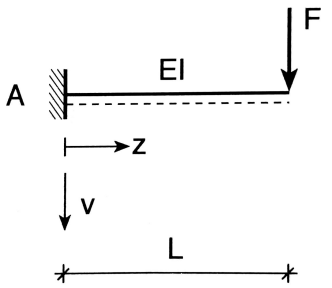


Figure 3.11. The cantilever beam with concentrated force at the extreme.
Image source: [13]

For the clamped cantilever beam problem that is solved with the use of neural networks in this thesis, an UB 305 x 127 x 42 beam steel flange cantilever beam has been chosen. The modulus of elasticity is $E = 200\ [GPa]$, the maximum distance to point from neutral axis of the beam is $y_{max} = 150\ [mm]$, and the moment of Inertia is $I = 81960000\ [mm^4]$. The cross section of the beam is represented in Fig. 3.12.

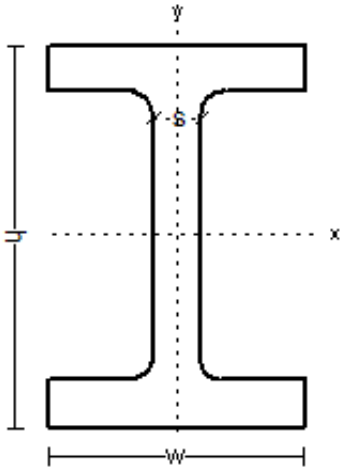


Figure 3.12. The cross section of British Universal 305 x 127 x 42 beam steel flange cantilever beam used, with $h = 307.2\,\text{mm}$, $w = 124.3\,\text{mm}$, $s = 8\,\text{mm}$, *sectional area* $= 53.4\,\text{cm}^2$, *weight* $= 41.9\,\text{kg/m}$.

Chapter 4

Neural networks used to study a cantilever clamped beam

This chapter describes in detail the neural network architecture used for the cantilever beam study, with particular reference to the network composition, the reasons behind the choices made and the results obtained.

For the study of the beam problem, it was decided to create a neural network that could investigate the displacement of the free end of the cantilever beam, and the maximum stress. The load has been chosen to be applied to the beam is a concentrated force placed at the free end of the beam itself: for this load condition, the maximum displacement, Eq. 3.51, suffered by the beam with reference to the initial condition of the unloaded beam is found at the free end, the point of application of the force itself; the maximum stress, Eq. 4.1, suffered by the beam with the described load condition coincides with the point at which the bending moment of the beam is maximum, which is the point of constraint of the cantilever beam. This load configuration was chosen because it allows, if a possible future user so desire, to move the point of force application along the beam without having to make major modifications to the neural network architecture. This fact is guaranteed by the analysis of the elastic line theory previously presented in Sec. 3.5.

$$\sigma_{max} = \frac{y_{max}\, F\, L}{I} \tag{4.1}$$

The network trained as described below will be able to predict the two required output of the maximum displacement and the maximum stress values for the training data-set characteristics, that are: cantilever beam lengths L between 500 and 5000 millimetres and the modulus of concentrated force applied to the free end of the beam F between 100 and 3000 Newtons. The input values provided to the neural network are the only values that vary in the input data-set, so the choice was made on the length of the beam and the modulus of the force applied to the beam itself. Of course, if the neural network has to be used for a different load condition, e.g. a uniformly distributed load along the beam, or for the analysis of a different problem, e.g. a torsion problem, it would be necessary to re-train the network with a data-set pertaining to the load condition case just described, or to implement new data in addition to those provided for bending, in order to be able to solve a torsion problem,

for example. See Fig. 4.1.

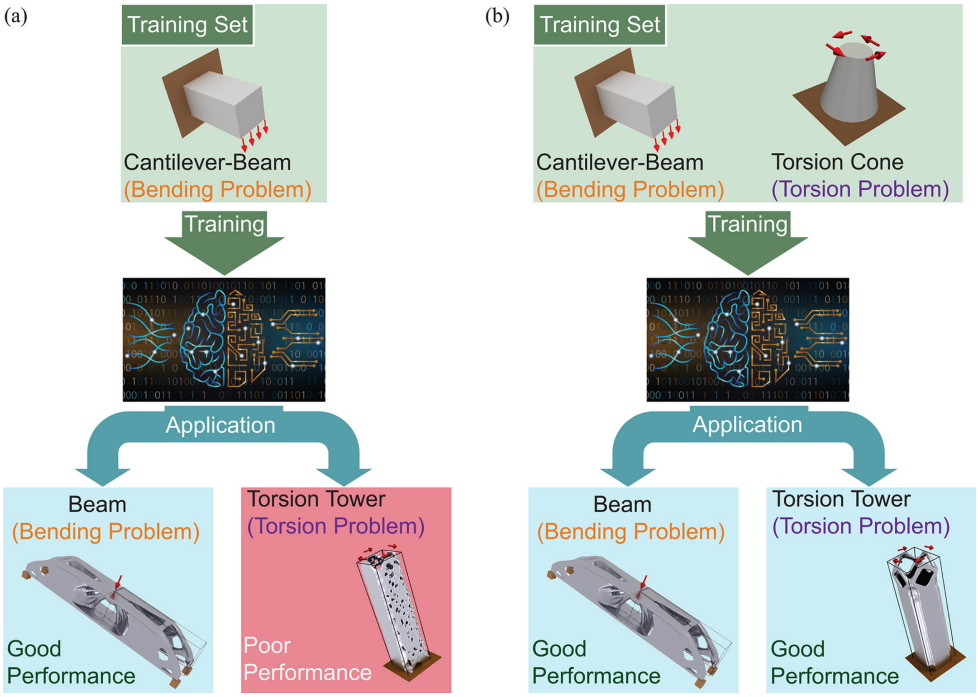


Figure 4.1. The training set affects how well a machine learning model works. If the model is only trained with bending problems, it will likely only do well on bending problems. The model doesn't work well when it faces a torsion problem. If one adds more training samples that include both bending and torsion problems for instance, the model performs well for both kinds of problems. .
Image source: [24]

The neural network is then given a training data-set, on which the learning algorithm creates the model it will use for prediction. The network, in order to implement a performance evaluation while being trained, is chosen to use 10% of the training data-set to make an estimate of the performance that the algorithm is making the network learn, and on this, through backward feedback, it goes on to improve the learning process itself. Hence, new data unknown to the network in terms of input parameters, i.e. beam lengths and concentrated force modulus values that are totally new and that the network has never seen during the training process. On this new, unknown data-set, the model obtained by the learning algorithm, in the network training phase, is used to make a prediction, comparing then with the analytically calculated reference results. The process is represented graphically for ease of understanding in Fig. 4.2.

4.1 Neural network architecture

A neural network architecture characterised by the element of three network layers was therefore created for this case study. The first layer, also called the input layer, is characterised the chosen activation function, that is the rectified linear unit type

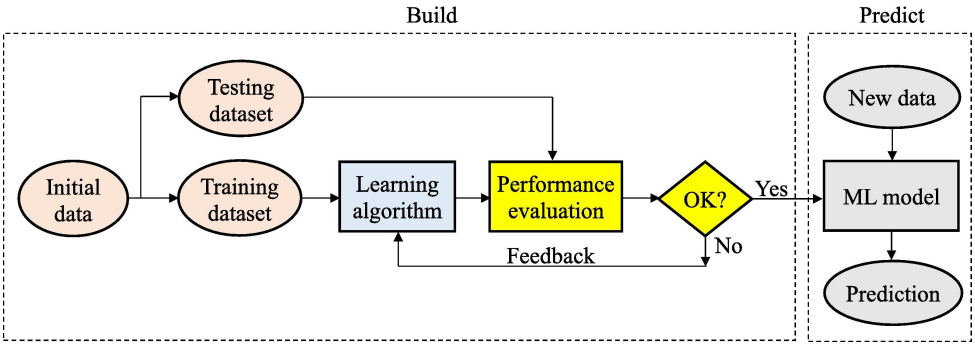


Figure 4.2. The workflow of machine learning for the implemented neural network. Image source: [27]

described in the chapter referring to the different possibilities of activation functions. The second and third layers were constructed by choosing, after many attempts, the same number of nodes of the input layer and using also for these the same activation function used in the first layer. This configuration proved to be the best in terms of performance and output of the results, allowing for low errors.

In order to create this neural network architecture, various attempts were of course made to analyse even networks with layers containing different numbers of nodes between them, but the configuration containing an equal number of nodes in each layer, with the network consisting of three layers, turned out to be much better. Immediately following the last layer is an output layer, with simple linear activation, which serves to collect the output produced by the third layer and present it to the end user. It is considered to be of great interest not so much to present the results by varying the number of layers or the combination of nodes within them, but rather to study the architecture defined above with the same number of nodes for each layer, thus varying the quantity of these nodes, and above all to study the number of example cases provided in the network's input data-sets, which were used to train the model, and then were evaluated with inputs unknown to the network to better analyse the performance of the model thus obtained.

The number of samples provided in the input data-set was made to vary from 300 to 20000, more precisely using data-sets with 300, 500, 800, 1000, 1500, 2000, 3000, 5000, 10000, 15000, 20000, always keeping the output number fixed at the two described at the beginning of this chapter, i.e. the maximum displacement with respect to the reference condition of the cantilever beam and the maximum stress it undergoes with the load condition described.

A very important parameter in neural network training is the batch size, which defines the number of samples that will be propagated through the neural network itself. This means that if for example the batch size chosen is 64, the algorithm takes the first 64 samples provided in the training data-set of the network right the network, then proceeds to take the next 64 and train the network again. This procedure is performed until the entire training data-set has been propagated through the network, resulting in a final training pattern of the network. Using a batch size smaller than the total number of samples certainly requires using less memory and

thus the complete network training procedure using fewer samples uses less memory in total. This is particularly important in the event that the complete data-set cannot be stored in the machine's memory. Usually, the fact of using a very small batch size means that network training is much faster as the network evaluation weights are updated with each propagation. On the other hand, it is important to note that a batch size smaller than the total number of samples, the smaller it is, the smaller the estimate of the gradient between the previous solution and the next solution used to train the network. For the beam case study, a batch size of 1/10 of the total number of samples in the data-set was chosen as the problem is linear and not difficult to interpret for the network itself.

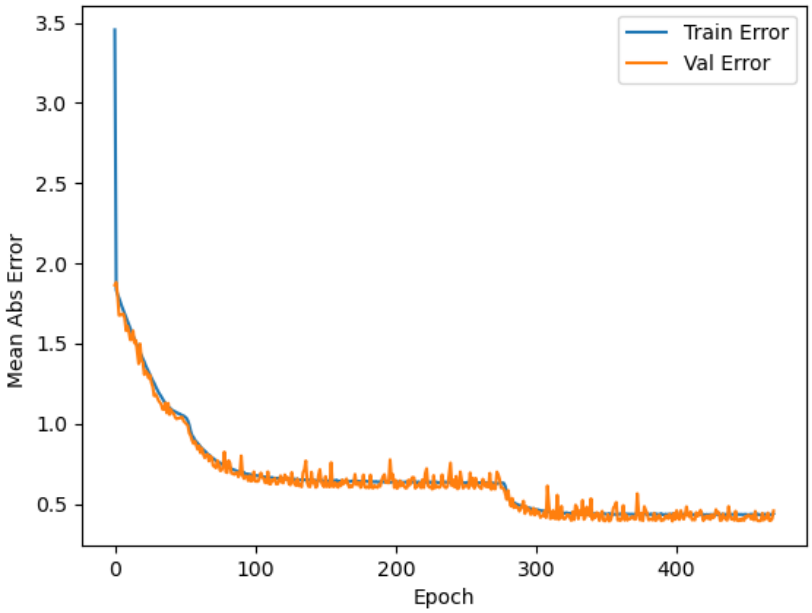
It is important to specify how the number of epoch before convergence is reached clearly depends on the training start point of the network itself. In order to avoid a random dependence on this value, each network training process is performed four times for each data-set, thus protecting against randomness, the results below all refer to the best case of network training for the cycle of four processes performed for each data-set.

4.2 Neural Network training

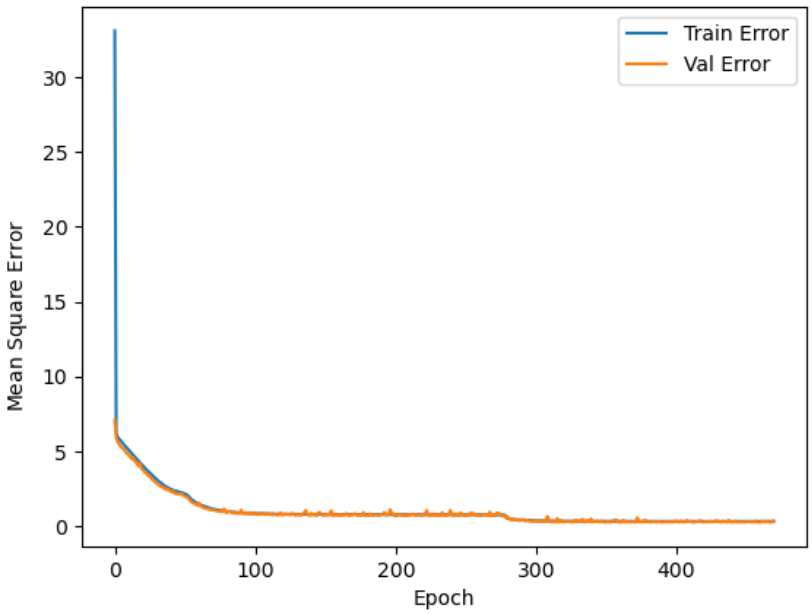
In the case study of a network architecture consisting of three layers each composed of two nodes, the results can be analysed in detail in Tab. 4.1. From a careful observation of the values obtained, where MAE and MSE are respectively the mean absolute error and mean squared error, Eq. 2.2 and Eq. 2.1, it is immediately clear that the best performing model is the one which uses a data-set composed of 10000 samples, which manages to reach convergence in 470 epochs of network training. In this case, the fact of using data-sets composed of a greater number of rows, equal to 10000 and 20000 respectively, can be seen by observing the graphs of the errors, which leads to a faster convergence as the number of rows increases, but at the same time also leads to an increase in the error values. For this reason, both models with a greater number of rows compared to the data-set corresponding to 10000 samples are not suitable for studying the beam problem for an optimal result.

data-set samples	**epoch**	**MAE**	**MSE**
300	365	1.120669	2.533913
500	293	1.065609	2.278502
800	205	0.738644	0.903677
1000	121	0.764835	0.987241
1500	143	1.091989	2.450032
2000	154	1.089331	2.432707
3000	196	1.073450	2.367747
5000	556	0.769681	1.105923
10000	470	0.438509	0.356045
15000	357	0.630449	0.797177
20000	314	0.763505	1.083946

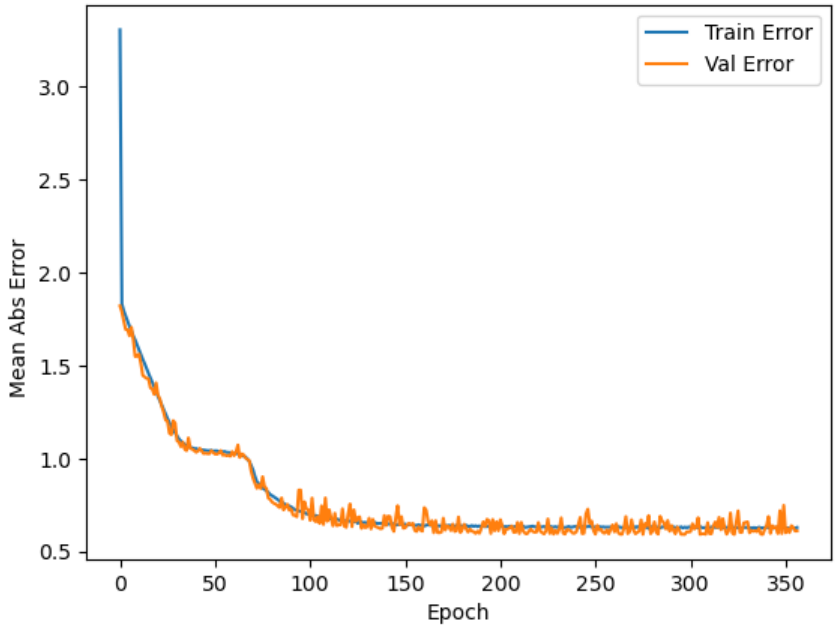
Table 4.1. Training performance analysis with Mean Absolute Error MAE and Mean Squared Error MSE for the neural network architecture with three identical layers, with 2 nodes each.



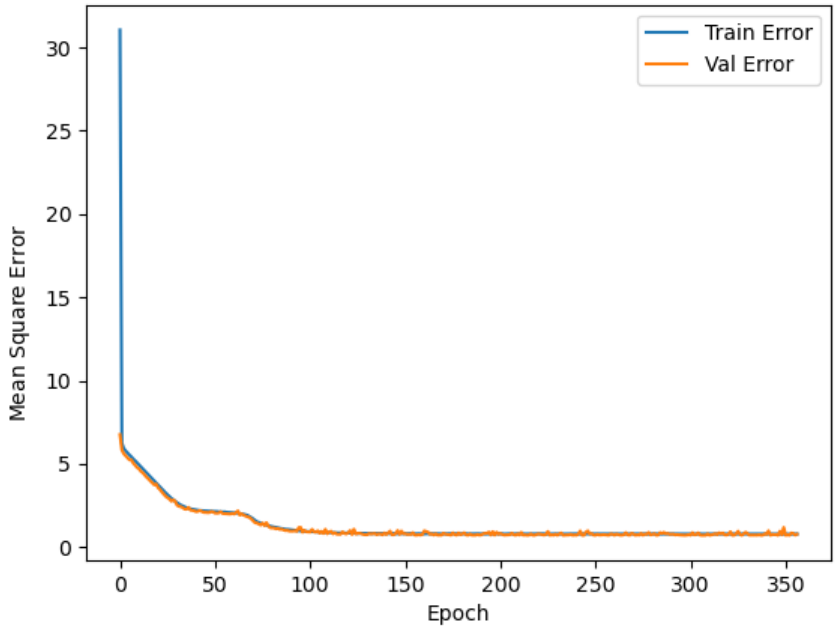
(a) The MAE Mean Absolute Error for the NN with 2 nodes per layer and 10000 rows data-set.



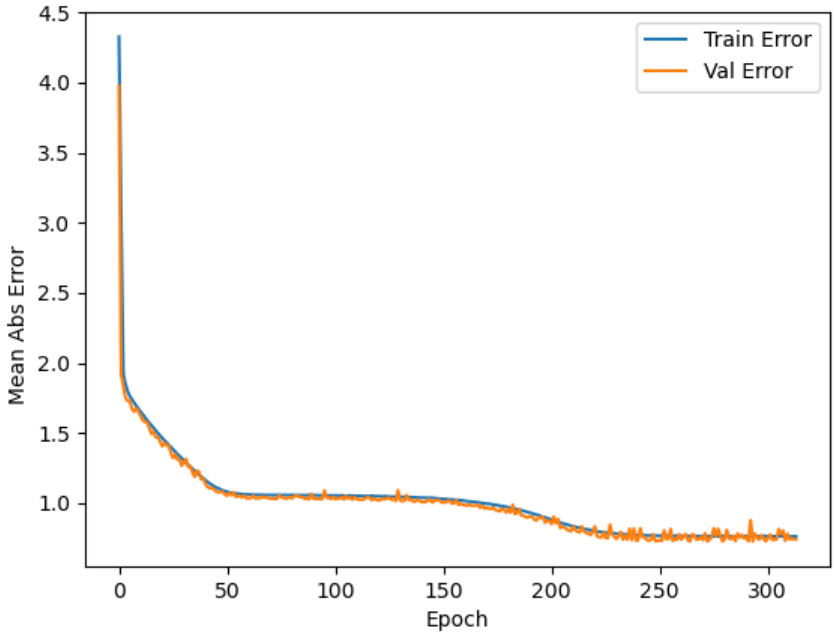
(b) The MSE Mean Squared Error for the NN with 2 nodes per layer and 10000 rows data-set.



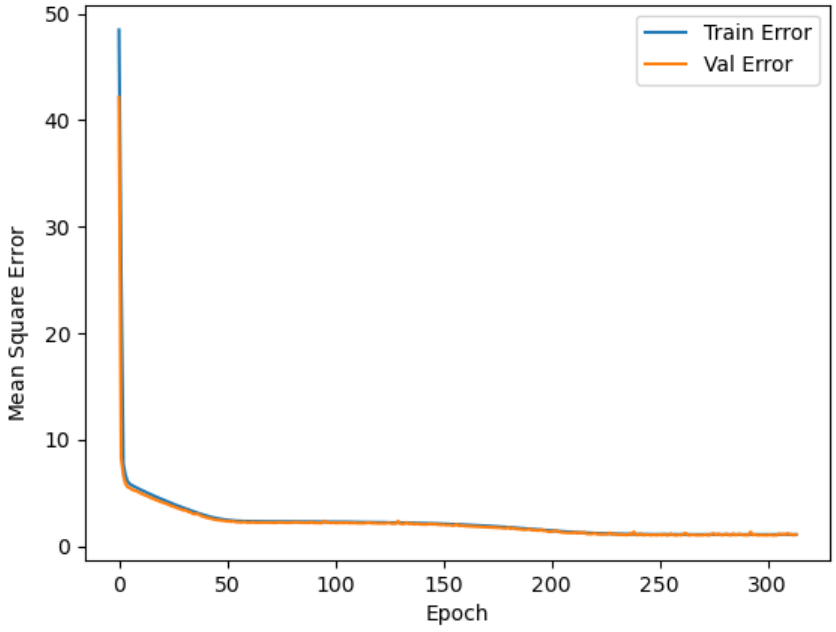
(a) The MAE Mean Absolute Error for the NN with 2 nodes per layer and 15000 rows data-set.



(b) The MSE Mean Squared Error for the NN with 2 nodes per layer and 15000 rows data-set.



(a) The MAE Mean Absolute Error for the NN with 2 nodes per layer and 20000 rows data-set.

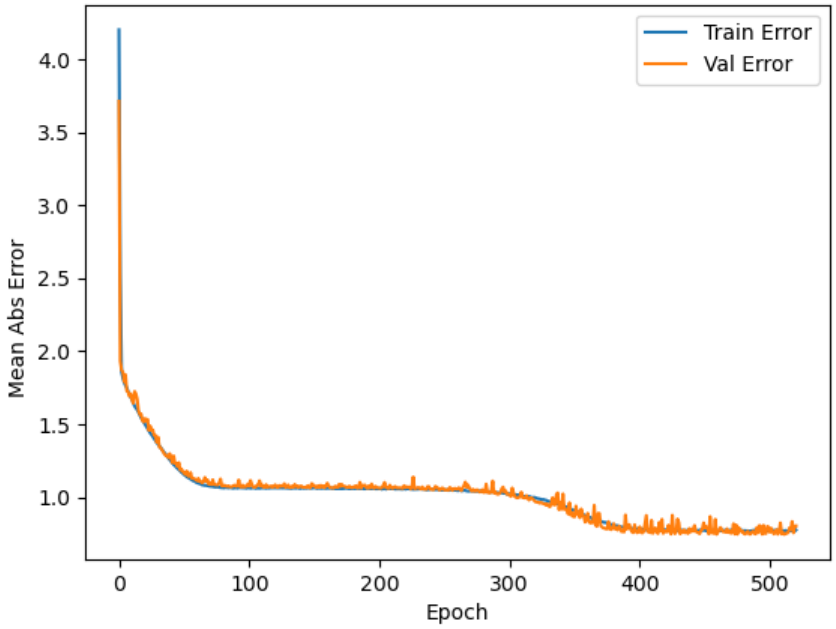


(b) The MSE Mean Squared Error for the NN with 2 nodes per layer and 20000 rows data-set.

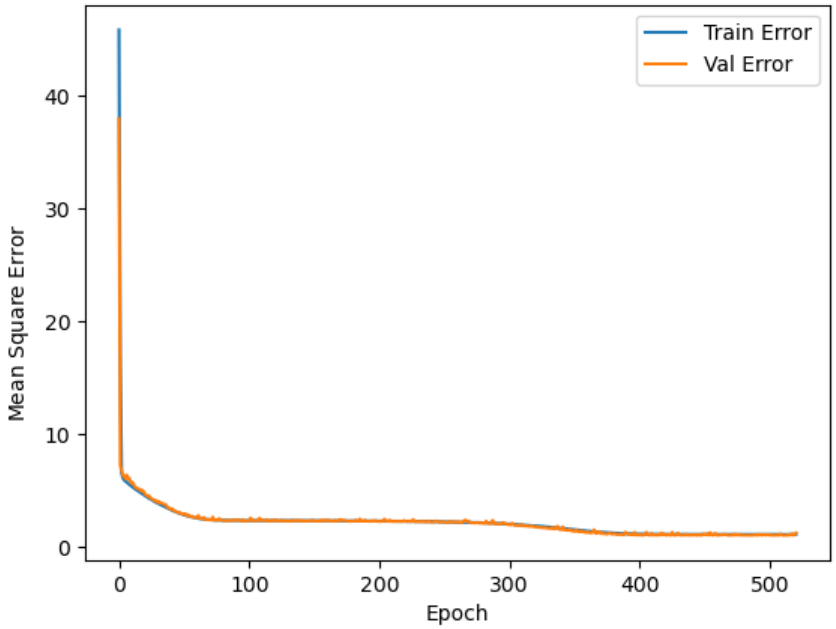
Going to increase the number of nodes used in each layer, and studying the configuration with four nodes for each layer, the results summarised in Tab. 4.2 indicate that the model trained with a data-set of 5000 rows converges in 875 epochs as the best performing model. A close look at the table shows how the model corresponding to a training data-set of 800 rows, and thus 800 samples, also performs very well, converging in 500 epochs and obtaining error values very close to those of the data-set of 5000 samples.

data-set samples	**epoch**	**MAE**	**MSE**
300	530	0.425301	0.337350
500	147	0.722469	0.918963
800	500	0.279488	0.176064
1000	701	0.387378	0.264482
1500	428	0.351970	0.237619
2000	294	0.385620	0.266079
3000	510	0.692519	0.882345
5000	875	0.264414	0.134604
10000	730	0.375680	0.257806
20000	117	0.655016	0.808569

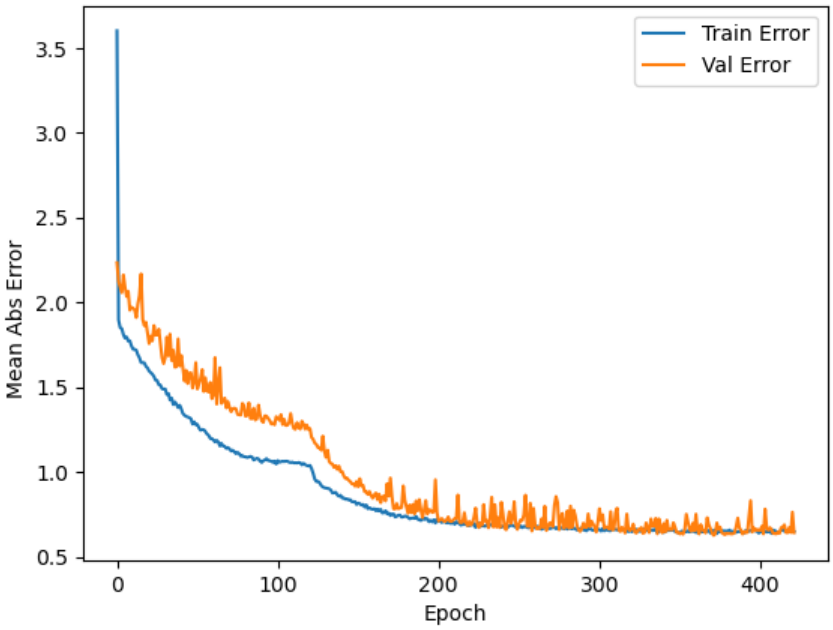
Table 4.2. Training performance analysis with Mean Absolute Error MAE and Mean Squared Error MSE for the neural network architecture with three identical layers, with 4 nodes each.



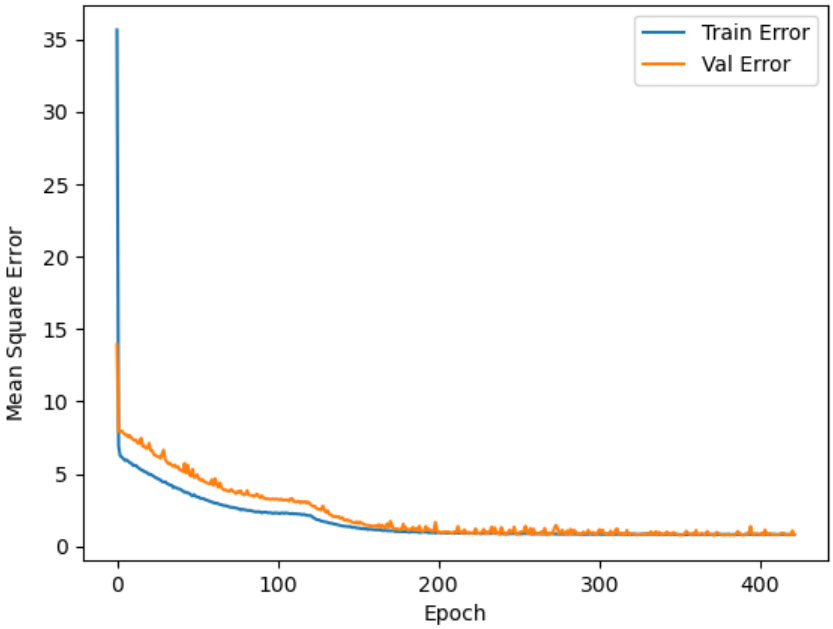
(a) The MAE Mean Absolute Error for the NN with 4 nodes per layer and 5000 rows data-set.



(b) The MSE Mean Squared Error for the NN with 4 nodes per layer and 5000 rows data-set.



(a) The MAE Mean Absolute Error for the NN with 4 nodes per layer and 800 rows data-set.

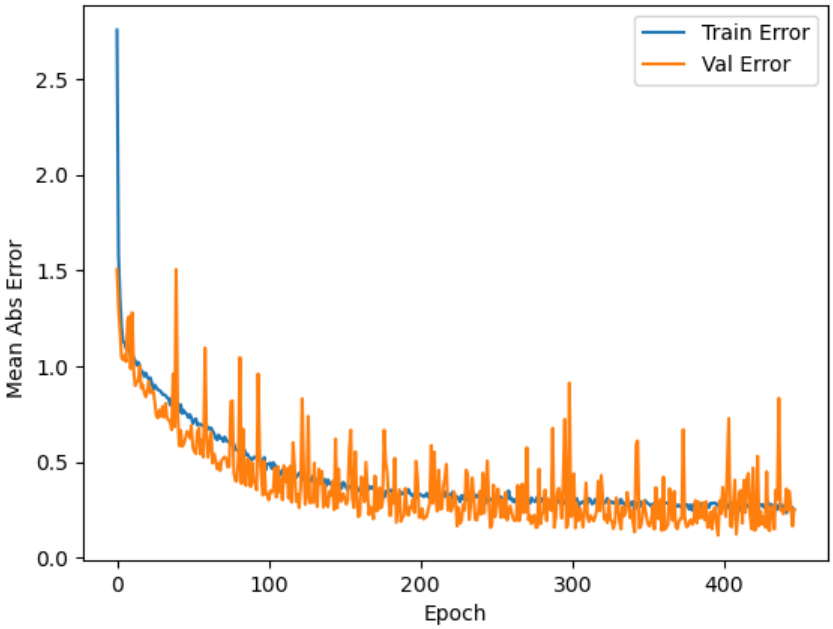


(b) The MSE Mean Squared Error for the NN with 4 nodes per layer and 800 rows data-set.

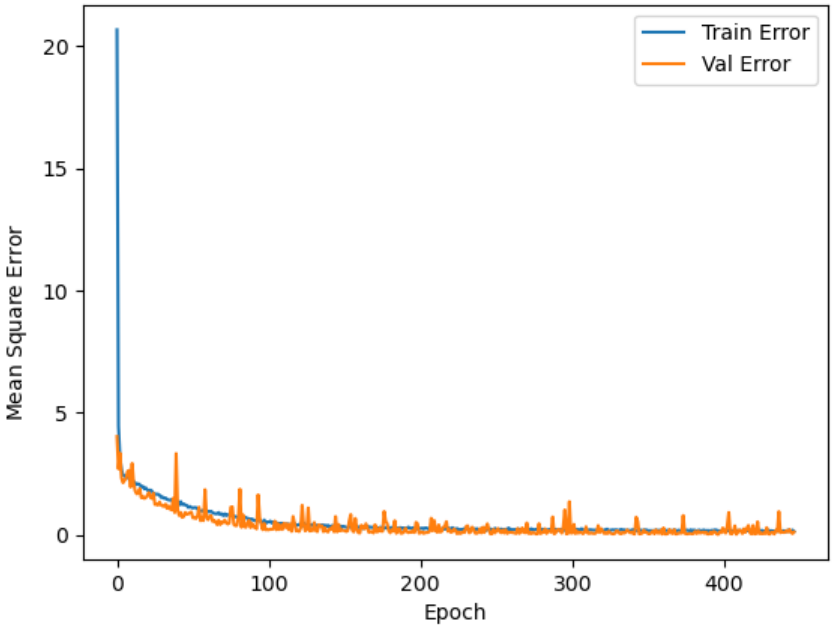
Finally, using a network architecture consisting of 8 nodes per layer, always using three layers, it is clear that it is sufficient to stop and use a data-set with a maximum of 800 samples, see Tab. 4.3. This is due to the fact that using a greater number of nodes makes the network more skilled and capable in terms of understanding the mathematical relationship describing the problem, thus also being able to reduce the number of samples needed to study the network itself in order to obtain a prediction model that is as generic as possible. In this case, in fact, the best performing training model turns out to be the one formed from a data-set of 500 rows which reaches convergence after 447 epochs, with very small error values.

data-set samples	**epoch**	**MAE**	**MSE**
300	555	0.282809	0.193953
500	447	0.251228	0.147881
800	643	0.273160	0.166848

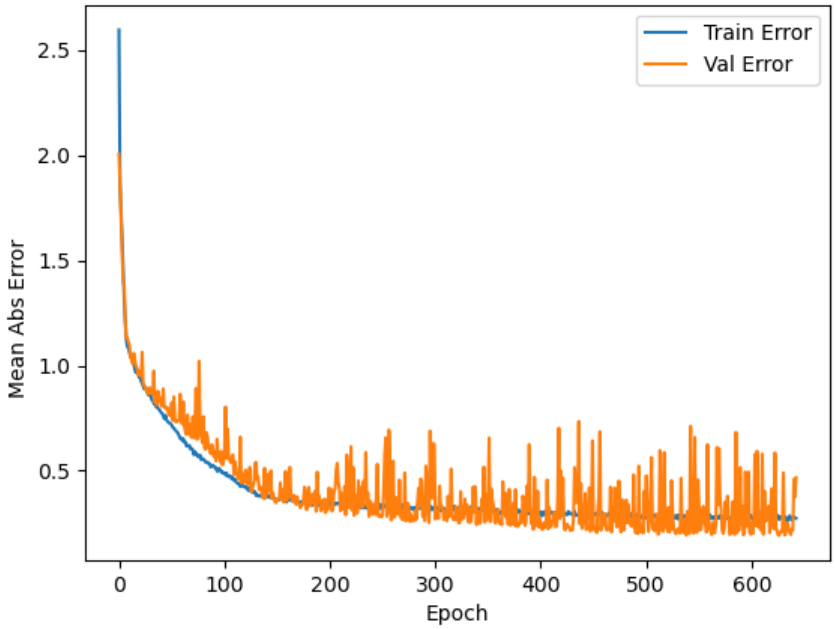
Table 4.3. Training performance analysis with Mean Absolute Error MAE and Mean Squared Error MSE for the neural network architecture with three identical layers, with 8 nodes each.



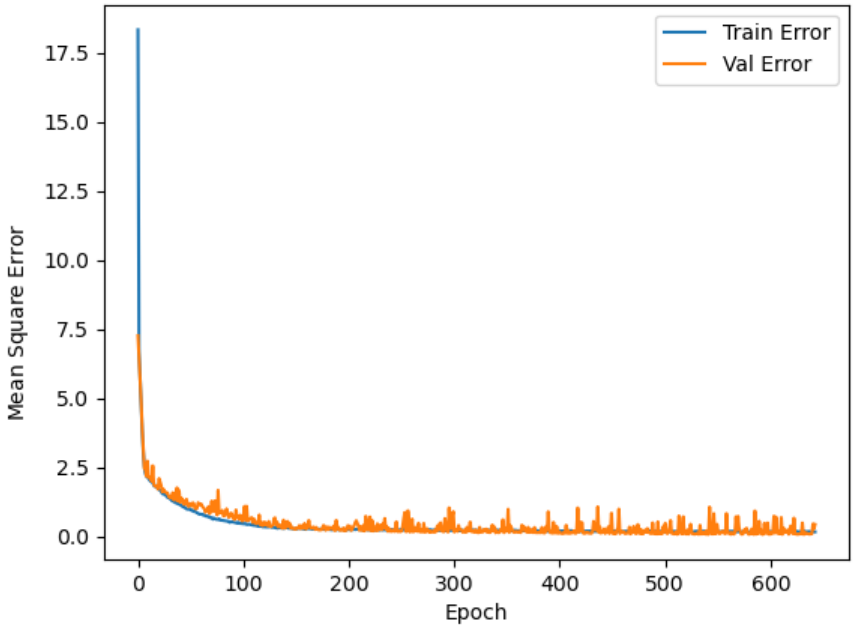
(a) The MAE Mean Absolute Error for the NN with 8 nodes per layer and 500 rows data-set.



(b) The MSE Mean Squared Error for the NN with 8 nodes per layer and 500 rows data-set.



(a) The MAE Mean Absolute Error for the NN with 8 nodes per layer and 800 rows data-set.

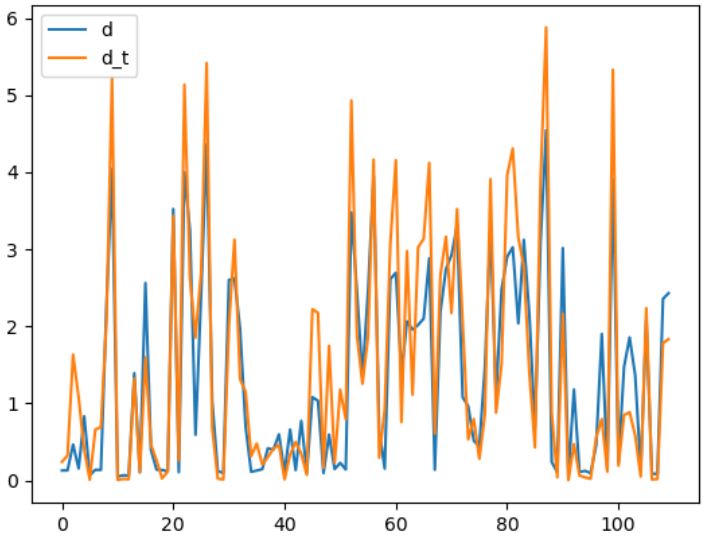


(b) The MSE Mean Squared Error for the NN with 8 nodes per layer and 800 rows data-set.

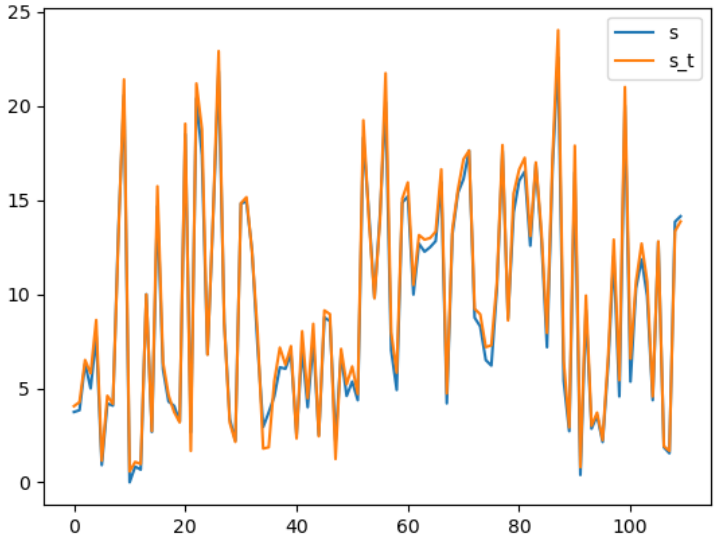
4.3 Neural Network evaluation

The oscillations that can be seen in the graphs, especially on the validation curve corresponding to the evaluation of the Mean Absolute Error MAE, are due to noise introduced by the fact that the validation data-set used by the network in the training phase, corresponding to 10% of the data supplied in the training data-set itself, fails to provide the algorithm with sufficient information to evaluate the ability of the model to generalise, even though the curve corresponding to the training of the neural network has a decidedly smoother and very stable behaviour, behaving as if it were learning the values by heart instead of learning the mathematical relationship hidden behind them. As the beam problem is a simple problem, this noise is extremely limited and therefore perfectly negligible, without affecting the performance of the neural network.

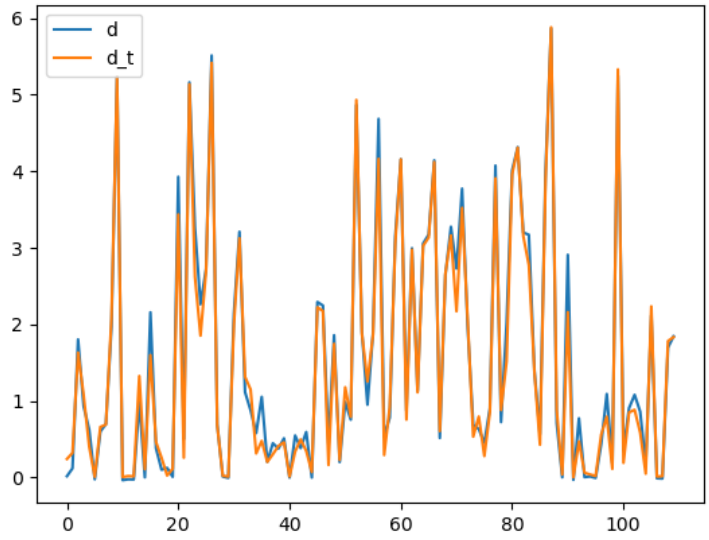
Therefore, when evaluating the best model for solving the problem of beam bending due to concentrated force applied to the free end of the cantilever beam, it can be seen that the two best models are the one with 500 data-set samples and 8 nodes for each layer and the model with 5000 data-set samples and 4 nodes for each layer. Below one can find the images of the neural network validation test done with beam length data and concentrated force modules as explained above. As a comparison, the same validation test performed with the neural network model consisting of 2 neurons per layer is also shown, although this performs less well than the other two.



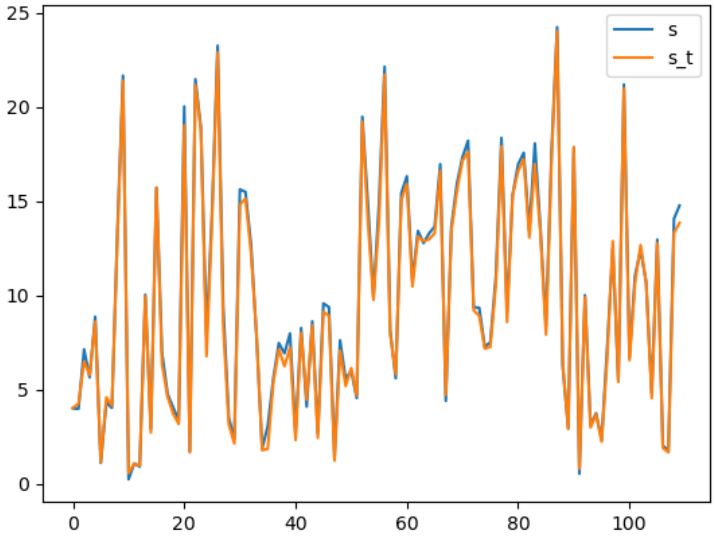
(a) Evaluation of the maximum displacement given as output from the 2-2-2 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 10000 rows data-set.



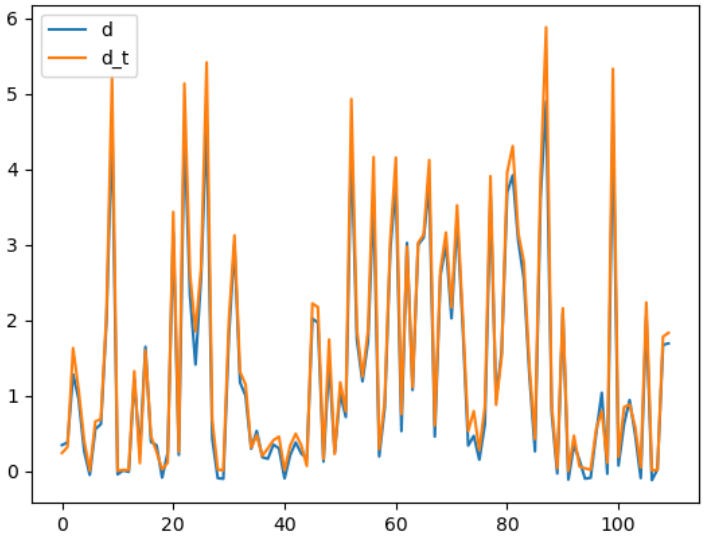
(b) Evaluation of the maximum stress given as output from the 2-2-2 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 10000 rows data-set.



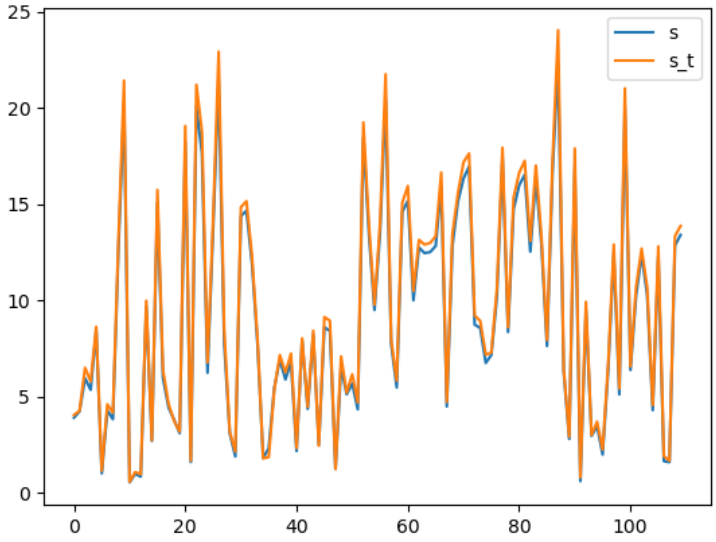
(a) Evaluation of the maximum displacement given as output from the 4-4-4 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 5000 rows data-set.



(b) Evaluation of the maximum stress given as output from the 4-4-4 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 5000 rows data-set.



(a) Evaluation of the maximum displacement given as output from the 8-8-8 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 500 rows data-set.



(b) Evaluation of the maximum stress given as output from the 8-8-8 nodes/layer NN, varying the length of the beam and the modulus of the concentrated force, for the 500 rows data-set.

Chapter 5

Shell structures and the Transfer Matrix Method

In this chapter the Transfer Matrix Method TMM is presented. It will be used to study the stresses in a cylindrical shell structure, allowing to compare the results obtained analytically implementing it, with the results obtained from the neural network. To use and understand this method, an introduction on the Fourier series is here presented.

5.1 The Fourier series

The Fourier series is a mathematical method that allows any periodic function to be described by the summation of simple harmonic functions whose frequencies are integer multiples of the frequency of the original function. The assumptions that allow a generic function f(t) to be developed in a Fourier series are known as Dirichlet conditions and are as follows:

- the function is periodic;
- J(t) is continuous at intervals, that is, in each period there are at most a finite number of discontinuities with finite amplitude;
- J(t) possesses a finite number of maxima and minima within a single period;
- the integral of |J(t)| calculated over a single period converges to a finite value.

When these conditions are met, the Fourier series converges to f(t) at every point where the function is continuous; at discontinuities, the series converges instead to the mean value. There are two alternative forms of the Fourier series, defined in the real and complex fields, respectively. The real one, that is the one used in this thesis work, will be now presented.

5.1.1 Real Fourier series

Given a function f(t) that satisfies Dirichlet conditions, the corresponding real Fourier series is written:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos n\omega_p t + b_n \sin n\omega_p t) \tag{5.1}$$

Where:

$$\omega_p = \frac{2\pi}{T} \tag{5.2}$$

is the circular frequency of the periodic function (also called the fundamental frequency), while:

$$\frac{a_0}{2} = \frac{1}{T} \int_0^T f(t) \mathrm{d}t \tag{5.3}$$

represents the average value of the function.
Fourier series terms have the important property of orthogonality:

$$\int_0^T \cos n\omega_p t \cdot \cos m\omega_p t \ \mathrm{d}t = 0 \qquad n \neq m$$

$$\int_0^T \sin n\omega_p t \cdot \cos m\omega_p t \ \mathrm{d}t = 0 \tag{5.4}$$

$$\int_0^T \sin n\omega_p t \cdot \sin m\omega_p t \ \mathrm{d}t = 0 \qquad n \neq m$$

To prove the orthogonality conditions, one uses Werner's trigonometric formulas and rewrite the integrals as:

$$\cos n\omega_p t \cdot \cos m\omega_p t = \frac{1}{2} [\cos(n-m)\omega_p t + \cos(n+m)\omega_p t]$$

$$\sin n\omega_p t \cdot \cos m\omega_p t = \frac{1}{2} [\sin(n-m)\omega_p t + \sin(n+m)\omega_p t] \tag{5.5}$$

$$\sin n\omega_p t \cdot \sin m\omega_p t = \frac{1}{2} [\cos(n-m)\omega_p t - \cos(n+m)\omega_p t]$$

It is now easy to calculate the integrals of Eq. 5.4 and prove the orthogonality conditions. Werner's own formulas make it possible to calculate:

$$\int_0^T \cos^2 n\omega_p t \ \mathrm{d}t = \int_0^T \sin^2 n\omega_p t \ \mathrm{d}t = \int_0^T \frac{1}{2} \ \mathrm{d}t = \frac{T}{2} \tag{5.6}$$

The orthogonality conditions are fundamental for calculating the coefficients of the series. For this purpose, one multiplies both terms of Eq. 5.1 by sin $m\omega_p t$ and integrate between 0 and T:

$$\begin{aligned}\int_0^T f(t)\sin m\omega_p t\ \mathrm{d}t = \frac{a_0}{2}\int_0^T \sin m\omega_p t\ \mathrm{d}t + \\ + \sum_{n=1}^{\infty} a_n \int_0^T \cos n\omega_p t \cdot \sin m\omega_p t\ \mathrm{d}t + \\ + \sum_{n=1}^{\infty} b_n \int_0^T \sin n\omega_p t \cdot \sin m\omega_p t\ \mathrm{d}t = b_m \frac{T}{2}\end{aligned} \tag{5.7}$$

In fact, the integral that multiplies a_0 is null because it is computed over an internal number of cycles; the integrals that multiply the a_n coefficients are all null according to Eq. 5.4, just as all integrals that multiply the b_n coefficients are null, except for the case $n = m$, which corresponds instead to equation Eq. 5.6. In a similar manner, the a_m coefficient can be isolated by multiplying both terms of Eq. 5.1 by $m\omega_p t$ and then integrating in the interval [0, T]. In conclusion, the coefficients of the real Fourier series are calculated as:

$$\begin{aligned}a_n &= \frac{2}{T}\int_0^T f(t)\cos n\omega_p t\ \mathrm{d}t \\ b_n &= \frac{2}{T}\int_0^T f(t)\sin n\omega_p t\ \mathrm{d}t\end{aligned} \tag{5.8}$$

Alternatively to Eq. 5.1, the real Fourier series can be written as:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} r_n \cos\left(n\omega_p t - \varphi_n\right) \tag{5.9}$$

In which amplitude and phase are:

$$\begin{aligned}r_n &= \sqrt{a_n^2 + b_n^2} \\ \varphi_n &= \arctan\left(b_n, a_n\right)\end{aligned} \tag{5.10}$$

5.1.2 Fourier series expansion for non-uniformly distributed loads

Loads distributed non-uniformly on the circumference of a shell are difficult to solve mathematically with differential equations. For this reason, for rotationally symmetrical structures, Fourier series are used in which the load is developed. For each member of the Fourier series, the differential equations are solved separately in order to later superimpose the partial solutions to an overall solution. This procedure is easily possible with a linear theory. The actually infinite Fourier series can practically be broken off after a finite number of terms. For an arbitrary Fourier series on the circumference distributed, periodic function applies (periodic with $T = 2\pi$): [18]

$$f(\varphi) = \frac{a_0}{2} + a_1\cos\varphi + b_1\sin\varphi + a_2\cos(2\varphi) + b_2\sin(2\varphi) + \ldots \tag{5.11}$$

respectively:

$$f(\varphi) = \frac{a_0}{2} + \sum_{m=1}^{\infty} a_m \cos(m\varphi) + \sum_{m=1}^{\infty} b_m \sin(m\varphi) \tag{5.12}$$

The coefficients are as follows:

$$\begin{aligned} a_m &= \frac{1}{\pi} \int_0^{2\pi} f(\varphi) \cos(m\varphi) \mathrm{d}\varphi \\ b_m &= \frac{1}{\pi} \int_0^{2\pi} f(\varphi) \sin(m\varphi) \mathrm{d}\varphi \end{aligned} \tag{5.13}$$

If the function is symmetrical to $\phi = 0$, the b_m become zero. If the function is anti-metric to $\phi = 0$, the a_m result in zero. The first four terms of a Fourier series for a periodic, symmetric function are shown in Fig. 5.1.

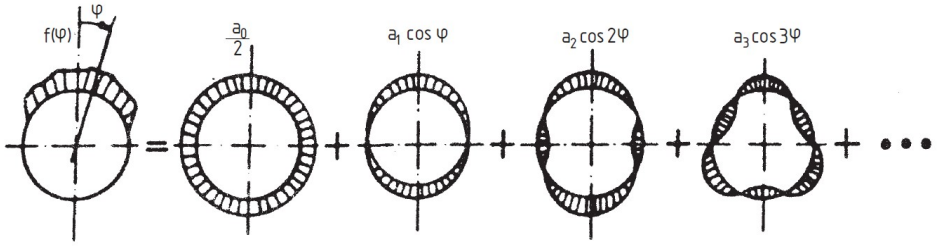


Figure 5.1. Fourier series of a periodic, symmetrical function. Image source: [18]

5.2 Shell theory

Shell structures are structures consisting of shells that can be stiffened by chords or edge elements for force transmission and stabilisation. A shell is idealised as a thin-walled, curved structure whose wall thickness is considerably smaller than the other dimensions. The curvature of the shell offers the possibility of absorbing distributed loads normal to its skin as pure normal forces (membrane forces) with almost constant stress over the wall thickness. This makes it possible to design structures that have a uniform material stress across the cross-section, and which are therefore very stable.[31]

Some examples of shell structures are a launcher fuel tank as in Fig. 5.5, a launcher fairing as in Fig. 5.2 and in Fig. 5.4, a rocket inter-stage or the fuselage of a plane, as in Fig. 5.3.

Figure 5.2. A fairing of a launcher system. Image source: [10]

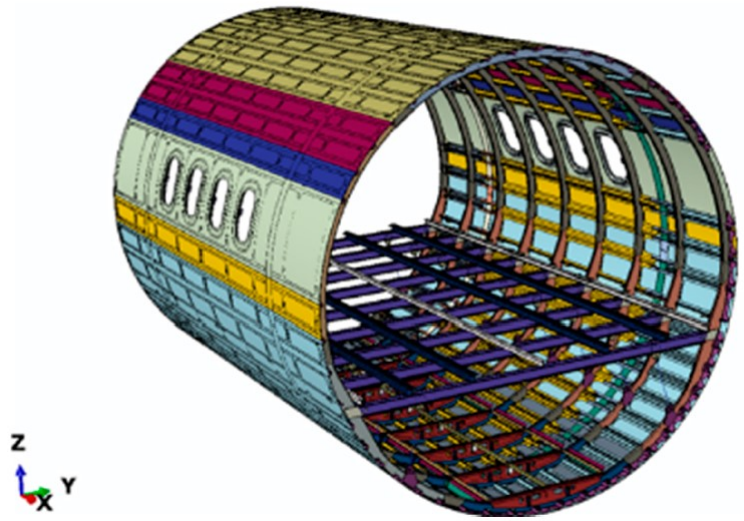


Figure 5.3. An airplane fuselage. Image source: [20]

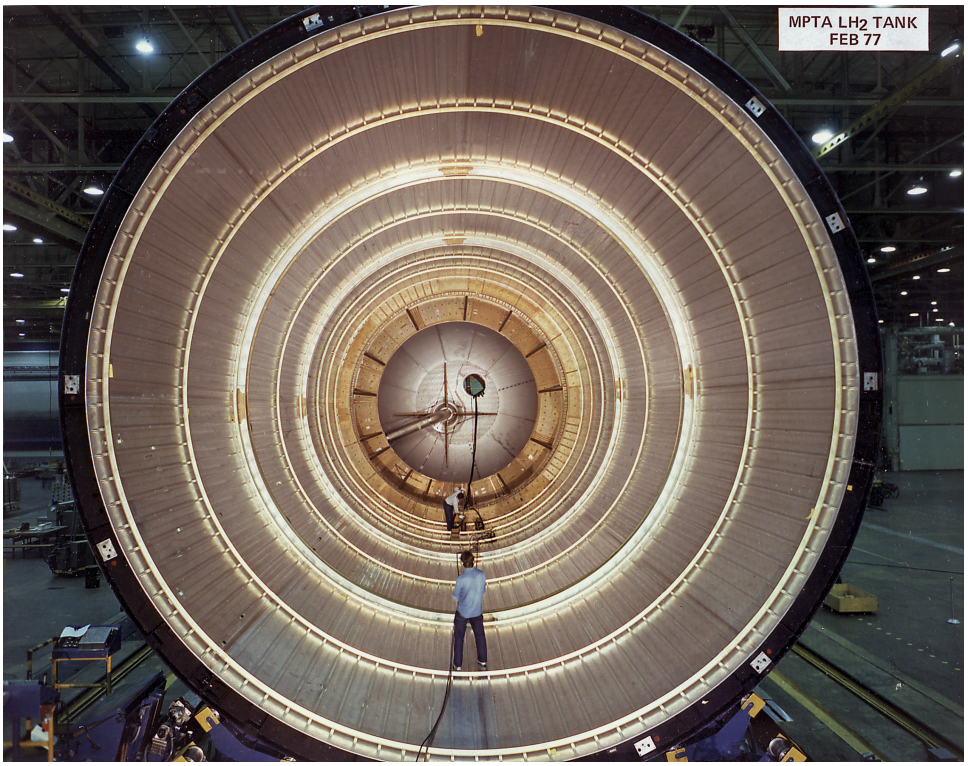


Figure 5.4. The Space Shuttle external tank. Image source:[16]

Figure 5.5. A fuel tank shell. Image source:[3]

Figure 5.6. The Orbital ATK Cygnus cargo ship is seen after final approach to the International Space Station. Image source:[4]

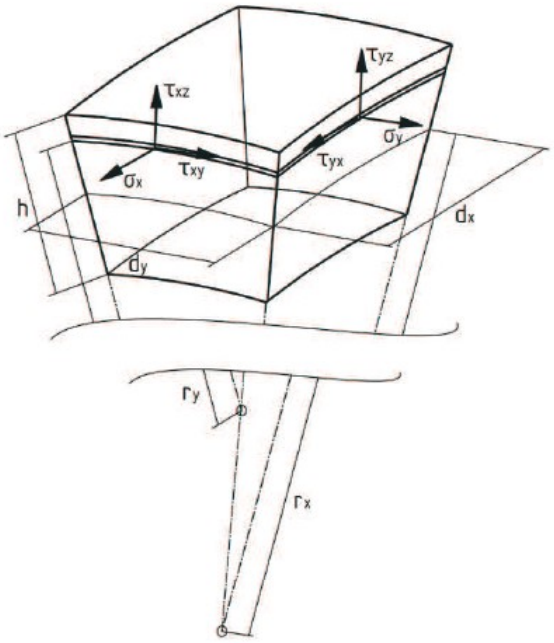


Figure 5.7. Stresses on the shell element. Image source: [9]

Due to the curvature in the arched beam, the stress becomes considerably smaller with decreasing beam height than in the straight beam. The calculation of shells is based on the following assumptions: [18]

- the shell wall thickness is much smaller than the shell dimensions;
- the deformations are smaller than the shell wall thickness (geometric linear theory);
- straight lines that are perpendicular to the center surface before deformation remain straight and are perpendicular to the deformed center surface after deformation;
- normal stresses acting perpendicular to the center surface are negligibly small.

The stresses acting on a partial shell are shown respectively in Fig. 5.7.

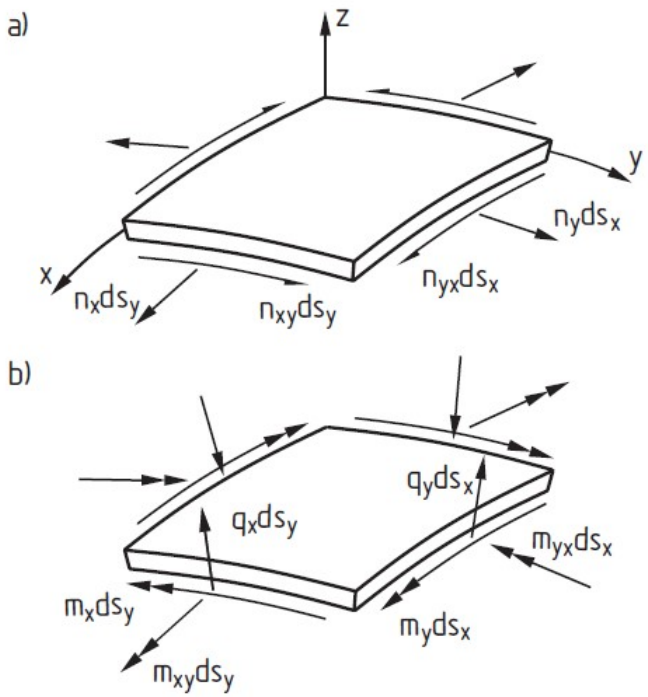


Figure 5.8. Force flows (a) and moment flows (b) on the shell element. Image source: [6]

By integrating the stresses over the wall thickness, one obtains the internal forces and moments on the shell element, as in Fig. 5.8, which are given as force flows (force/length) and moment flows (moment/length), as is usual for shell structures. The shear stresses on the side faces of the element which are perpendicular to each other are equal: $\tau_{xy} = \tau_{yx}$. Nevertheless, the shear fluxes n_{xy} and n_{yx} as well as the torsional moment fluxes m_{xy} and m_{yx} are different in the general case when the radii of curvature r_x and r_y are not equal.

The force flows (force/length) can be written as:

$$\begin{aligned} n_x &= \int_{-h/2}^{h/2} \sigma_x \left(1 + \frac{z}{r_y}\right) \mathrm{d}z \\ n_y &= \int_{-h/2}^{h/2} \sigma_y \left(1 + \frac{z}{r_x}\right) \mathrm{d}z \\ n_{xy} &= \int_{-h/2}^{h/2} \tau_{xy} \left(1 + \frac{z}{r_y}\right) \mathrm{d}z \\ n_{yx} &= \int_{-h/2}^{h/2} \tau_{yx} \left(1 + \frac{z}{r_x}\right) \mathrm{d}z \end{aligned} \tag{5.14}$$

The moment flows (moment/length) can be written as:

$$\begin{aligned} m_x &= -\int_{-h/2}^{h/2} \sigma_x \left(1 + \frac{z}{r_y}\right) z \, \mathrm{d}z \\ m_y &= -\int_{-h/2}^{h/2} \sigma_y \left(1 + \frac{z}{r_x}\right) z \, \mathrm{d}z \\ m_{xy} &= -\int_{h/2}^{h/2} \tau_{xy} \left(1 + \frac{z}{r_y}\right) z \, \mathrm{d}z \\ m_{yx} &= -\int_{-h/2}^{h/2} \tau_{yx} \left(1 + \frac{z}{r_x}\right) z \, \mathrm{d}z \end{aligned} \tag{5.15}$$

$$\begin{aligned} q_x &= \int_{-h/2}^{h/2} \tau_{xz} \left(1 + \frac{z}{r_y}\right) \mathrm{d}z \\ q_y &= \int_{-h/2} \tau_{yz} \left(1 + \frac{z}{r_x}\right) \mathrm{d}z \end{aligned} \tag{5.16}$$

The shell works most effectively when it is stressed in the direction of the shell skin, i.e. a membrane stress state is present. In this case, the deformations u and v in x and y direction are constant over the wall thickness. Because of the assumption of small wall thickness, constant stresses σ_x , σ_y , τ_{xy} and τ_{yx} can be assumed over the wall thickness. For the force fluxes of the membrane stress state, Eq. 5.14, if the constant stresses are drawn in front of the integral, one finds:

$$\int_{-h/2}^{h/2} \frac{z}{r_y} \, \mathrm{d}z = \int_{-h/2}^{h/2} \frac{z}{r_x} \, \mathrm{d}z = 0 \tag{5.17}$$

finally:

$$\begin{aligned} n_x = \sigma_x \int_{-h/2}^{h/2} \mathrm{d}z = \sigma_x h \\ n_y = \sigma_y h \\ n_{xy} = n_{yx} = \tau_{xy} h = \tau_{yx} h \end{aligned} \tag{5.18}$$

The membrane tension state is established when:

- loads are evenly distributed normal to the shell skin and do not change abruptly;
- the wall thickness does not change abruptly;
- the curvature of the shell does not change abruptly;
- there are no discontinuous inclinations of the shell skin;
- if the boundary conditions do not hinder the deformations of the membrane stress ratio.

During the phase of designing, shell structures are frequently analyzed using the membrane theory due to its ability to deliver precise outcomes across extensive regions. In regions featuring a bending aberration, the optimization of the solution can be augmented through the employment of a corrective methodology employing the principles of the vessel theory. The theories of membrane and vessel belong to the fourth order and exhibit the potential of closed-form solutions in numerous scenarios.[31] The corresponding solutions appear to be fitting for parameter inquiries. The analysis of coupling between the membrane and bending state necessitates sophisticated computations, which can be accomplished via the application of shell bending theory. This approach yields 8th order theories, for which an exact solution can only be derived in select, uncomplicated instances. Generally speaking, the issue at hand can be resolved through the use of numerical techniques. Based on the elucidated load-bearing characteristics, certain guidelines can be inferred for the design of formwork structures, which presents a significant challenge:[18]

- Avoid direct concentrated introduction of forces normal to the shell skin. Bending-resistant, curved elements (e.g., spans) are required for force application.

- Avoid discontinuous changes in the inclination of the shell skin. If they are unavoidable, suitable curved elements (e.g., spans) should be used for load transfer.
- Avoid wall thickness jumps.
- Avoid discontinuous changes in curvature. Under certain conditions, undesirable bending disturbances due to changes in curvature can be compensated by wall thickness jumps.
- Apply concentrated longitudinal forces via force application elements to avoid local stress concentrations.

In this context only rotational shells will be considered, which are an important application in technology. The central surface of the shell is created by rotating a meridian curve around the shell axis as in Fig. 5.9. Depending on the shape of the meridian, different shell shapes are obtained, such as circular cylindrical shell, conical shell, spherical shell, etc. The position of a point C on the rotation shell is determined by two angles, the angle ϕ of a meridian plane through C related to the zero meridian and the angle θ that the surface normal through C encloses with the shell axis. In the case of straight meridians or meridian curves with a point of inflection, the angle θ does not lead to clear results. In these cases, the meridian arc length s is used instead. [18]

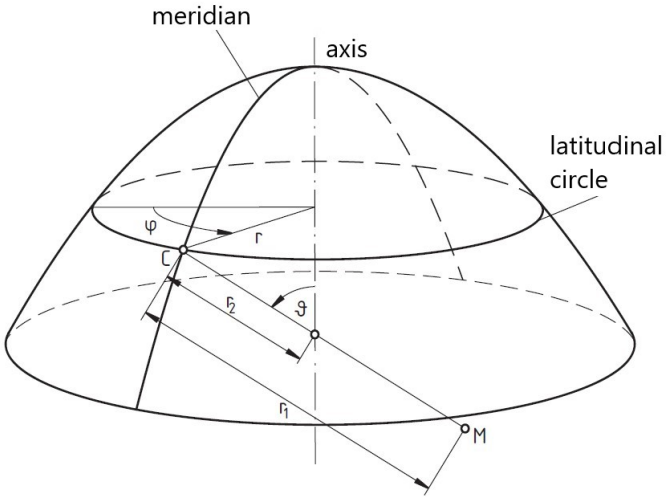


Figure 5.9. Designations on the rotation shell. Image source: [7]

Designations:

- r radius of the latitudinal circle
- r_1 radius of curvature of the meridian curve
- r_2 radius of curvature of the latitudinal circle

For the cylindrical shell one obtains:

$$\begin{aligned} r_1 &\to \infty \\ r_2 = r &= a \end{aligned} \tag{5.19}$$

where a is the distance between the axis of symmetry of the cylinder and the lateral surface, considering the inner part of it (i.e., neglecting the thickness of the shell itself).

5.2.1 Membrane theory for rotational shells

Within the framework of the membrane theory, three internal forces occur in the shell, Eq. 5.18. Three equilibrium conditions are available so that these forces can be calculated without considering deformations. The task is therefore statically determined.[31] In order to be able to establish the elastomechanical relations, an infinitesimal element is cut out of the shell skin, which is bounded by two neighbouring meridians and two neighbouring breaker circles, Fig. 5.10.
The basic equations or the elastomechanical relations describe the equilibrium of forces, the distortion-displacement relations and Hooke's law. Combined, these result in the differential equations, whose solution is provided by the stresses and deformations in the shell. If the deformations are not hindered, the equations of the equilibrium of forces alone provide the stress on the shell.[18]

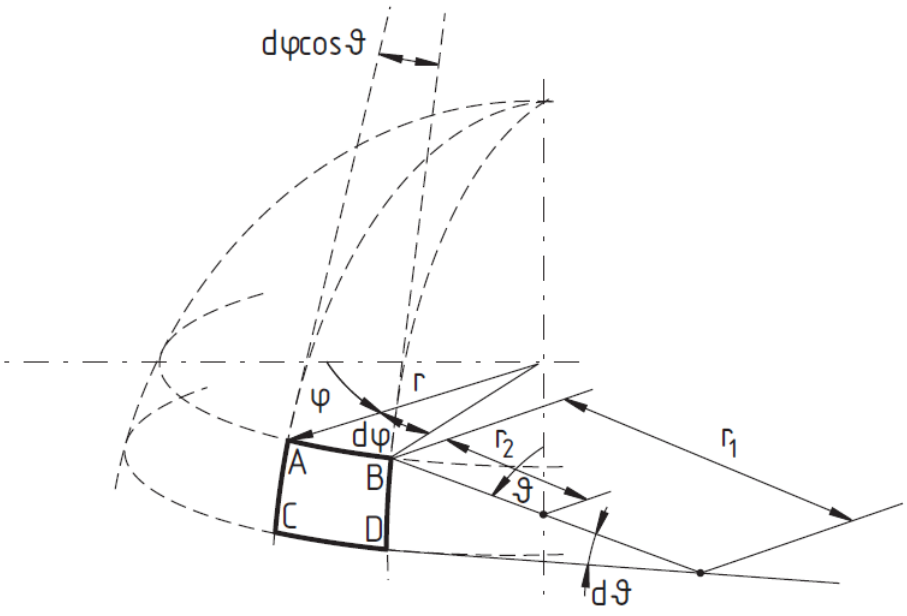


Figure 5.10. Geometric relationships on the shell element. Image source: [18]

Fig. 5.11 shows the internal and external forces that can act on the shell element. If the forces are counted positively in the positive coordinate direction, one obtains:

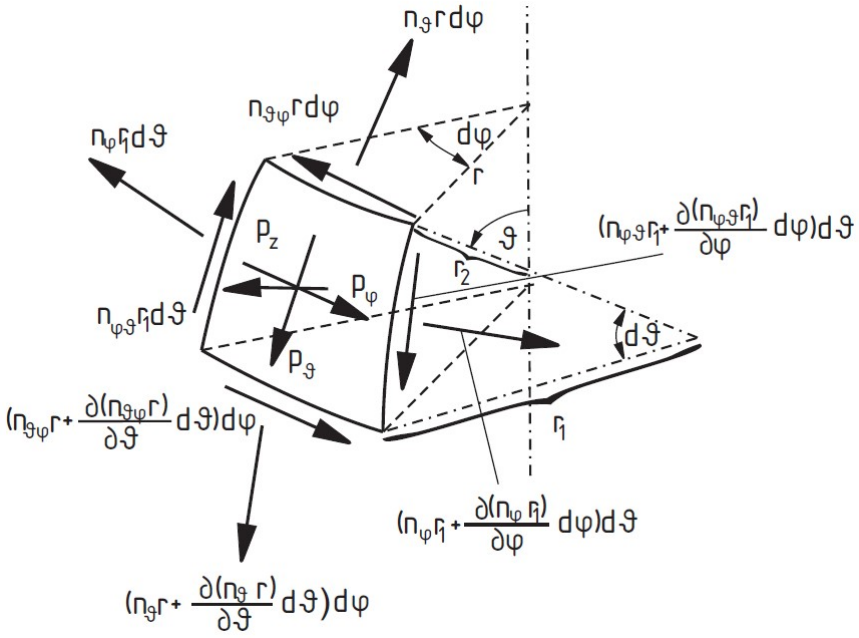


Figure 5.11. Forces on the shell element. Image source: [18]

Equilibrium in the direction of the meridian tangent:

$$\begin{aligned}
&\left(n_\vartheta r + \tfrac{\partial(n_\vartheta r)}{\partial\vartheta}\mathrm{d}\vartheta\right)\mathrm{d}\varphi - n_\vartheta r d\varphi + \\
&+\left(n_{\varphi\vartheta} r_1 + \tfrac{\partial(n_{\varphi\vartheta} r_1)}{\partial\varphi}\mathrm{d}\varphi\right)\mathrm{d}\vartheta - n_{\varphi\vartheta} r_1\ \mathrm{d}\vartheta + \\
&-n_\varphi r_1\ \mathrm{d}\vartheta\mathrm{d}\varphi\cos\vartheta + p_\vartheta(\vartheta,\varphi) r d\varphi r_1 d\vartheta = 0
\end{aligned} \tag{5.20}$$

Equilibrium in the direction of the latitudinal tangent:

$$\begin{aligned}
&\left(n_\varphi r_1 + \tfrac{\partial(n_\varphi r_1)}{\partial\varphi}\mathrm{d}\varphi\right)\mathrm{d}\vartheta - n_\varphi r_1\ \mathrm{d}\vartheta + \\
&+\left(n_{\vartheta\varphi} r + \tfrac{\partial(n_{\vartheta\varphi} r)}{\partial\vartheta}\mathrm{d}\vartheta\right)\mathrm{d}\varphi - n_{\vartheta\varphi} r\ \mathrm{d}\varphi + \\
&+n_{\varphi\vartheta} r_1\ \mathrm{d}\vartheta\mathrm{d}\varphi\cos\vartheta + p_\varphi(\vartheta,\varphi) r d\varphi r_1 d\vartheta = 0
\end{aligned} \tag{5.21}$$

Equilibrium in the direction of the normals:

$$-n_\varphi r_1\ \mathrm{d}\vartheta\mathrm{d}\varphi\sin\vartheta - n_\vartheta r\ \mathrm{d}\varphi\mathrm{d}\vartheta + p_z(\vartheta,\varphi) r d\varphi r_1 d\vartheta = 0 \tag{5.22}$$

After dividing Eq. 5.20 5.21 5.22 by $d\varphi\ d\vartheta$ and introducing the radius r_2:

$$r_2 = \frac{r}{\sin\vartheta} \tag{5.23}$$

one finally gets:

$$\begin{aligned}
\frac{\partial(n_\vartheta r)}{\partial\vartheta} + \frac{\partial\left(n_{\varphi\vartheta} r_1\right)}{\partial\varphi} - n_\varphi r_1 \cos\vartheta + p_\vartheta(\vartheta,\varphi) r r_1 = 0 \\
\frac{\partial(n_\varphi r_1)}{\partial\varphi} + \frac{\partial\left(n_{\theta\varphi} r\right)}{\partial\vartheta} + n_{\varphi\vartheta} r_1 \cos\vartheta + p_\varphi(\vartheta,\varphi) r r_1 = 0 \\
n_\varphi r_1 + n_\theta r_2 - p_z(\vartheta,\varphi) r_1 r_2 = 0
\end{aligned} \tag{5.24}$$

For the cases mentioned above, i.e. in case of straight meridians or meridian curves with a point of inflection, in which the meridian arc length s is introduced instead of the angle ϑ, the result is given by:

$$\mathrm{d}s = r_1 \, \mathrm{d}\vartheta \tag{5.25}$$

then:

$$\begin{aligned}
\frac{\partial\left(n_s r\right)}{\partial s} + \frac{\partial n_{\varphi s}}{\partial\varphi} - n_\varphi \cos\vartheta + p_s(s,\varphi) r = 0 \\
\frac{\partial\left(n_{s\varphi} r\right)}{\partial s} + \frac{\partial n_\varphi}{\partial\varphi} + n_{\varphi s} \cos\vartheta + p_\varphi(s,\varphi) r = 0 \\
n_\varphi r_1 + n_s r_2 - p_z(s,\varphi) r_1 r_2 = 0
\end{aligned} \tag{5.26}$$

The three unknown internal forces can be calculated from Eq. 5.24 or Eq. 5.26. In each case the 3rd equation is used to eliminate an unknown from the first two equations. This results in a system of two coupled partial differential equations for two unknown forces, whose integration provides the solution of the membrane theory. The important special case of the cylindrical shell is presented below.
For the cylindrical shell, the longitudinal coordinate x is introduced. Furthermore, the following applies: $r = r_2 = a$. Thus follows:

$$\begin{aligned}
\frac{\partial n_x}{\partial x} + \frac{\partial n_{\varphi x}}{a\partial\varphi} + p_x(x,\varphi) = 0 \\
\frac{\partial n_{x\varphi}}{\partial x} + \frac{\partial n_\varphi}{a\partial\varphi} + p_\varphi(x,\varphi) = 0 \\
n_\varphi - p_z(x,\varphi) a = 0
\end{aligned} \tag{5.27}$$

The central surface of the membrane shell experiences distortions in the meridian direction ε_ϑ and in the circumferential direction ε_φ as well as an angular change (shear distortion) $\gamma_{\vartheta\varphi}$. These depend on the deformations of the shell skin in the direction of the coordinates. In contrast to the flat skin in Cartesian coordinates, the geometric relationships of curved or doubly curved shells become relatively complex. In analytical shell theory, the resulting complex distortion-displacement relationships are determined by means of tensor calculus. The corresponding relations

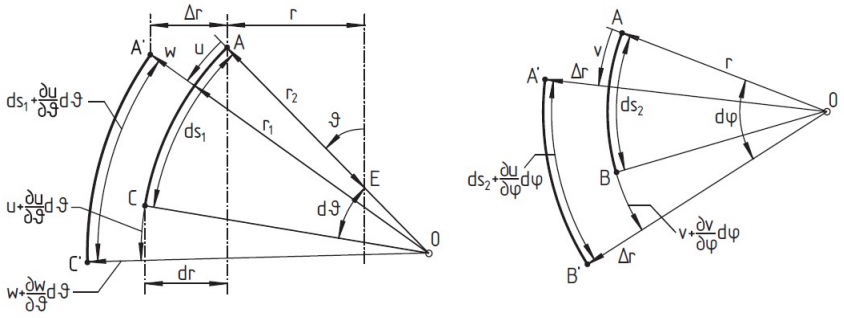


Figure 5.12. Deformation of (a) meridian element $\overline{AC}$ and (b) latitudinal element $\overline{AB}$. Image source: [18]

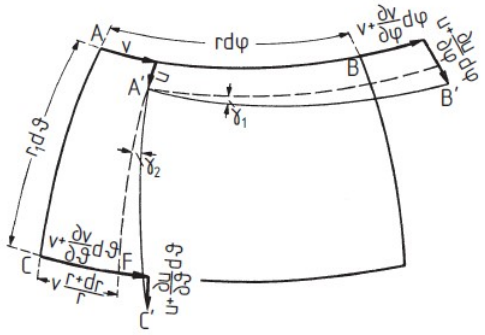


Figure 5.13. Shear distortion of the shell element. Image source: [18]

are determined here by means of geometric considerations. The displacements in the direction of the ϑ, φ, z coordinates are denoted by u, v, w. [18]

The distortion in meridian direction ε results from the change in length of an element related to the initial length. For ε_ϑ , the designations from Fig. 5.12a result:

$$\varepsilon_\theta = \frac{\overline{A'C'} - \overline{AC}}{\overline{AC}} \tag{5.28}$$

with:

$$\begin{aligned} \overline{AC} &= \mathrm{d}s_1 \\ \overline{A'C'} &= \left(\mathrm{d}s_1 + \frac{\partial u}{\partial \theta}\mathrm{d}\vartheta\right)\frac{r_1 + w}{r_1} \end{aligned} \tag{5.29}$$

one gets:

$$\varepsilon_\theta = \frac{1}{r_1}\left(\frac{\partial u}{\partial \vartheta} + w\right) \tag{5.30}$$

Distortion in circumferential direction, ε_φ, the designations from Fig. 5.12b

results:

$$\varepsilon_\varphi = \frac{\overline{A'B'} - \overline{AB}}{\overline{AB}} \tag{5.31}$$

with:

$$\begin{aligned} \overline{AB} &= \mathrm{d}s_2 \\ \overline{A'B'} &= \left(\mathrm{d}s_2 + \frac{\partial v}{\partial \varphi}\mathrm{d}\varphi\right)\frac{r+\Delta r}{r} \end{aligned} \tag{5.32}$$

and:

$$\Delta r = (u\cos\theta + w\sin\theta) \tag{5.33}$$

results:

$$\varepsilon_\varphi = \frac{1}{r}\left(\frac{\partial v}{\partial \varphi} + u\cos\theta + w\sin\theta\right) \tag{5.34}$$

For the shear distortion of the shell element, $\gamma_{\vartheta\varphi}$, the designations from Fig. 5.13 results:

$$\gamma_{\vartheta\varphi} = \gamma_1 + \gamma_2 \tag{5.35}$$

with:

$$\begin{aligned} \gamma_1 &= \frac{\frac{\partial u}{\partial \varphi}\mathrm{d}\varphi}{r\ \mathrm{d}\varphi + \frac{\partial v}{\partial \varphi}d\varphi} \approx \frac{1}{r}\frac{\partial u}{\partial \varphi} \\ \cos\theta &= \frac{\mathrm{d}r}{r_1\ \mathrm{d}\theta} \end{aligned} \tag{5.36}$$

it becomes:

$$\gamma_2 = \frac{1}{r_1}\frac{\partial v}{\partial \theta} - \frac{v}{r}\cos\theta \tag{5.37}$$

this results in:

$$\gamma_{\vartheta\varphi} = \frac{1}{r_1}\frac{\partial v}{\partial \theta} + \frac{1}{r}\left(\frac{\partial u}{\partial \varphi} - v\cos\theta\right) \tag{5.38}$$

The following applies in the s, φ, z coordinate system:

$$\varepsilon_s = \frac{\partial u}{\partial s} + \frac{w}{r_1}$$

$$\varepsilon_\varphi = \frac{1}{r}\left(\frac{\partial v}{\partial \varphi} + u\cos\theta + w\sin\theta\right) \tag{5.39}$$

$$\gamma_{s\varphi} = \frac{\partial v}{\partial s} + \frac{1}{r}\left(\frac{\partial u}{\partial \varphi} - v\cos\theta\right)$$

The distortion-displacement relationships (kinematic relationships) for the cylindrical shell are given below, with the longitudinal coordinate x and $r = r_2 = a$ follows: [30]

$$\varepsilon_x = \frac{\partial u}{\partial x}$$

$$\varepsilon_\varphi = \frac{1}{a}\left(\frac{\partial v}{\partial \varphi} + w\right) \tag{5.40}$$

$$\gamma_{x\varphi} = \frac{\partial v}{\partial x} + \frac{1}{a}\left(\frac{\partial u}{\partial \varphi}\right)$$

5.2.2 Rotationally symmetrical load and pressure vessels

If the loads are rotationally symmetrical, the internal forces and the distortions do not change with the circumferential angle φ, so that the partial derivatives according to φ become zero. This simplifies the differential equations of the equilibrium weight, Eq. 5.24 to:

$$\begin{aligned} \frac{\partial\,(n_\vartheta r)}{\partial \vartheta} - n_\varphi r_1 \cos\vartheta + p_\vartheta(\vartheta) r r_1 &= 0 \\ \frac{\partial\,(n_{\vartheta\varphi} r)}{\partial \vartheta} + n_{\varphi\vartheta} r_1 \cos\vartheta + p_\varphi(\vartheta) r r_1 &= 0 \\ n_\varphi r_1 + n_\vartheta r_2 - p_z(\vartheta) r_1 r_2 &= 0 \end{aligned} \tag{5.41}$$

or Eq. 5.26 in the s, φ, z coordinate system:

$$\begin{aligned} \frac{\partial\,(n_s r)}{\partial s} - n_\varphi \cos\vartheta + p_s(s) r &= 0 \\ \frac{\partial\,(n_{s\varphi} r)}{\partial s} + n_{\varphi s} \cos\vartheta + p_\varphi(s) r &= 0 \\ n_\varphi r_1 + n_s r_2 - p_z(s) r_1 r_2 &= 0 \end{aligned} \tag{5.42}$$

The new equilibrium relations shows that the middle equation for the equilibrium in circumferential direction is decoupled from the other two and thus the shear flow $n_{\vartheta\varphi}$ depends only on the load p_φ. The kinematic relations Eq. 5.30, Eq. 5.34 and Eq. 5.38 simplify to:

$$\begin{aligned}
\varepsilon_\theta &= \frac{1}{r_1}\left(\frac{\partial u}{\partial \theta} + w\right) \\
\varepsilon_\varphi &= \frac{1}{r}(u\cos\theta + w\sin\theta) \\
\gamma_{\theta\varphi} &= \frac{1}{r_1}\frac{\partial v}{\partial \theta} - \frac{v\cos\theta}{r}
\end{aligned} \tag{5.43}$$

or Eq. 5.39 in the s, φ, z coordinate system:

$$\begin{aligned}
\varepsilon_s &= \frac{\partial u}{\partial s} + \frac{w}{r_1} \\
\varepsilon_\varphi &= \frac{1}{r}(u\cos\theta + w\sin\theta) \\
\gamma_{s\varphi} &= \frac{\partial v}{\partial s} - \frac{v}{r}\cos\theta
\end{aligned} \tag{5.44}$$

For the pressure vessels, the above equations are simplified. There is now only a radial load that is constant in both the meridian and circumferential directions:

$$p_z(\theta, \varphi) = p_0 \tag{5.45}$$

This means that there are no thrust flows and also no thrust distortions, and one obtains:

$$\begin{aligned}
\frac{\partial\,(n_t r)}{\partial \theta} - n_\varphi r_1 \cos\theta &= 0 \\
n_\varphi r_1 + n_\theta r_2 - p_0 r_1 r_2 &= 0
\end{aligned} \tag{5.46}$$

or in the s, φ, z coordinate system:

$$\begin{aligned}
\frac{\partial\,(n_s r)}{\partial s} - n_\varphi \cos\theta &= 0 \\
n_\varphi r_1 + n_s r_2 - p_0 r_1 r_2 &= 0
\end{aligned} \tag{5.47}$$

and:

$$\begin{aligned} \varepsilon_\theta &= \frac{1}{r_1}\left(\frac{\partial u}{\partial \theta} + w\right) \\ \varepsilon_\varphi &= \frac{1}{r}(u\cos\theta + w\sin\theta) \end{aligned} \tag{5.48}$$

respectively:

$$\begin{aligned} \varepsilon_s &= \frac{\partial u}{\partial s} + \frac{w}{r_1} \\ \varepsilon_\varphi &= \frac{1}{r}(u\cos\theta + w\sin\theta) \end{aligned} \tag{5.49}$$

The sphere is the most favourable vessel shape for internal pressure and one obtains in each point of the shell the same force flows in sphere directions. A constant wall thickness results in a pure membrane stress state with a constant stress in all directions. Because of the geometric conditions, the spherical shape cannot always be realised. Examples are:

- gas cylinders,
- tanks of launch vehicles,
- pressurised cabin of a commercial aircraft.

In these cases, the circular cylindrical shell with special bottom shapes is often used. In the following, the cylinder shell vessel is examined. In the cylinder shell the following force flows result from Eq. 5.27:

$$\frac{\partial n_x}{\partial x} = 0 \qquad\qquad n_\varphi = p_0\ a \tag{5.50}$$

The force flow on the circumference n_{x0} of the lid force from the internal pressure must maintain equilibrium. The lid force is non-depending on the shape of the lid or base:

$$\begin{aligned} P_D = p_0\ a^2\ \pi &= n_{x0}\ 2\ \pi\ a \\ n_{x0} &= \frac{p_0\ a}{2} \end{aligned} \tag{5.51}$$

For the cylindrical shell of a pressure vessel with constant wall thickness, a membrane stress state with constant longitudinal stress σ_x and constant circumferential stress $\sigma\varphi$ is obtained. The circumferential stress is twice as large as the longitudinal stress:

$$\sigma_x = \frac{p_0 a}{2h} \qquad\qquad \sigma_\varphi = \frac{p_0 a}{h} \tag{5.52}$$

these two last equations are called boiler formulas.

This membrane stress state only occurs if the cylinder shell can deform unhindered (which is not the case at the ends because of the covers). The unhindered deformation would result for the length change:

$$\begin{aligned} \varepsilon_x = \frac{\partial u}{\partial x} = \frac{n_x}{Eh} - \frac{v n_\varphi}{Eh} = \frac{p_0 a}{2Eh}(1-2v) \\ \Rightarrow \Delta l = \int_0^l \varepsilon_x \, \mathrm{d}x = \frac{p_0 a}{2Eh}(1-2v)l \end{aligned} \tag{5.53}$$

and for the radius change:

$$\begin{aligned} \varepsilon_\varphi = \frac{w}{a} = \frac{-v n_x}{Eh} + \frac{n_\varphi}{Eh} \\ w = \frac{p_0 a^2}{Eh}\left(1 - \frac{1}{2}v\right) \end{aligned} \tag{5.54}$$

At the transition from the cylindrical part to the bottom (or lid), deviations from the from the membrane state occur. Their size depends on the shape of the bottom. Even at the smooth transition to the hemispherical bottom (but unsteady curvature change), the membrane stress state is disturbed, as both shells experience different radius changes. These must be compensated in the transition area by bending moments and transverse forces. A disturbance of the membrane stress state can be avoided by increasing the wall thickness in the cylindrical part in order to obtain the same change in radius there as in the spherical shell.[18]

5.3 The vessel theory

The analysis of pressure vessels with the membrane theory has shown that in general the membrane stress state is disturbed at the edges or at the transitions to the vessel bottom. In order to describe the shell with bending disturbances, the bending theory is used. A simple special case of this theory is the vessel theory, which is valid for rotationally symmetrical shells with a rotationally symmetrical load normal to the shell skin. The vessel theory then can be applied on cylindrical shells of revolution with constant wall thickness.[31] The solutions developed for this can also be transferred to spherical shells.
A circular cylinder shell is considered, which is firmly clamped upright at one end and is filled with a liquid, Fig. 5.14. The liquid filling creates a rotationally symmetrical, hydrostatic pressure that increases linearly from top to bottom:

$$p_z = \gamma(H - x)$$

The membrane theory provides a circumferential force flow for this loading condition:

$$n_\varphi = p_z \; a = \gamma(H - x)a$$

and a radial expansion results:

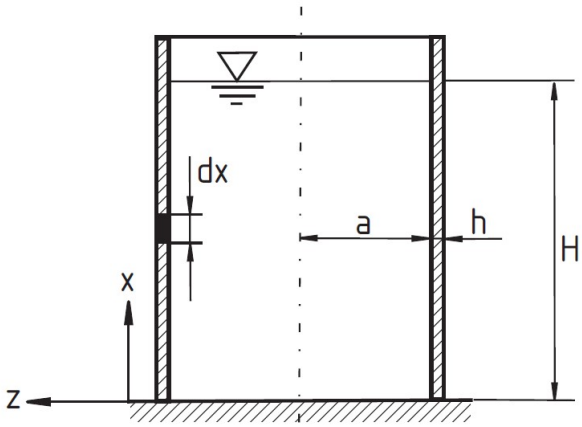


Figure 5.14. Standing vessel with liquid filling . Image source: [18]

$$w = \frac{p_z a^2}{Eh} = \frac{\gamma(H-x)a^2}{Eh} \tag{5.55}$$

If one considers an infinitesimal element, the element from the shell skin (Figure 3.2), the membrane force flows n_x and $n\varphi$, the bending moment flows m_x and m_φ, and the shear force flow q_x can act there. For symmetrical reasons, the shear force fluxes $n_{x\varphi}$ and $n_{\varphi x}$, the torsional moment fluxes $m_{x\varphi}$ and $m_{\varphi x}$, and the shear force fluxes q_φ are not possible, i.e. they are equal to zero. Except- n_φ and m_φ are constant on the circumference. According to Eq. 5.14, Eq. 5.15, Eq. 5.16, with $z/a << 1$:

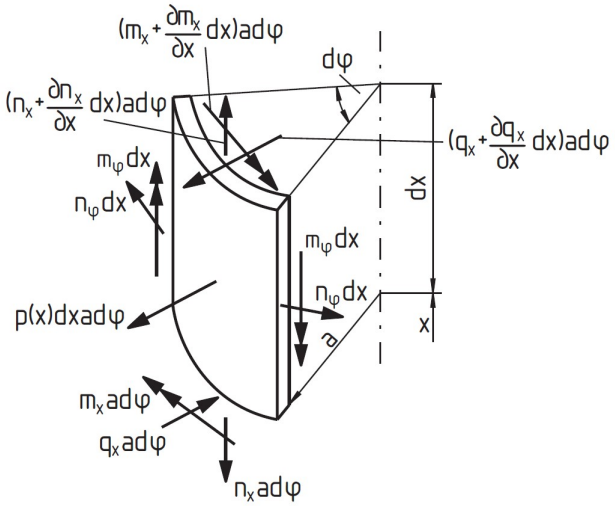


Figure 5.15. vessel theory: forces on the infinitesimal shell element. Image source: [18]

$$n_x = \int_{-h/2}^{h/2} \sigma_x \left(1 + \frac{z}{a}\right) \mathrm{d}z \approx \int_{-h/2}^{h/2} \sigma_x \, \mathrm{d}z$$

$$q_x = \int_{-h/2}^{h/2} \tau_{xz} \left(1 + \frac{z}{a}\right) \mathrm{d}z \approx \int_{-h/2}^{h/2} \tau_{xz} \, \mathrm{d}z$$

$$m_x = -\int_{-h/2}^{h/2} \sigma_x \left(1 + \frac{z}{a}\right) z \, \mathrm{d}z \approx -\int_{-h/2}^{h/2} \sigma_x z \, \mathrm{d}z \tag{5.56}$$

$$n_\varphi = \int_{-h/2}^{h/2} \sigma_\varphi \mathrm{d}z$$

$$m_\varphi = -\int_{-h/2} \sigma_\varphi z \, \mathrm{d}z$$

Equilibrium provides:

$$\begin{aligned} \frac{\partial n_x}{\partial x} &= 0 \\ a\frac{\partial q_x}{\partial x} - n_\varphi + p(x)\ a &= 0 \\ \frac{\partial m_x}{\partial x} + q_x &= 0 \\ \frac{\partial m_\varphi}{\partial \varphi} &= 0 \end{aligned} \tag{5.57}$$

In case of open vessel it follows $n_x = 0$, and the second and third equations of Eq. 5.57 provide after eliminating the shear force:

$$a\frac{\partial^2 m_x}{\partial x^2} + n_\varphi - p(x)\ a = 0 \tag{5.58}$$

Hooke's law provides for isotropic material:

$$\begin{aligned} \sigma_x(z) &= \frac{E}{1 - v^2} \left(\varepsilon_x + v\varepsilon_\varphi\right) \\ &= \frac{E}{1 - v^2} \left[\varepsilon_{x_0} + v\varepsilon_{\varphi_0} - z\left(\kappa_x + v\kappa_\varphi\right)\right] \end{aligned}$$

$$\sigma_\varphi(z) = \frac{E}{1 - v^2} \left(\varepsilon_\varphi + v\varepsilon_x\right) = \frac{E}{1 - v^2} \left[\varepsilon_{\varphi_0} + v\mathcal{E}_{x_0} - z\left(\kappa_\varphi + v\kappa_x\right)\right]$$

The cutting forces become:

$$\begin{aligned} n_x &= \int_{-h/2}^{h/2} \sigma_x \, \mathrm{d}z = \frac{Eh}{1-v^2} \left(\varepsilon_{x_0} + v\varepsilon_{\varphi_0}\right) \\ &= D\left(\varepsilon_{x_0} + v\varepsilon_{\varphi_0}\right) = D\left(\frac{\partial u_0}{\partial x} + v\frac{w}{a}\right) \end{aligned}$$

$$\begin{aligned} n_\varphi &= D\left(\varepsilon_{\varphi_0} + v\varepsilon_{x_0}\right) = D\left(\frac{w}{a} + v\frac{\partial u_0}{\partial x}\right) \\ m_x &= \frac{Eh^3}{12\left(1-v^2\right)} \left(\kappa_x + v\kappa_\varphi\right) = K\left(\frac{\partial^2 w}{\partial x^2} + v\frac{w}{a^2}\right) \\ m_\varphi &= \frac{Eh^3}{12\left(1-v^2\right)} \left(\kappa_\varphi + v\kappa_x\right) = K\left(\frac{w}{a^2} + v\frac{\partial^2 w}{\partial x^2}\right) \end{aligned} \tag{5.59}$$

In the case that $n_x = 0$, it follows from the first equation of Eq. 5.59:

$$\frac{\partial u_0}{\partial x} = -v\frac{w}{a} \tag{5.60}$$

and thus from the second equation of Eq. 5.59:

$$n_\varphi = \frac{Eh}{1-v^2}\frac{w}{a}\left(1-v^2\right) = \frac{Eh}{a}w \tag{5.61}$$

If, in addition, $\nu \, \frac{w}{a^2}$ is neglected, from the third equation of Eq. 5.59:

$$m_x = K\frac{\partial^2 w}{\partial x^2} \tag{5.62}$$

Thus in Eq. 5.58 the moment m_x and the force flow n_φ can be expressed by the deformation w and the following equation is obtained:

$$\frac{\partial^4 w}{\partial x^4} + \frac{12\left(1-v^2\right)}{a^2h^2}w = \frac{p(x)}{K} \tag{5.63}$$

Other important parameters are K, ξ, α and β. K is the bending stiffness of the isotropic shell skin with wall thickness h:

$$K = \frac{Eh^3}{12(1-\nu^2)} \tag{5.64}$$

ξ is the dimensionless longitudinal coordinate:

$$\xi = \frac{x}{a} \tag{5.65}$$

α is the shell parameter:

$$\alpha = \sqrt{\frac{a}{h}} \sqrt[4]{3(1-\nu^2)} \tag{5.66}$$

β is the opening angle at the edge of the shell:

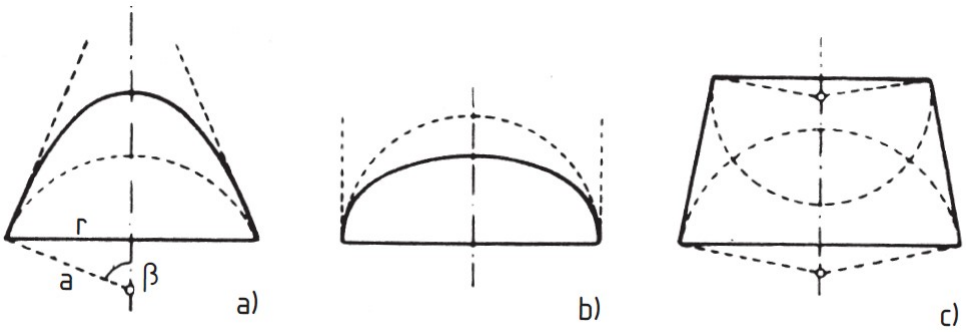


Figure 5.16. Opening angle at the edge of the shell. Image source: [18]

Therefore, Eq. 5.63 becomes:

$$\frac{\partial^4 w}{\partial \xi^4} + 4\alpha^4 w = \frac{p(\xi)a^4}{K} \tag{5.67}$$

This fourth-order differential equation with constant coefficients is the vessel equation, the solution of which can be formally expressed as:

$$w(x) = w_h(x) + w_p(x) \tag{5.68}$$

The homogeneous solution w_h describes the disturbance of the membrane state due to the boundary conditions of the edges; the particulate solution w_p describes the radial deformation of the membrane solution.

5.4 Calculation with the transfer matrix method

The calculations carried out analytically lead to a complex set of formulas and thus to time-consuming calculation steps. Schematising the calculation process by formulating matrices is possible and thus make it accessible for calculation with the computer.[26] [28] For questions whose answer is found by solving the differential equations, the transfer matrix method is suitable. If the vector of all quantities that completely describe the state of an edge i is called the state vector $\vec{U}_i$, then the following applies with the transfer matrix $\{W\}_{i,i+1}$ and a load vector $\vec{L}$ for the state vector $\vec{U}_{i+1}$ of the second edge:[18]

$$\vec{U}_{i+1} = \{W\}_{i,i+1}\vec{U}_i + \vec{L} \tag{5.69}$$

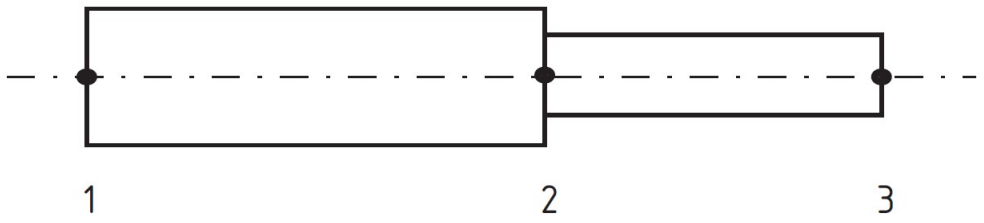


Figure 5.17. Structure consisting of two sections. Image source: [18]

The state vector of an nth-order problem consists of n state variables. To formalise the calculation process, the load vector can be integrated into the transfer matrix as the $n+1th$ vector. With the new transfer matrix $\{W_L\}_{i,i+1}$ then applies:

$$\vec{U}_{i+1} = \{W_L\}_{i,i+1}\,\vec{U}_i \tag{5.70}$$

If the structure consists of several sections, the calculation can be carried out section by section:

$$\begin{aligned} \vec{U}_2 &= \{W_L\}_{1,2}\,\vec{U}_1 \\ \vec{U}_3 &= \{W_L\}_{2,3}\,\vec{U}_2 \end{aligned} \tag{5.71}$$

These equations can either be solved simultaneously or the inner state vectors can first be eliminated by an intermediate step:

$$\vec{U}_3 = \{W_L\}_{2,3}\,\{W_L\}_{1,2}\,\vec{U}_1 \tag{5.72}$$

With the outer state vectors now known, the inner state vectors can be calculated by the initial equations.

The transfer matrix of the cylindrical shell according to the vessel theory, leads to a 4th order differential equation, so that the state of a vessel edge is clearly described by four state variables. These are for the edge with coordinate x_0:

$$w(x_0), \qquad w'(x_0), \qquad m_x(x_0), \qquad q_x(x_0)$$

If one uses the homogeneous solution of the equation with the hyperbolic functions, one obtains, if the boundary conditions at the boundary x_0 are assumed to be known, four equations for the state variables of the right boundary as a function of the state variables of the left boundary and the load. With the state vector $\vec{U}_i$:

$$\vec{U}_i = \begin{bmatrix} w_i \\ w_i' \\ m_{x_i} \\ q_{x_i} \end{bmatrix}$$

and the abbreviations:

$$\begin{aligned} \operatorname{sh} s &= \sinh\left(\frac{\alpha l}{a}\right) \sin\left(\frac{\alpha l}{a}\right) \\ \operatorname{sh} c &= \sinh\left(\frac{\alpha l}{a}\right) \cos\left(\frac{\alpha l}{a}\right) \\ \operatorname{ch} s &= \cosh\left(\frac{\alpha l}{a}\right) \sin\left(\frac{\alpha l}{a}\right) \\ \operatorname{ch} c &= \cosh\left(\frac{\alpha l}{a}\right) \cos\left(\frac{\alpha l}{a}\right) \end{aligned}$$

these equations yield the following transfer matrix W_{CTH}:

$$W_{CTH} = \begin{Bmatrix} \operatorname{ch} c & \frac{1}{2\alpha}(\operatorname{ch} s + \operatorname{sh} c) & \frac{1}{2\alpha^2 K} \operatorname{sh} s & \frac{1}{4\alpha^3 K}(\operatorname{ch} s - \operatorname{sh} c) \\ -\alpha(\operatorname{chs} - \operatorname{sh} c) & \operatorname{ch} c & \frac{1}{2\alpha K}(\operatorname{ch} s + \operatorname{sh} c) & \frac{1}{2\alpha^2 K} \operatorname{sh} s \\ -2\alpha^2 K \operatorname{sh} s & -\alpha K(\operatorname{ch} s - \operatorname{sh} c) & \operatorname{ch} c & \frac{1}{2\alpha}(\operatorname{ch} s + \operatorname{sh} c) \\ -2\alpha^3 K(\operatorname{ch} s + \operatorname{sh} c) & -2\alpha^2 K \operatorname{sh} s & -\alpha(\operatorname{ch} s - \operatorname{sh} c) & \operatorname{ch} c \end{Bmatrix}$$

and the load vector $\vec{L}_{CTH}$ in the case of a constant internal pressure p_0:

$$\vec{L}_{CTH} = \begin{bmatrix} -\frac{p_0}{\beta}(1 - \operatorname{ch} c) \\ -\frac{p_0}{\beta}\alpha(\operatorname{chs} - \operatorname{sh} c) \\ -\frac{p_0}{\beta} 2\alpha^2 K \operatorname{sh} s \\ -\frac{p_0}{\beta} 2\alpha^3 K(\operatorname{ch} s + \operatorname{sh} c) \end{bmatrix}$$

where CTH stands for (Cylindrical) vessel Theory.

Frames in the shell structure can also be analysed using the transfer matrix method. For this purpose, a frame matrix is formulated which describes the elastic behaviour of the rib. The frame deforms due to radial loads Q_{Fr} and carding moments M_{Fr}.

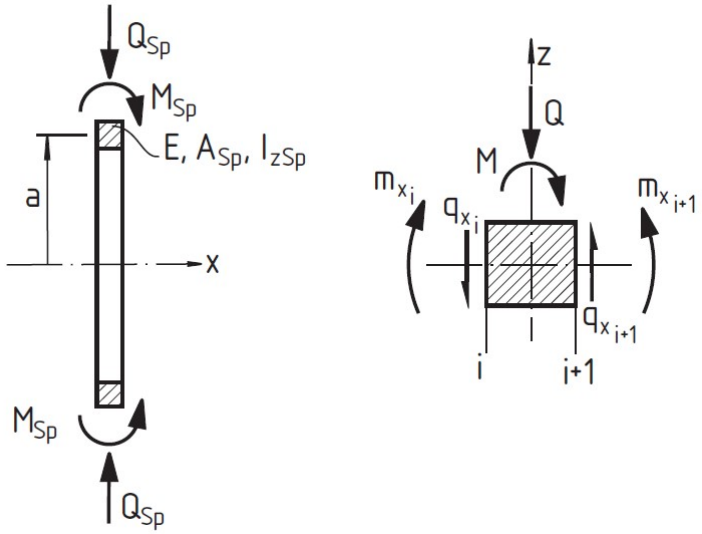


Figure 5.18. Loads on the frame. Image source: [18]

If shells are connected to the frame on the left and right and the frame is loaded by loads Q and M from the outside, the following applies, as in Fig. 5.18:

$$\begin{aligned} Q_{Fr} &= Q + q_{x_i} - q_{x_{i+1}} \\ M_{Fr} &= M + m_{x_i} - m_{x_{i+1}} \end{aligned} \tag{5.73}$$

If it is further assumed that the deformation over the frame does not change, the following applies:

$$\begin{aligned} w_i &= w_{i+1} = w_{Fr} \\ w_i' &= w_{i+1}' = -\psi_{Fr} \end{aligned} \tag{5.74}$$

The last four equations give the frame matrix Fr and the load vector L_{Fr}:

$$\{Fr\} = \begin{Bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \frac{EI_{z_{Fr}}}{a^2} & 1 & 0 \\ \frac{EA_{Fr}}{a^2} & 0 & 0 & 1 \end{Bmatrix} \tag{5.75}$$

$$\vec{L}_{Fr} = \begin{bmatrix} 0 \\ 0 \\ M \\ Q \end{bmatrix} \tag{5.76}$$

5.5 Composite shells under any load

In the previous section Sec. 5.3, the partial differential equations for the rotationally symmetric shear structure could be transferred into ordinary differential equations and solved because of the rotationally symmetric load. In the following, any φ-dependent loads are considered, which also lead to φ-dependent deformations and internal forces. The differential equations initially remain partial. With the help of the Fourier series development, the external loads, the internal forces and the deformations are developed into periodic approach functions, so that the derivatives according to φ can be replaced by algebraic expressions.[29] This makes it possible to convert the partial differential equations into ordinary ones. Composite and, if necessary, tension-stiffened shell structures are broken down into elements, for each of which transfer matrices or tension matrices are developed, with the help of which the individual structural sections can be linked.[15]

First, the transfer matrices for the isotropic cylindrical shells are determined according to the membrane theory by solving the differential equations from section Sec. 5.2.1. Then, the equilibrium relationships of Eq. 5.26 for the membrane cylinder shell are compiled again here. Only the normal load p_z on the shell skin is considered. In order to further simplify the solution of differential equations, it is assumed that this load is not dependent on the coordinate x, but only on φ for the considered shell sections.[18]

$$\begin{aligned} \frac{\partial n_x}{\partial x} + \frac{\partial n_{\varphi x}}{a\,\partial \varphi} &= 0 \\ \frac{\partial n_{x\varphi}}{\partial x} + \frac{\partial n_{\varphi}}{a\,\partial \varphi} &= 0 \\ n_{\varphi} - p_z(\varphi) a &= 0 \end{aligned} \tag{5.77}$$

The kinematic relations for the cylindrical shell are:

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} \\ \varepsilon_{\varphi} &= \frac{1}{a}\left(\frac{\partial v}{\partial \varphi} + w\right) \\ \gamma_{x\varphi} &= \frac{\partial v}{\partial x} + \frac{1}{a}\frac{\partial u}{\partial \varphi} \end{aligned} \tag{5.78}$$

Hooke's law for the isotropic cylindrical shell is:

$$\begin{aligned}\varepsilon_x &= \frac{1}{Eh}\left(n_x - v\ n_\varphi\right)\\ \varepsilon_\varphi &= \frac{1}{Eh}\left(-v\ n_x + n_\varphi\right)\\ \gamma_{x\varphi} &= \frac{1}{Gh} n_{x\varphi}\end{aligned} \tag{5.79}$$

From Eq. 5.77 to Eq. 5.79, four relations can be found for the first derivatives with respect to x of the quantities $n_{x\varphi}$, n_x , u and v, the state variables as a function of these quantities. From the second and third equations of Eq. 5.77 follows:

$$\frac{\partial n_{x\varphi}}{\partial x} = -\frac{\partial p_z}{\partial \varphi} \tag{5.80}$$

From the 1st equation of Eq. 5.77 results:

$$\frac{\partial n_x}{\partial x} = -\frac{\partial n_{\varphi x}}{a\ \partial \varphi} \tag{5.81}$$

from the 1st of Eq. 5.78, the 1st of Eq. 5.79 and the 3rd of Eq. 5.77:

$$\frac{\partial u}{\partial x} = \frac{1}{Eh} n_x - \frac{v\ a}{Eh} p_z \tag{5.82}$$

Finally, it follows from the 3rd of Eq. 5.78 and the 3rd of Eq. 5.79:

$$\frac{\partial v}{\partial x} = -\frac{1}{a}\frac{\partial u}{\partial \varphi} + \frac{1}{Gh} n_{x\varphi} \tag{5.83}$$

The last four equations presented are transformed into ordinary differential equations using Fourier series expansions for the load and the state variables. Stresses in the plane of the shell sections are n_{x_φ} and stresses perpendicular to shell section plane are n_x, thus in the direction of the length of the beam. If only symmetrical loads are assumed, one gets:

$$\begin{aligned}p_z(\varphi) &= p_{z_0} + \sum_{m=1}^{\infty} p_{z_m}\ \cos(m\varphi)\\ v(\varphi) &= \sum_{m=1}^{\infty} v_m\ \sin(m\varphi)\\ u(\varphi) &= u_0 + \sum_{m=1}^{\infty} u_m\ \cos(m\varphi)\\ n_x(\varphi) &= n_{x_0} + \sum_{m=1}^{\infty} n_{x_m}\ \cos(m\varphi)\\ n_{x\varphi}(\varphi) &= \sum_{m=1}^{\infty} n_{x\varphi_m}\ \sin(m\varphi)\end{aligned} \tag{5.84}$$

The first derivatives according to φ result with the abbreviation:

$$\partial_\varphi() = \frac{\partial()}{\partial\varphi} \tag{5.85}$$

to:

$$\begin{aligned}
\partial_\varphi p_z &= -\sum_{m=1}^{\infty} p_{z_m}\ m\sin(m\varphi) \\
\partial_\varphi v &= \sum_{m=1}^{\infty} v_m\ m\cos(m\varphi) \\
\partial_\varphi u &= -\sum_{m=1}^{\infty} u_m\ m\sin(m\varphi) \\
\partial_\varphi n_x &= -\sum_{m=1}^{\infty} n_{x_m}\ m\sin(m\varphi) \\
\partial_\varphi n_{x\varphi} &= \sum_{m=1}^{\infty} n_{x\varphi_m}\ m\cos(m\varphi)
\end{aligned} \tag{5.86}$$

The following applies to the required second derivatives:

$$\begin{aligned}
\partial^2_\varphi p_z &= -\sum_{m=1}^{\infty} p_{z_m} \quad m^2\cos(m\varphi) \\
\partial^2_\varphi n_{x\varphi} &= -\sum_{m=1}^{\infty} n_{x\varphi_m} m^2\sin(m\varphi)
\end{aligned} \tag{5.87}$$

Finally, the third derivative of p_z is:

$$\partial^3_\varphi p_z = \sum_{m=1}^{\infty} p_{z_m} m^3\sin(m\varphi) \tag{5.88}$$

If the variables in Eq. 5.80 to Eq. 5.83 are replaced by the Fourier series and a coefficient comparison is carried out, the following results for each wave number $m \geq 1$ the following system of four 1st order differential equations:

$$\begin{bmatrix} v_m \\ u_m \\ n_{x_m} a \\ n_{x\varphi_m} a \end{bmatrix}' = \begin{Bmatrix} 0 & \frac{m}{a} & 0 & \frac{1}{G\ h\ a} \\ 0 & 0 & \frac{1}{E\ h\ a} & 0 \\ 0 & 0 & 0 & -\frac{m}{a} \\ 0 & 0 & 0 & 0 \end{Bmatrix} \begin{bmatrix} v_m \\ u_m \\ n_{x_m}\ a \\ n_{x\varphi_m}\ a \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{v}{E}\frac{a}{h} p_{z_m} \\ 0 \\ m\ a\ p_{z_m} \end{bmatrix} \tag{5.89}$$

These four equations can be integrated one after the other starting from the bottom and one obtains with the dimensionless coordinate $\xi = x/a$ and the state vector $\vec{U}_i$:

$$\vec{U}_i = \begin{bmatrix} v_m \\ u_m \\ n_{x_m}\ a \\ n_{x\varphi_m}\ a \end{bmatrix}_i \tag{5.90}$$

following transfer matrix $\{W_{MTH}\}$ of the cylindrical shell according to the membrane theory:

$$\{W_{MTH}\} = \begin{Bmatrix} 1 & m\xi & \frac{m}{2Eh}\xi^2 & -\frac{1}{Eh}\frac{m^2}{6}\xi^3 + \frac{1}{Gh}\xi \\ 0 & 1 & \frac{1}{Eh}\xi & -\frac{m}{2Eh}\xi^2 \\ 0 & 0 & 1 & -m\xi \\ 0 & 0 & 0 & 1 \end{Bmatrix} \tag{5.91}$$

With the assumption that the normal load p_{z_m} is constant over the section length, the integration results in the following load column:

$$[L_{MTH}] = \begin{bmatrix} -\frac{1}{Eh}\left[\frac{m^3}{24}\xi^4 + \frac{mv}{2}\xi^2\right] + \frac{m}{2Gh}\xi^2 \\ -\frac{1}{Eh}\left(v\xi + \frac{m^2}{6}\xi^3\right) \\ -\frac{m^2}{2}\xi^2 \\ m\xi \end{bmatrix} p_{z_m} a^2 \tag{5.92}$$

For $m = 0$, only symmetrical state variables occur ($v = 0$ and $n_{x\varphi} = 0$). Therefore, by deleting the 1st and 4th rows and the corresponding columns with the state vector $\vec{U}_i$, one obtains:

$$\vec{U}_i = \begin{bmatrix} u_m \\ n_{x_m} a \\ 1 \end{bmatrix}_i \tag{5.93}$$

the transfer matrix:

$$\{W_{MTH_L}\} = \begin{Bmatrix} 1 & \frac{1}{Eh}\xi & -\frac{v}{Eh}\xi p_{z_0} a^2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix} \tag{5.94}$$

The boundary conditions for differently shaped shell edges must be respectively defined. Deformations u deform the cross-section out of the plane, one speaks of

warping, while displacements v change the cross-sectional shape, i.e. ovalise.
The calculation is carried out separately for each Fourier element of the external load with the wave number m and the amplitude p_{z_m}, and then the respective results are superimposed.

In the application studied in this thesis, a horizontal cylindrical tank half-filled with liquid is considered, Fig. 5.19. The tank is closed at both ends with a flat, circular plate which is rigid in its plane but flexible vertically. The forces exerted by the liquid on the lids are neglected in the calculation.

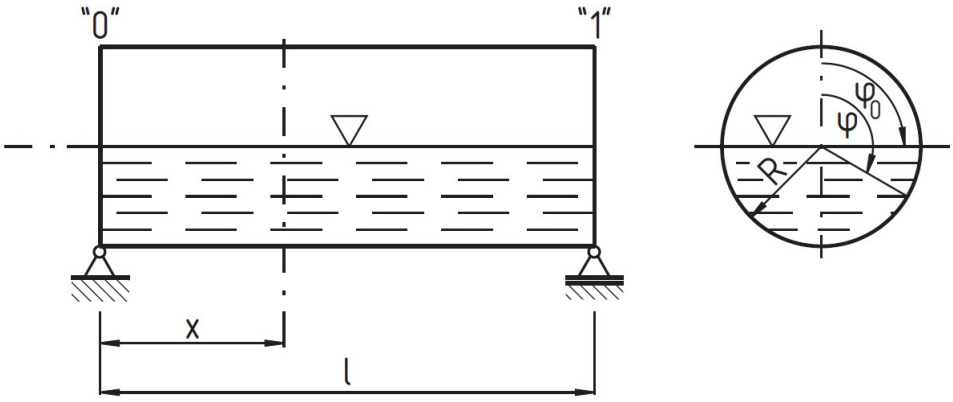


Figure 5.19. Half-filled vessel. Image source: [18]

The following boundary conditions arise at the edges:

$$v_{m_0} = v_{m_1} = 0$$
$$n_{x_{m_0}} = n_{x_{m_1}} = 0$$

As a result of the liquid filling, a hydrostatic pressure $p^*(\varphi)$ arises:

$$p^*(\varphi) = \gamma R(cos\ \varphi_0 - cos\ \varphi)$$

which can be described by a Fourier series:

$$p^*(\varphi) = \frac{a_0}{2} + \sum_{m=1}^{\infty} a_m \cos m\varphi$$

with the coefficients:

$$a_m = \frac{1}{\pi} \int_0^{2\pi} p^*(\varphi) \cos m\varphi \mathrm{d}\varphi$$

For the half-filled cylinder ($\varphi_0 = \frac{\pi}{2}$), one gets:

$$p^*(\varphi) = \frac{2P}{R\pi l}(0,3183 - 0,5\cos\varphi + 0,2122\cos 2\varphi - 0,0424\cos 4\varphi + \ldots)$$

where P is the filling weight:

$$P = \frac{1}{2}\gamma R^2 \pi l$$

In the solution of this problem, x has been chosen as $x = l/2$, so in the middle of the cylindrical shell, where the stress on the direction of the length of the shell is maximum. The liquid that fills the cylindrical shell has been chosen as water, with a specific weight of $\gamma = 1000\ kg/m^3$ at $4°C$.

The approximation of the real pressure distribution by a Fourier series, which is terminated after the term with $m = 4$, is shown in Fig. 5.20. The further calculation is carried out with the wave numbers $0 \leq m \leq 4$.

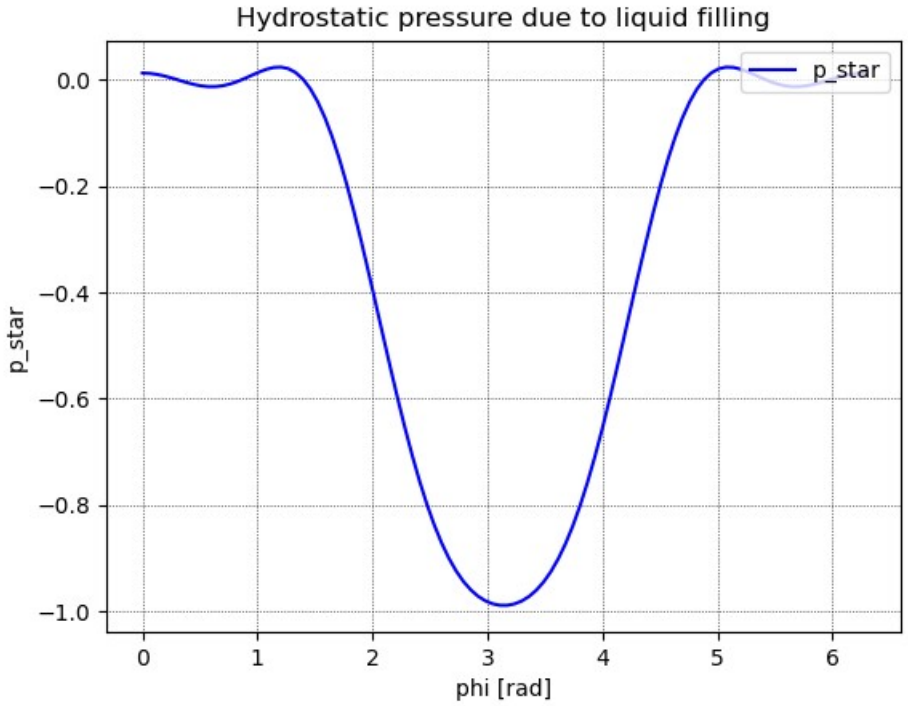


Figure 5.20. Pressure load Fourier series up to wave number $m = 4$.

To solve the problem, the equations resulting from Eq. 5.90 to Eq. 5.92, can be written in a system of equations as:

$$\vec{U}_{i+1} = \{W_{L_{MTH}}\}\ \vec{U}_i \tag{5.95}$$

where $\{W_{L_{MTH}}\}$ is the transfer matrix of Eq. 5.91, with the load vector of Eq. 5.92 appended as the last column of the matrix. In order to get the right dimensions to solve the system, the value 1 is appended as the last element of the two state vectors of Eq. 5.90, and the row vector $\{0\ 0\ 0\ 0\ 1\}$ is appended as the last row of the matrix $\{W_{L_{MTH}}\}$.

The resulting transfer matrix with the load vector appended is: $\{W_{L_{MTH}}\} =$

$$\begin{Bmatrix} 1 & m\xi & \frac{m}{2Eh}\xi^2 & -\frac{1}{Eh}\frac{m^2}{6}\xi^3 + \frac{1}{Gh}\xi & \left[-\frac{1}{Eh}\left[\frac{m^3}{24}\xi^4 + \frac{mv}{2}\xi^2\right] + \frac{m}{2Gh}\xi^2\right]\ p_{z_m}a^2 \\ 0 & 1 & \frac{1}{Eh}\xi & -\frac{m}{2Eh}\xi^2 & \left[-\frac{1}{Eh}\left(v\xi + \frac{m^2}{6}\xi^3\right)\right]\ p_{z_m}a^2 \\ 0 & 0 & 1 & -m\xi & \left[-\frac{m^2}{2}\xi^2\right]\ p_{z_m}a^2 \\ 0 & 0 & 0 & 1 & [m\xi]\ p_{z_m}a^2 \\ 0 & 0 & 0 & 0 & 1 \end{Bmatrix}$$

and the resulting state vectors are:

$$\vec{U}_0 = \begin{bmatrix} v_m \\ u_m \\ n_{x_m}\ a \\ n_{x\varphi_m}\ a \\ 1 \end{bmatrix}_0 \tag{5.96}$$

and:

$$\vec{U}_1 = \begin{bmatrix} v_m \\ u_m \\ n_{x_m}\ a \\ n_{x\varphi_m}\ a \\ 1 \end{bmatrix}_1 \tag{5.97}$$

The following applies to the stress $n_x(\varphi)$ of the shell:

$$n_x(\varphi) = \sum_{m=0}^{4} \bar{n}_{x_m} \cos m\varphi \tag{5.98}$$

where m is the Fourier wave number, and $\bar{n}_{x_m}$ is the coefficient of the Fourier series correspondent to the respective wave number that have been computed solving the Eq. 5.95 and are:

$$\begin{aligned} \bar{n}_{x_1} &= -1 \\ \bar{n}_{x_2} &= \ \ 1.698 \\ \bar{n}_{x_4} &= -1.357 \end{aligned} \tag{5.99}$$

In Fig. 5.21 one can see the course of the stress $n_x(\varphi)$ in the centre of the shell, so with x = l/2. The stress is adimensionalized dividing it by a $factor = (8\pi)/(P\ l)$ in order to have a better understanding and comparison of the behaviour along the circumference of the cylindrical shell in the centre of the shell.[18]

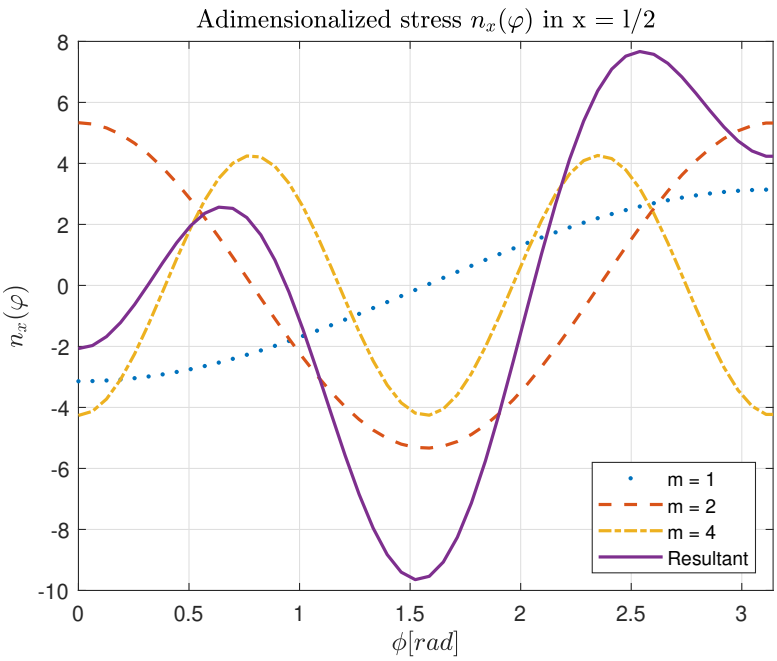


Figure 5.21. Stress $n_x(\varphi)$ in the centre of the shell, for x = l/2 , where m is the wave number considered in the Fourier series representation, and the continuous line is the complete Fourier series resultant of the summation of all the wave numbers from m = 1 to m = 4.

Once the problem has been analytically solved, two different data-sets has been created, with the implementation of a `MATLAB` script. The first data-set, used for the training of the neural network, is composed of 16000 samples with different lengths of the shell and different diameters, using the same material and the same applied load. The second data-set, used for the validation of the neural network, is composed of samples completely different from the samples of the first data-set, as well with different lengths of the shell and different diameters, using the same material and the same applied load.

A part of the validation data-set is represented from the analytical computations, in order to have a better understanding of the behaviour, in Fig. 5.22 fixing a diameter, and varying just the lengths of the different shells used.

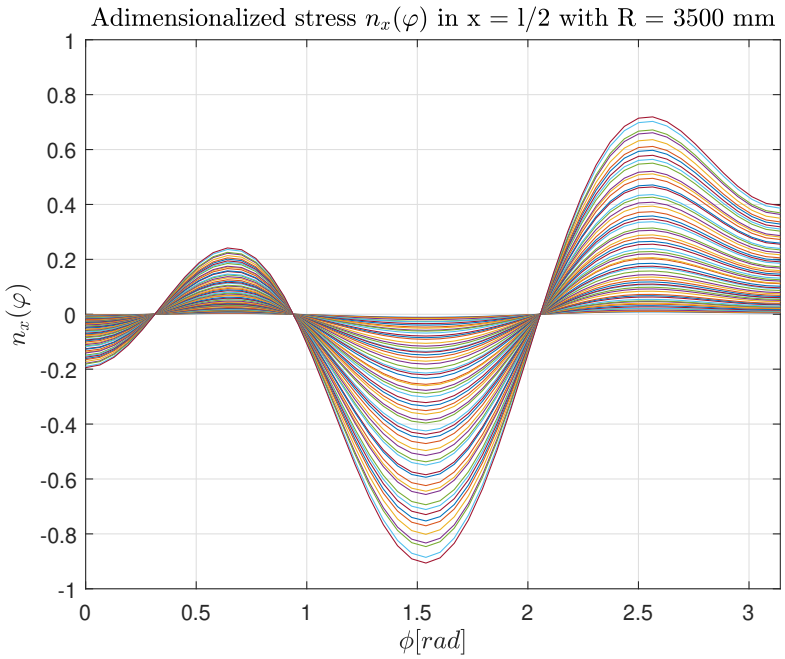


Figure 5.22. Stress $n_x(\varphi)$ in the centre of the shell, for x = l/2 with R = 3500 mm.

Chapter 6

Neural Network used to study a cylindrical shell under pressure load

6.1 The direct problem: shell length and diameters given as input to get the stresses

This chapter describes in detail the neural network architecture used for the cylindrical shell structure study, with particular reference to the network composition, the reasons behind the choices made and the results obtained using an internal pressure load.

For the study of the cylindrical shell problem represented in Fig. 6.1, the choice was made to create a neural network that could calculate the stresses arising out of the plane of the shell section, distributed around the circumference of the shell itself.

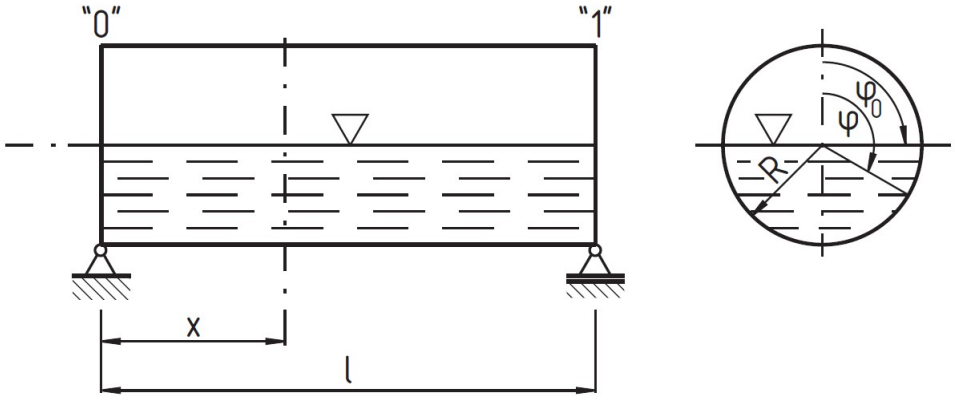


Figure 6.1. Half-filled vessel. Image source: [18]

It should be noted that the load applied to the shell varies around the circumference of the shell itself, but remains constant along the longitudinal direction of the structure. This choice stems from the fact that in this way it is possible to calculate

where, if any, the stress exceeds the ultimate one and causes the material to break.

For this reason, different neural network architectures have been created, all characterised by using the same neural network training inputs. The inputs given for the training phase are the diameter and length of the shell, varying the values for length between 350 and 3500 mm and the inner diameter of the shell between 3005 and 5000 mm. The choice of these values is dictated by the fact that one wishes to remain within the range of validity of container theory and membrane theory, which prove to be particularly effective when the shell length does not exceed the inner diameter and the wave number used for the approximation by means of a Fourier series of the internal pressure load does not exceed m = 4.

The inputs given to the beam are only the values that vary in the samples of the training data-set, making it possible to make the architecture of the neural network completely independent of the Young's modulus, the Poisson's coefficient, and consequently of the material chosen to make the Shell, possibly even implementing a shell made of composite material, using equivalent Young modulus and Poisson coefficient values that best represent the composite structure.

For the shell under study was chosen to use aluminium as a material, characterised by a Young modulus of 70 GPa, a shear modulus of 26 GPa and a Poisson coefficient of 0.3. The thickness of the shell structure was set at 1 mm, which is also a realistic value in terms of a lower limit for the actual manufacturability of this type of structure in the manufacturing field, as it has to withstand an internal pressure load.

Of course, as in the case of the beam analysed in Chap. 4, if one wanted to study stresses in the plane of the shell section or other types of parameters of a structural analysis of the shell (changing also the material of the shell for instance), it would be necessary to re-train the neural network with data-sets that refer to the particular load condition desired and the particular output desired. The neural network is given a training data-set on which the algorithm then goes on to study the model that it will use in predicting the results. With the aim of obtaining an evaluation of the performance during the training process, it was also chosen here, as in the case of cantilever beam, to set the amount of data used to create an initial performance estimate of how the algorithm is allowing the neural network to learn the mathematical relationships, by means of a backward feedback process, at 10% of the total data provided. Then, a new data-set is supplied to the network with the diameter and shell length values provided as input, with the aim of evaluating the performance of the learning algorithm and comparing the predictions obtained by the neural network with the results calculated analytically. The process scheme is therefore the same as that used for the cantilever beam, see Fig. 4.2.

For the shell structure many attempts were made before identifying the most suitable neural network architecture for studying the problem. With reference to the section on the cantilever beam neural network Sec. 4.1, also in this case the final choice of architecture fell on a network composed of three different layers, in which the input layer uses a rectified linear unit type activation function described in the chapter referring to the different activation functions. The second and third layers therefore use the same number of nodes as the input layer and the same activation function. Immediately following the third layer is the output player which uses a simply linear activation function, with the purpose of collecting the outputs produced by the third layer of the network. In the remainder of the chapter, we

therefore present the study of the various possibilities by varying the number of nodes, and the number of samples provided in the network's training data-set, with the aim of assessing which predictive model is the best.[2]

The number of samples provided in the training data-set is either 10000 or 16000. As in the case of the beam, the samples contained in the training data-set are arranged randomly in order to avoid any type of correlation that the network could detect simply by following the trend of the variation in the length and diameter of the shell, thus allowing the neural network to focus only on the mathematical relationships linking the study parameters of this structural analysis.

As the problem under study is a complex one, a neural network architecture approach consisting of a normalisation performed immediately in the first layer was also analysed, however, providing poor results and therefore discarding the possibility of introducing a normalisation from the outset. As an example, the training graph obtained by means of a training with implementation of normalisation is also provided: as can be seen from Fig. 6.2, the network fails to identify a prediction model and generalise it, soon diverging before reaching the maximum number of iterations allowed for the neural network itself.[11]

The stresses provided in output by the neural network are dimensionless thanks to the use of a factor $f = (8\,\pi)/(P\,l)$ that allows a better comparison of the results by varying the input data. Since the cylindrical shell is a symmetrical structure with respect to its axis, only the semicircle from an angle value $\varphi = 0$ up to $\varphi = \pi$ is studied. You only need to perform a rotation of the stresses obtained for the first semicircle to obtain the values on the other semicircle.

In order to best study the applications of the solution of this shell problem, experiments were carried out to obtain as output the n_x stress values corresponding to 1 angle, 2 different angles on the half-circle, 4 angles, 10 angles, 25 angles, and finally 50 angles to best characterise the entire half-circle.

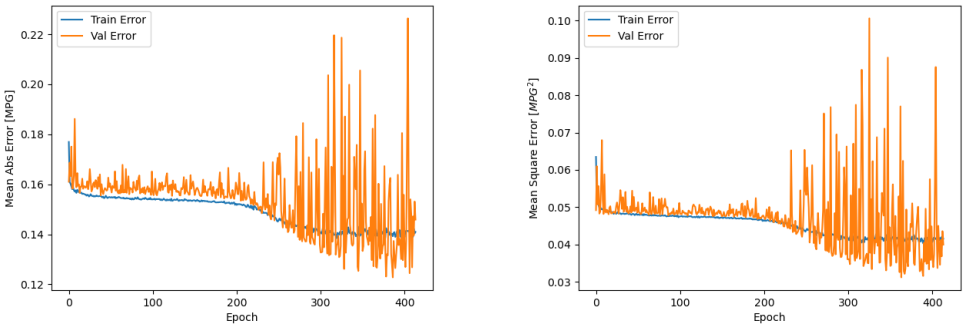
The values of mean absolute error and mean squared error you turn out to be very small as the scaling performed on the stress values meant that all the values were in the vicinity of zero, varying at most by 2 orders of magnitude as the angle considered, the length, and the diameter of the shell varied, and the values of MAE and MSE presented in Eq. 2.2 and Eq. 2.1 depend directly on these.

The batch size value chosen for the study of the shell problem was set at 64, as this value proved to be after many different attempts to be the best compromise value to ensure that the neural network was not trained excessively long in terms of time and performed very well in terms of the predictions provided by the learning algorithm.

Again, as with the beam cantilever, in order to guard against any randomness in the number of iterations required to achieve convergence, each individual neural network training process was performed four times for each data-set point, and the best results obtained for each cycle are presented below.

This chapter presents tables corresponding to the study of neural networks by varying the number of nodes for each layer, completed for each case study by fixing the number of outputs as previously explained.

An angle of $\varphi = 1.602$ rad was used for the problem characterized by only one value in output of angle φ.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.2. MAE and MSE for the 64 nodes normalized NN with 1 angle value output and 10000 rows data-set.

Angles equal to $\varphi = 0.769, 1.602$ rad were used for the problem characterized by 2 values in output of angle φ.

Angles equal to $\varphi = 0.769, 1.602, 2.308, 3.141$ rad were used for the problem characterized by 4 values in output of angle φ.

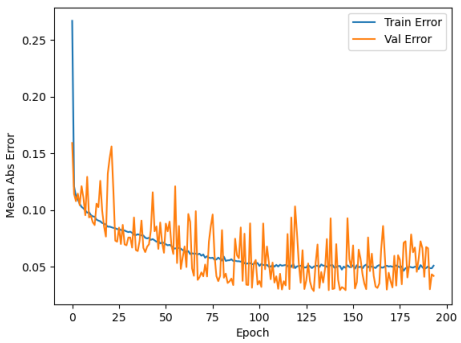
Angles equal to $\varphi = 0.064, 0.384, 0.769, 1.089, 1.602, 1.923, 2.308, 2.628, 2.821,$ 3.141 rad were used for the problem characterized by 10 output values of angle φ.

For the problem characterized by 25 output values of angle φ the angles equal to $\varphi = 0.064, 0.192, 0.320, 0.448, 0.577, 0.705, 0.833, 0.961, 1.089, 1.218, 1.346, 1.474,$ $1.602, 1.731, 1.859, 1.987, 2.115, 2.243, 2.372, 2.500, 2.628, 2.756, 2.885, 3.013, 3.141$ rad.

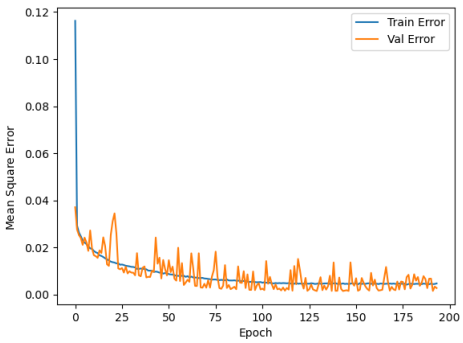
For the problem characterized by 50 output values of angle φ the angles equal to $\varphi = 0, 0.064, 0.128, 0.192, 0.256, 0.320, 0.384, 0.448, 0.512, 0.577, 0.641, 0.705, 0.769,$ $0.833, 0.897, 0.961, 1.025, 1.089, 1.154, 1.218, 1.282, 1.346, 1.410, 1.474, 1.538, 1.602,$ $1.666, 1.731, 1.795, 1.859, 1.923, 1.987, 2.051, 2.115, 2.179, 2.243, 2.308, 2.372, 2.436,$ $2.500, 2.564, 2.628, 2.692, 2.756, 2.821, 2.885, 2.949, 3.013, 3.077, 3.141$ rad.

node per layer	epoch	MAE	MSE
4-4-4	170	0.095778	0.014715
8-8-8	498	0.039823	0.003097
16-16-16	172	0.046193	0.004116
32-32-32	158	0.045433	0.004189
64-64-64	231	0.041474	0.004119
128-128-128	382	0.038397	0.003386

Table 6.1. Training performance analysis for the neural network architecture with 10000 samples data-set and 1 output.

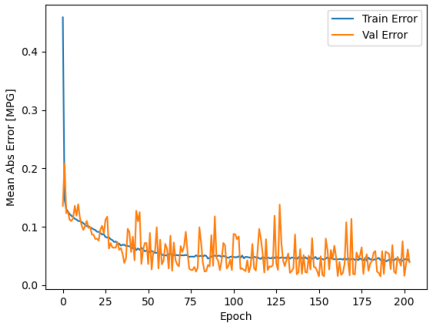


(a) The MAE Mean Absolute Error.

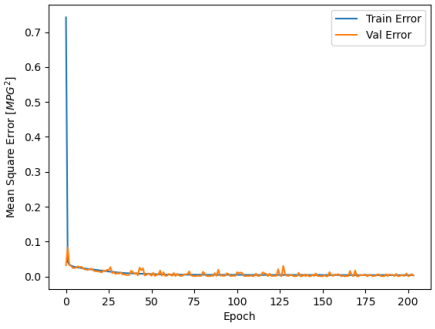


(b) The MSE Mean Squared Error.

Figure 6.3. MAE and MSE for the 8 nodes NN with 1 angle value output and 10000 rows data-set.



(a) The MAE Mean Absolute Error.

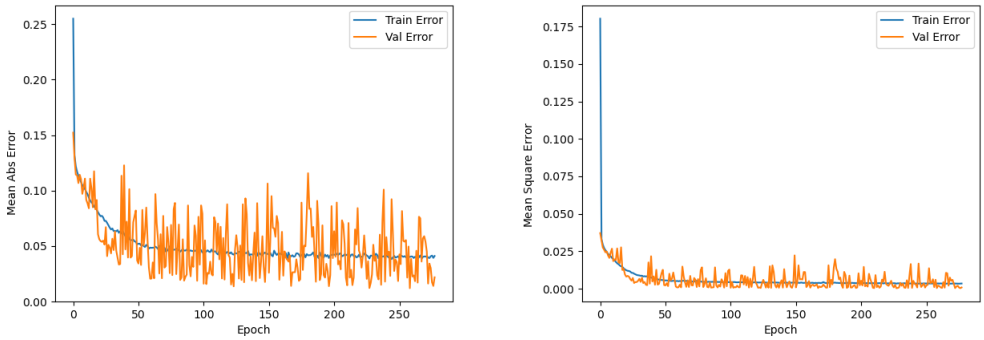


(b) The MSE Mean Squared Error.

Figure 6.4. MAE and MSE for the 64 nodes NN with 1 angle value output and 10000 rows data-set.

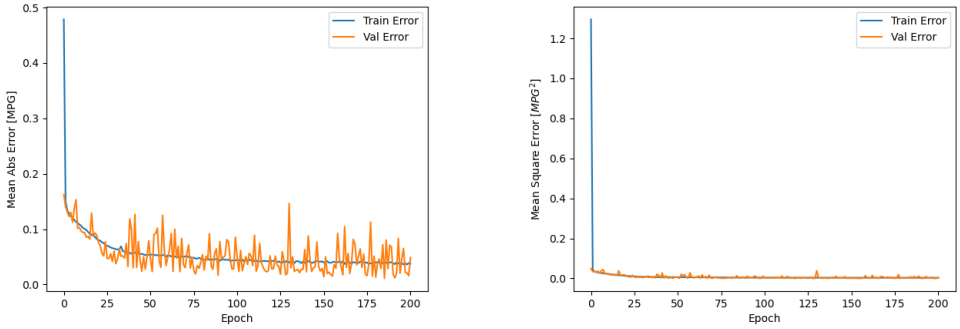
node per layer	epoch	MAE	MSE
4-4-4	299	0.056972	0.005747
8-8-8	348	0.050528	0.004408
16-16-16	233	0.041033	0.003619
32-32-32	523	0.036913	0.002416
64-64-64	146	0.057068	0.006257
128-128-128	218	0.046602	0.004302

Table 6.2. Training performance analysis for the neural network architecture with 16000 samples data-set and 1 output.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.5. MAE and MSE for the 32 nodes NN with 1 angle value output and 16000 rows data-set.

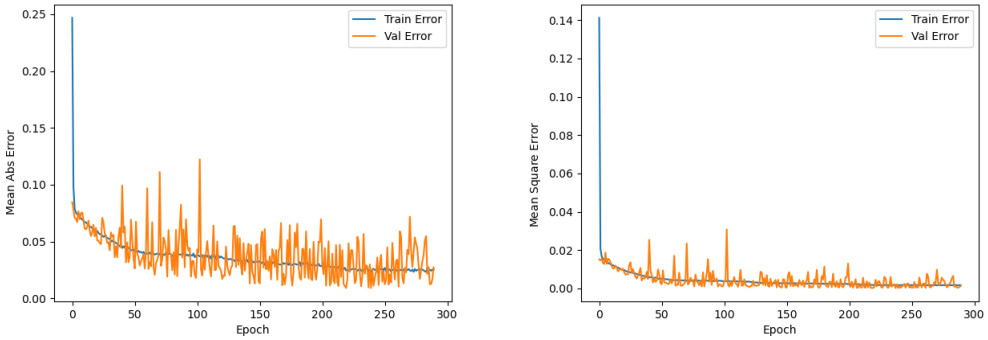


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.6. MAE and MSE for the 64 nodes NN with 1 angle value output and 16000 rows data-set.

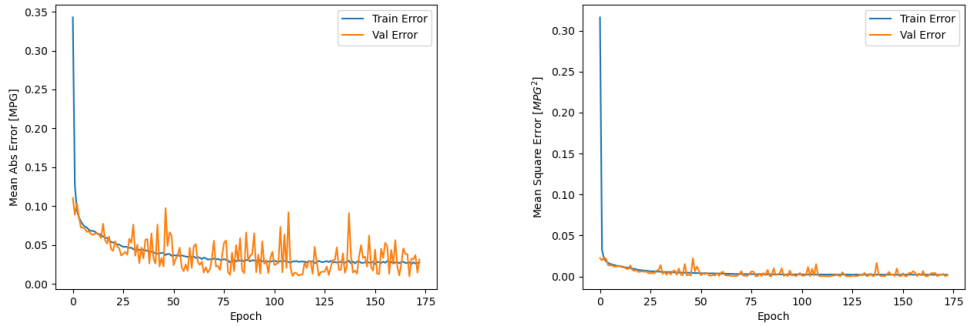
node per layer	epoch	MAE	MSE
4-4-4	244	0.030422	0.002218
8-8-8	348	0.037174	0.003757
16-16-16	393	0.025958	0.001773
32-32-32	387	0.025600	0.001725
64-64-64	223	0.029398	0.002406
128-128-128	187	0.025982	0.002074

Table 6.3. Training performance analysis for the neural network architecture with 10000 samples data-set and 2 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.7. MAE and MSE for the 32 nodes NN with 2 angle values output and 10000 rows data-set.

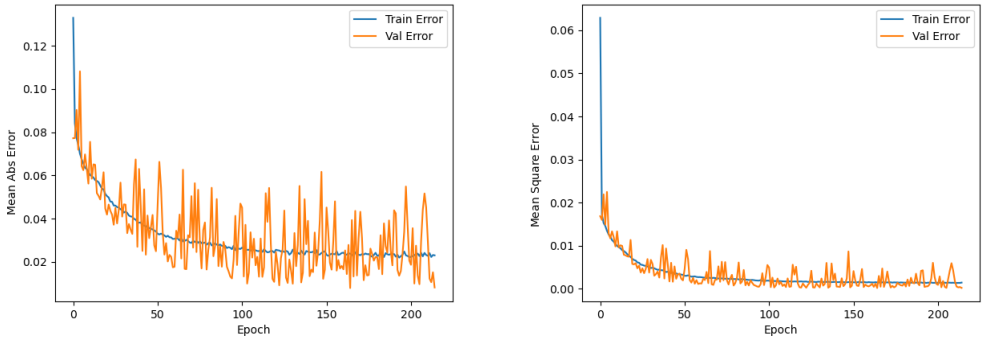


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.8. MAE and MSE for the 64 nodes NN with 2 angle values output and 10000 rows data-set.

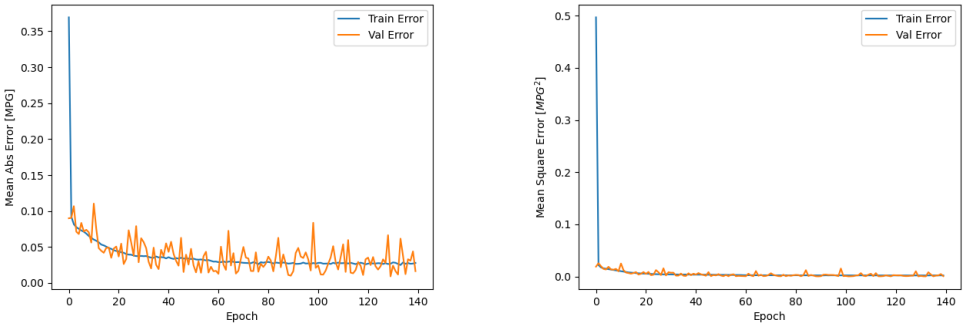
node per layer	epoch	MAE	MSE
4-4-4	64	0.212596	0.090133
8-8-8	271	0.024770	0.001588
16-16-16	215	0.023025	0.001466
32-32-32	217	0.026679	0.002002
64-64-64	147	0.025701	0.001894
128-128-128	290	0.024276	0.001496

Table 6.4. Training performance analysis for the neural network architecture with 16000 samples data-set and 2 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.9. MAE and MSE for the 16 nodes NN with 2 angle values output and 16000 rows data-set.

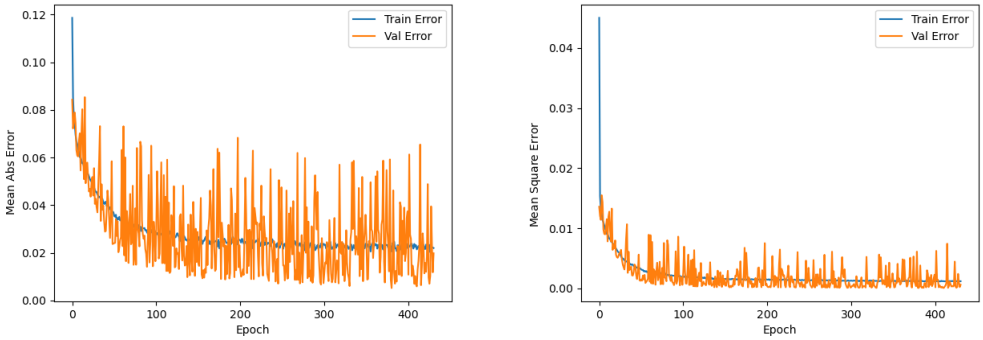


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.10. MAE and MSE for the 64 nodes NN with 2 angle values output and 16000 rows data-set.

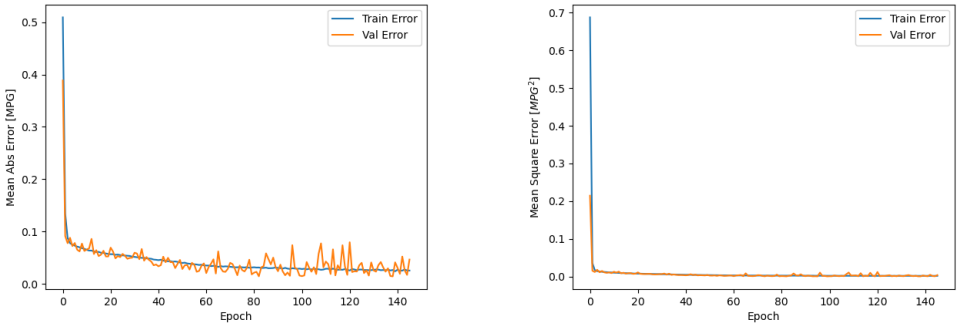
node per layer	epoch	MAE	MSE
4-4-4	443	0.027254	0.001523
8-8-8	329	0.025498	0.001483
16-16-16	431	0.021999	0.001206
32-32-32	227	0.024241	0.001486
64-64-64	130	0.029329	0.002224
128-128-128	299	0.021172	0.001255

Table 6.5. Training performance analysis for the neural network architecture with 10000 samples data-set and 4 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.11. MAE and MSE for the 16 nodes NN with 4 angle values output and 10000 rows data-set.

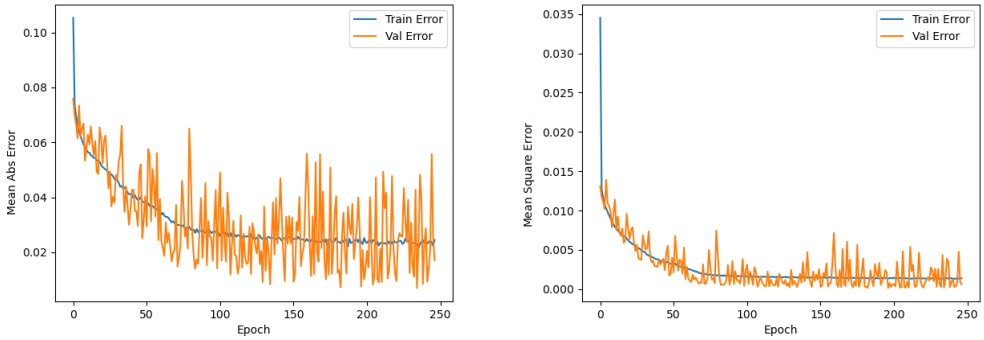


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.12. MAE and MSE for the 64 nodes NN with 4 angle values output and 10000 rows data-set.

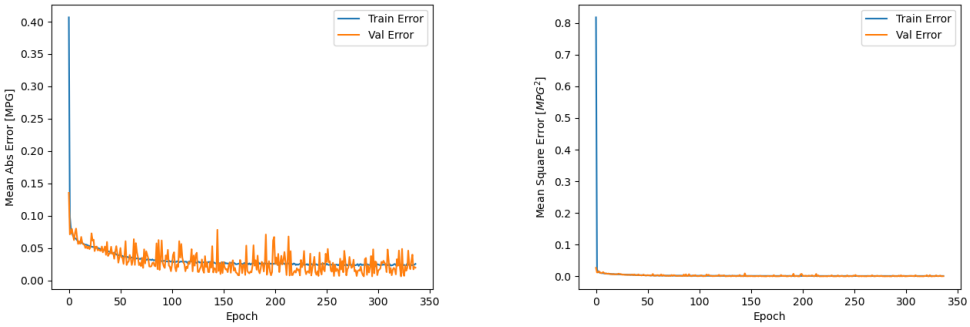
node per layer	epoch	MAE	MSE
4-4-4	127	0.197724	0.069611
8-8-8	387	0.024524	0.001305
16-16-16	125	0.022751	0.001402
32-32-32	302	0.023975	0.001473
64-64-64	285	0.022852	0.001415
128-128-128	169	0.025298	0.001511

Table 6.6. Training performance analysis for the neural network architecture with 16000 samples data-set and 4 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.13. MAE and MSE for the 16 nodes NN with 4 angle values output and 16000 rows data-set.

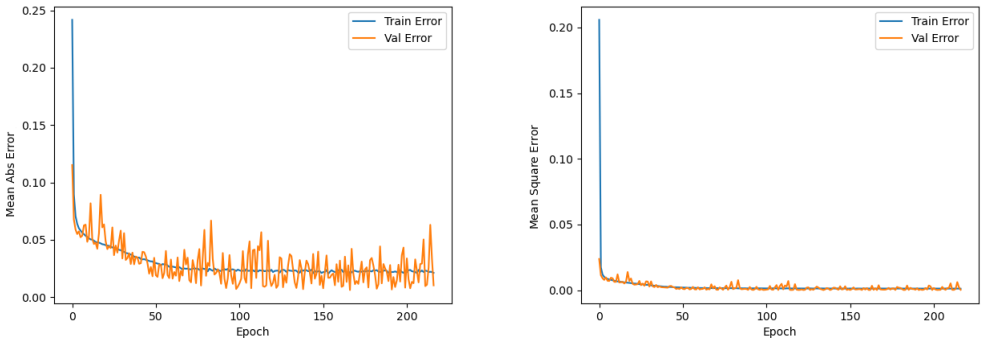


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.14. MAE and MSE for the 64 nodes NN with 4 angle values output and 16000 rows data-set.

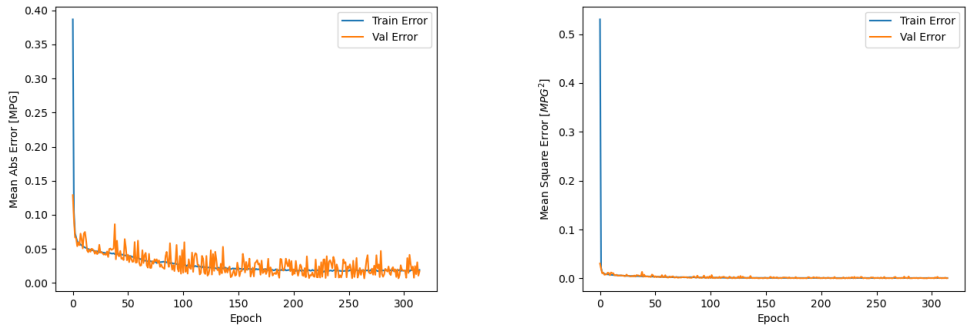
node per layer	epoch	MAE	MSE
4-4-4	130	0.163534	0.050737
8-8-8	234	0.020221	0.001254
16-16-16	317	0.021598	0.001096
32-32-32	327	0.020029	0.001083
64-64-64	248	0.018936	0.001035
128-128-128	393	0.019155	0.000813

Table 6.7. Training performance analysis for the neural network architecture with 10000 samples data-set and 10 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.15. MAE and MSE for the 128 nodes NN with 10 angle values output and 10000 rows data-set.

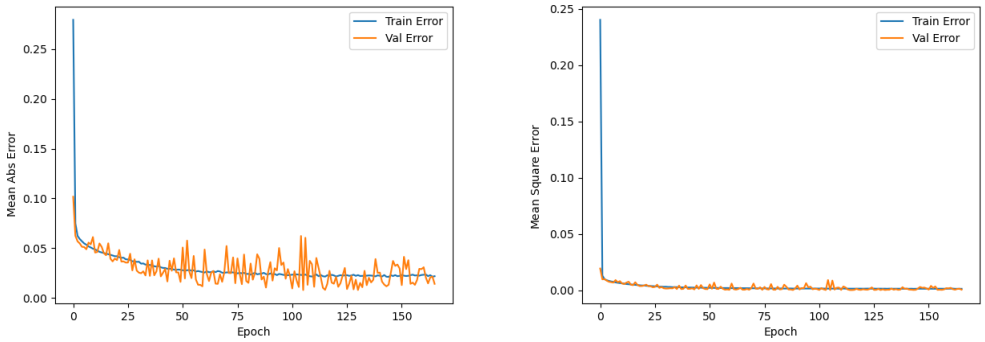


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.16. MAE and MSE for the 64 nodes NN with 10 angle values output and 10000 rows data-set.

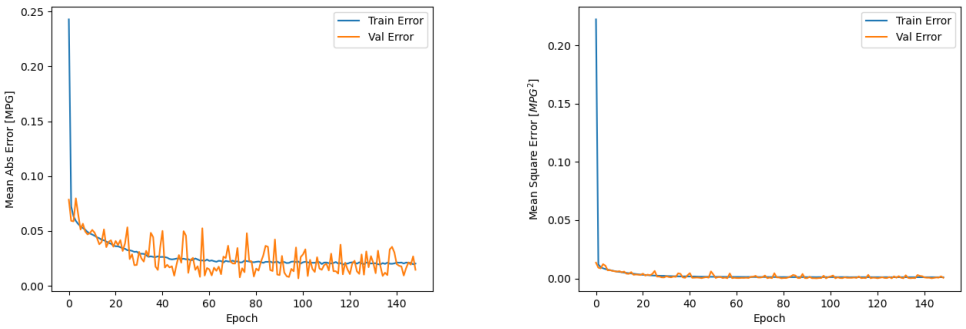
node per layer	epoch	MAE	MSE
4-4-4	233	0.022956	0.001261
8-8-8	168	0.025441	0.001386
16-16-16	194	0.021315	0.001126
32-32-32	195	0.019784	0.001014
64-64-64	125	0.021273	0.001176
128-128-128	324	0.018338	0.000876

Table 6.8. Training performance analysis for the neural network architecture with 16000 samples data-set and 10 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.17. MAE and MSE for the 128 nodes NN with 10 angle values output and 16000 rows data-set.

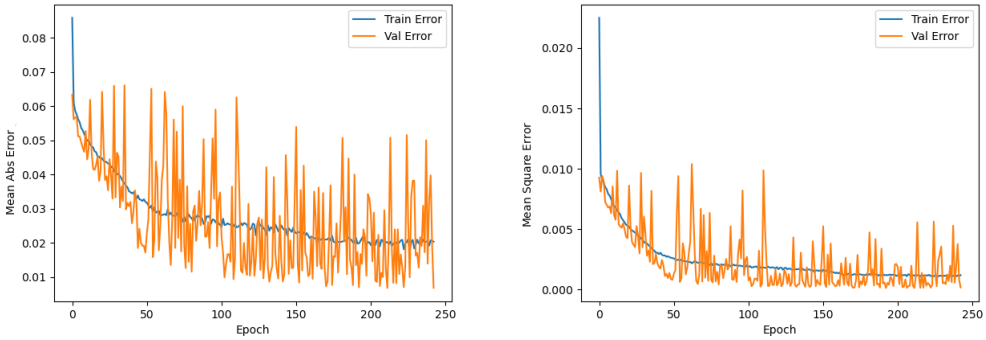


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.18. MAE and MSE for the 64 nodes NN with 10 angle values output and 16000 rows data-set.

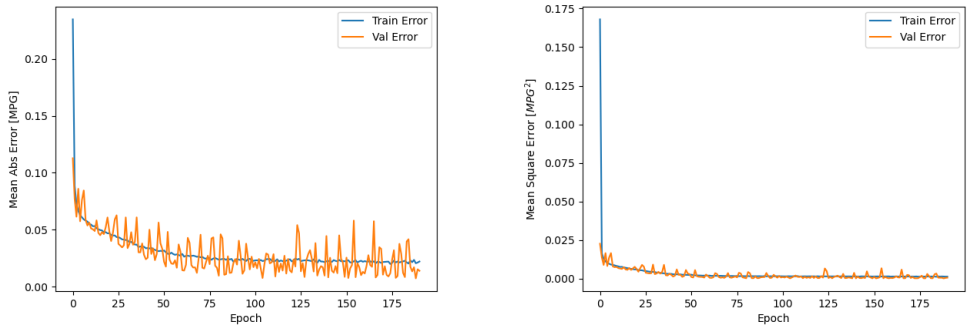
node per layer	epoch	MAE	MSE
4-4-4	388	0.024247	0.001414
8-8-8	339	0.021138	0.001204
16-16-16	518	0.017996	0.000856
32-32-32	237	0.020170	0.001122
64-64-64	238	0.021953	0.001235
128-128-128	243	0.020358	0.001199

Table 6.9. Training performance analysis for the neural network architecture with 10000 samples data-set and 25 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.19. MAE and MSE for the 16 nodes NN with 25 angle values output and 10000 rows data-set.

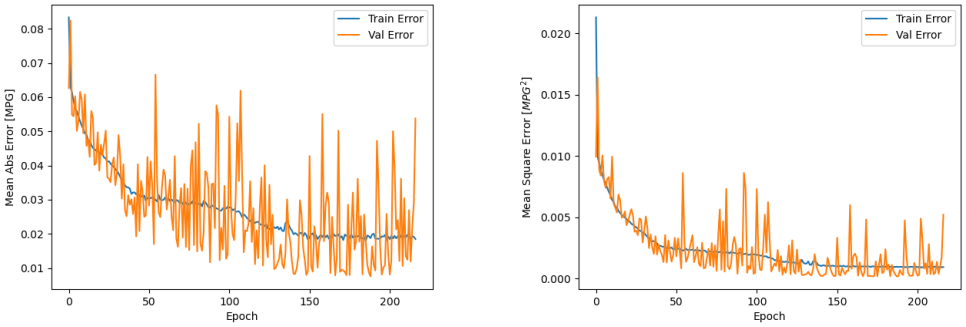


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.20. MAE and MSE for the 64 nodes NN with 25 angle values output and 10000 rows data-set.

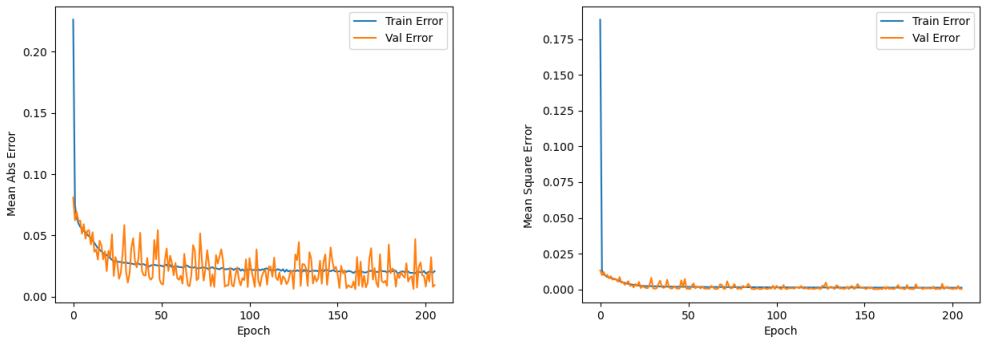
node per layer	epoch	MAE	MSE
4-4-4	120	0.163027	0.053760
8-8-8	183	0.020574	0.001158
16-16-16	206	0.020717	0.001167
32-32-32	168	0.020846	0.001248
64-64-64	401	0.019138	0.000907
128-128-128	127	0.026385	0.001537

Table 6.10. Training performance analysis for the neural network architecture with 16000 samples data-set and 25 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.21. MAE and MSE for the 16 nodes NN with 25 angle values output and 16000 rows data-set.

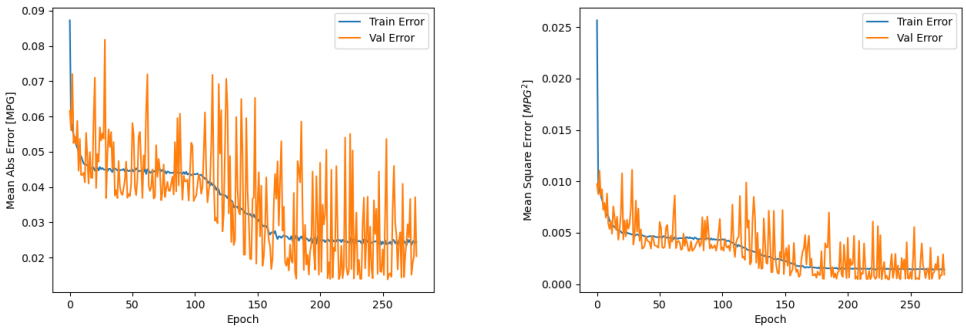


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.22. MAE and MSE for the 64 nodes NN with 25 angle values output and 16000 rows data-set.

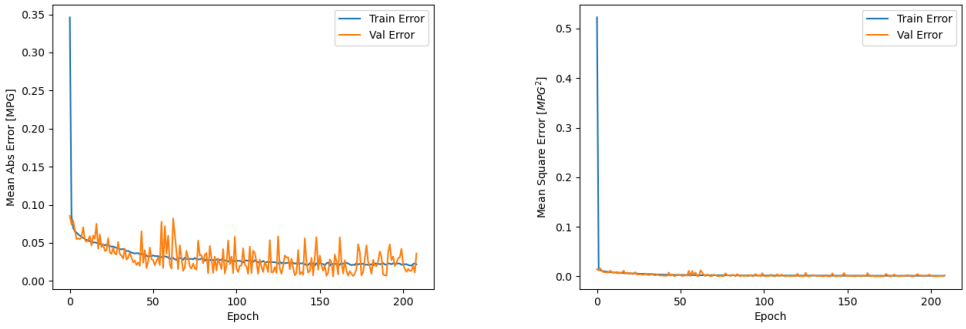
node per layer	epoch	MAE	MSE
4-4-4	466	0.030013	0.002371
8-8-8	209	0.021741	0.001278
16-16-16	154	0.024214	0.001467
32-32-32	193	0.022444	0.001268
64-64-64	194	0.025819	0.001701
128-128-128	60	0.162566	0.053502

Table 6.11. Training performance analysis for the neural network architecture with 10000 samples data-set and 50 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.23. MAE and MSE for the 8 nodes NN with 50 angle values output and 10000 rows data-set.

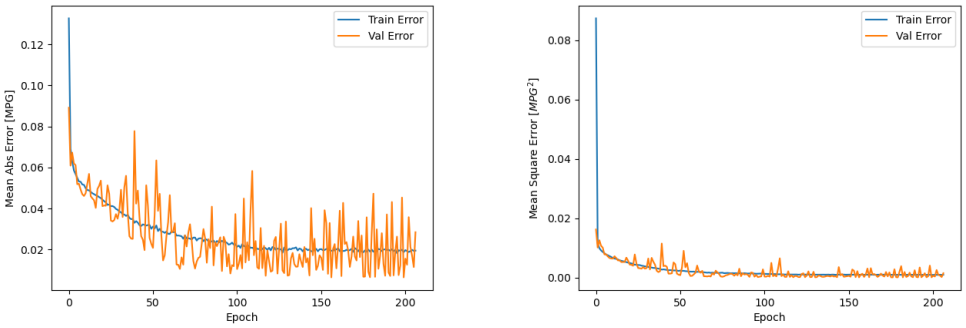


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.24. MAE and MSE for the 64 nodes NN with 50 angle values output and 10000 rows data-set.

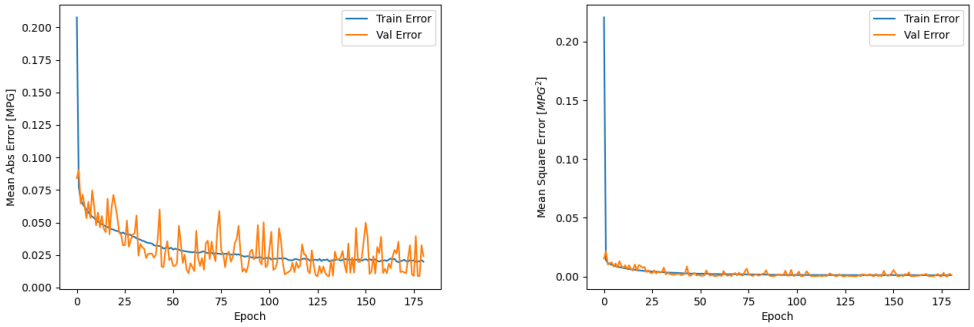
node per layer	epoch	MAE	MSE
4-4-4	296	0.019424	0.000985
8-8-8	157	0.022347	0.001261
16-16-16	197	0.021403	.001265
32-32-32	258	0.018480	0.001043
64-64-64	181	0.019944	0.001133
128-128-128	84	0.162238	0.053503

Table 6.12. Training performance analysis for the neural network architecture with 16000 samples data-set and 50 outputs.



(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

Figure 6.25. MAE and MSE for the 32 nodes NN with 50 angle values output and 16000 rows data-set.

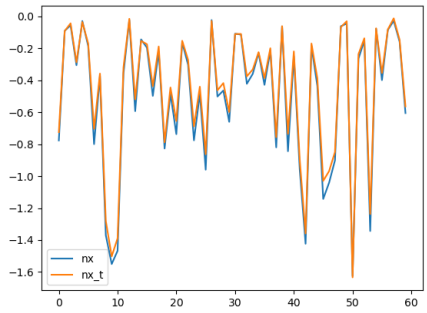


(a) The MAE Mean Absolute Error. **(b)** The MSE Mean Squared Error.

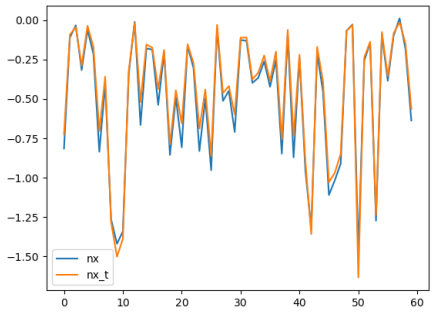
Figure 6.26. MAE and MSE for the 64 nodes NN with 50 angle values output and 16000 rows data-set.

A careful analysis of results shows that the network that performs best in terms of predicting results is the one consisting of three layers each composed of 64 nodes, using the one composed of 16000 examples as the training data-set. This consideration stems from the fact that using neural networks composed of fewer nodes per layer sometimes yields lower MAE and MSE error values as output than using the network consisting of 64 nodes per layer, but this error is actually obtained as the average of the oscillations that the learning algorithm makes in the validation curve on the training curve. In fact, the oscillations that are noticed on the validation curve are due to the fact that the network using 10% of the data provided in the training data-set, fails to develop a generalized prediction model but tends to produce noise thus going to "learn by heart" the data instead of studying its relationships to make a prediction. In these cases, in fact, the training curve has lower error values than those produced by the neural network consisting of 64 nodes per layer, but as seen in the validation curves, the predictions on the stress values are reproduced perfectly due to the fact that the network has "learned by heart" and reconstructed relationships based only on the trend of the curve and not by going to study its mathematical properties that read the inputs to the outputs.

In doing so, therefore, it turns out that the network fails to go to generalize the problem by means of the learning algorithm, and then provides the output results by going to reconstruct the trend of the curve that refers to how the stress is worth by varying the length and diameter of the Shell. In the present case, this behavior does not induce major problems simply because the number of samples provided with the training data-set has a limited range of validity, as explained at the beginning of the chapter, thus making the results valid for shell structures that have a length and internal diameter within the ranges explained above. In the event that one were to analyze a problem with much wider ranges of variation, this could go on to create reliability problems, as the network would not necessarily be able to provide a result using an architecture composed of fewer than 64 nodes that is reliable is based on an understanding of the generalized problem and not on predictions dictated by the curve trend. Considering these evaluations, it is therefore possible to state that although sometimes the predictive model obtained with a network architecture is formed of 64 nodes per layer provides greater MAE and MSE errors than those obtained with architectures consisting of a smaller number of nodes, the predictive models that networks formed by architecture with 64 nodes per layer turn out to be better in terms of understanding and generalizing the problem. In conclusion, it can be seen that as the number of required outputs increases, the neural network gradually reduces its performance, going on to produce a prediction error that gradually increases slightly with respect to the expected result calculated analytically. The following pages show graphs with the neural network evaluations for each case study by varying the number of outputs. the validation data-set consists of length and internal diameter of the Shell structure varying in the ranges described at the beginning of this chapter, all values being absolutely unknown to the neural network, and thus avoiding the possibility of the network directly providing a number taken solely from memory. The y-axis of each graph represents the dimensionless stress n_x value, while the x-axis represents which sample is evaluated with the corresponding length and diameter assigned. The n_x stress results shown in the graph are all calculated for an angle $\varphi = 1.602\,rad \approx \pi/2$.

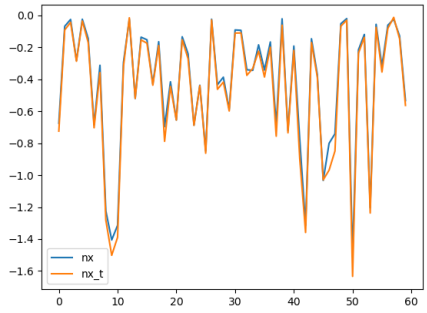


(a) NN trained with 10000 samples.

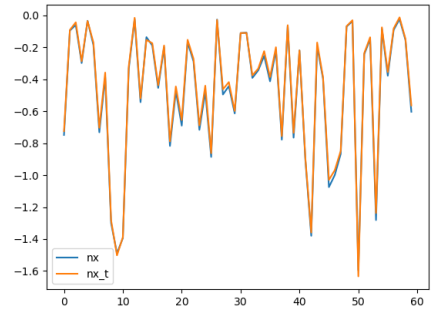


(b) NN trained with 16000 samples.

Figure 6.27. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 1 output from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.

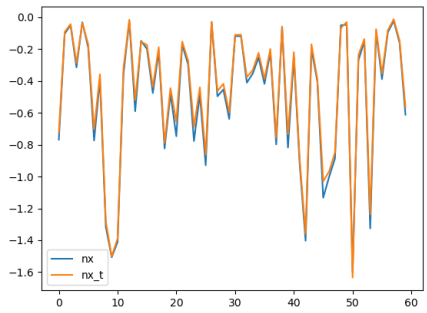


(a) NN trained with 10000 samples.

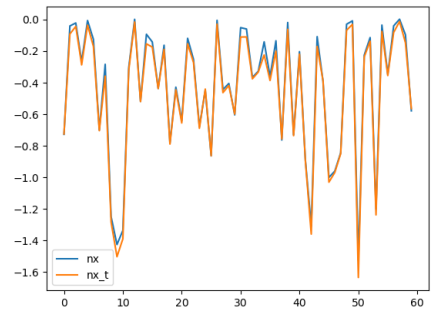


(b) NN trained with 16000 samples.

Figure 6.28. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 2 outputs from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.

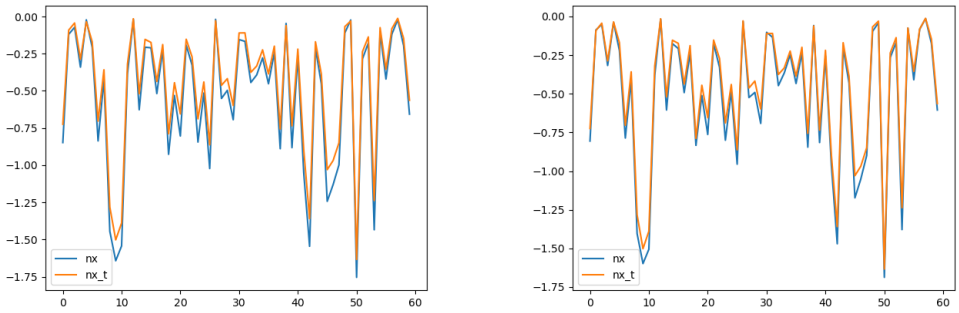


(a) NN trained with 10000 samples.



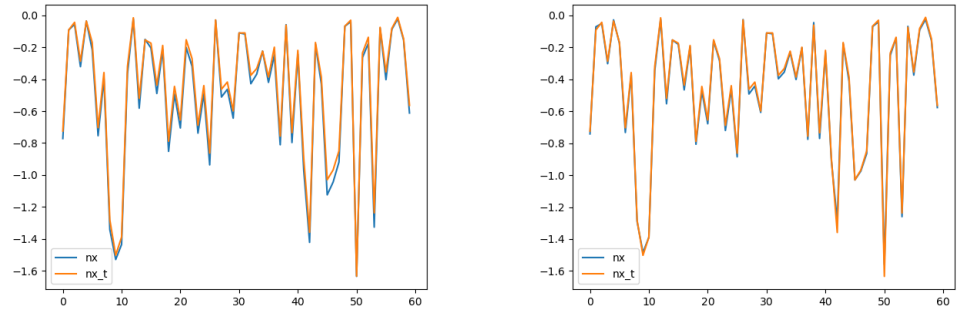
(b) NN trained with 16000 samples.

Figure 6.29. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 4 outputs from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.



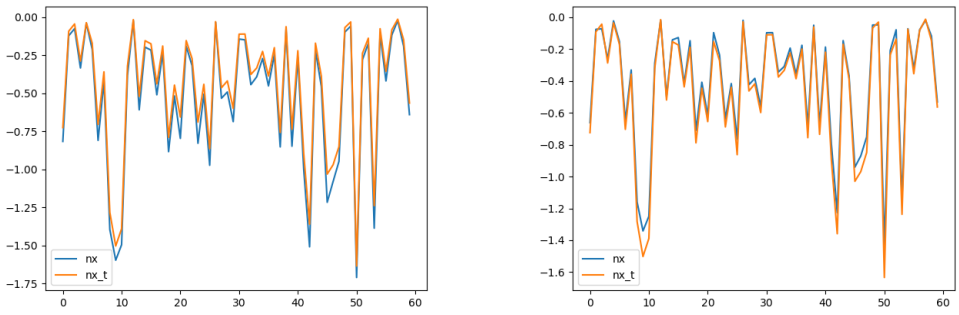
(a) NN trained with 10000 samples. **(b)** NN trained with 16000 samples.

Figure 6.30. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 10 outputs from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.



(a) NN trained with 10000 samples. **(b)** NN trained with 16000 samples.

Figure 6.31. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 25 outputs from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.



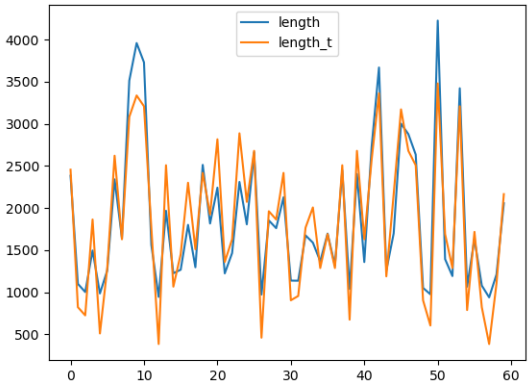
(a) NN trained with 10000 samples. **(b)** NN trained with 16000 samples.

Figure 6.32. Evaluation of the stress n_x at the angle value $\varphi = \pi/2$ rad, with 50 outputs from the 64 nodes/layer NN, varying the length of the shell and the inner diameter.

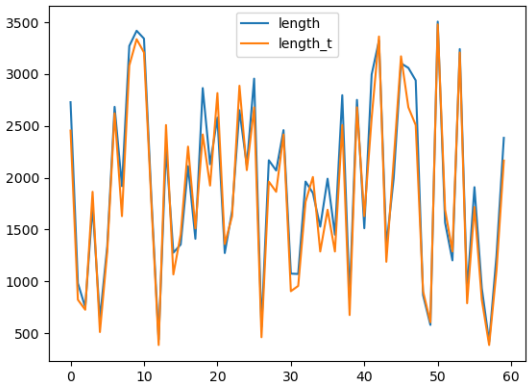
6.2 The inverse problem: stresses given as input to get shell length and diameters

An attempt was also made to study the shell-type structure by giving as input to the network the stress values measured at certain corners of the structure, and requesting as output the diameter and length of the structure affected by these stresses, as a consequence of the same internal pressure load also applied in this case. All these attempts were performed with the exact same network architecture used for the direct solution of the problem, i.e. the study described earlier in this chapter, giving as input diameter and length, and obtaining the stress value for different angles along the circumference of the shell cross-section. The results obtained in this way, however, are not satisfactory in terms of the inner diameter, as the network is unable to determine correct predictions of this value even when giving as input stress values measured for the same 50 different angles as previously mentioned. In terms of beam length prediction, however, it can be seen that the neural network manages to predict the correct value with good results, with errors of between 1% and 5%.

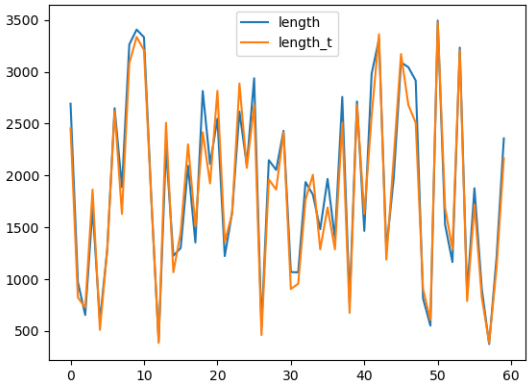
Attempts were made both with a neural network consisting of 64 nodes per layer, which turned out to be the best in predictive terms, and with 128 nodes per layer and then with 32 nodes per layer, obtaining, however, for the latter two, slightly lower performance levels than the architecture consisting of 64 nodes per layer. This fact is due to the density and distribution of samples contained in the training data-set of the network, which is sufficiently high and adequate for the lengths of the shell structure, but too sparse in terms of the diameter values provided as training, which are consequently not representative for the learning algorithm in the study of the network's inverse problem. As an example, the figure below, Fig. 6.33 shows the evaluation of the model that proved to be best for the prediction of the shell structure length, using as specified earlier a neural network architecture with 64 nodes per layer, Fig. 6.33c.



(a) Performance using a 128 nodes per layer NN, asking as output only the length of the shell structure.



(b) Performance using a 128 nodes per layer NN, asking as output the length of the shell structure and its diameter.



(c) Performance using a 64 nodes per layer NN, asking as output the length of the shell structure and its diameter.

Figure 6.33. Performance evaluation for the inverse problem giving as input n_x stress measured in 50 different angles.

Chapter 7

Conclusions

This chapter summarises the thesis work completed. Starting with an explanation of how artificial intelligence is nowadays employed by means of machine learning techniques and neural networks, passing through a chapter dealing with the theory of bending and normal force for a beam, the implementation of a neural network for the study of a cantilever beam, i.e. a clamped beam subjected to a concentrated force, was dealt with. The neural network was able to predict almost perfectly the maximum displacement that the final configuration of the beam undergoes with respect to the initial reference configuration, and the maximum stress to which the beam is subjected, by providing as input to the network only the length of the cantilever beam and the modulus of the force applied to it in terms of load.

After a chapter in which the Transfer Matrix Method and the theories used for the structural study of a shell and more generally of pressure vessels were described and explained, one moved on to the description of the architecture used for the study of the cylindrical type axle subjected to constant internal pressure loading along the longitudinal direction of the structure and varying in the circumferential direction. It was seen that this network succeeds in forming well in terms of predicting the stress n_x out of the plane of the cylinder cross-section, especially when 1, 2 or 4 output in terms of the angle of stress measurement on the circumference of the cross-section is required. By increasing the number of angles at which the stress measurement n_x is to be determined, the results gradually reduce the accuracy with which they are predicted by the neural network.

To further improve the results obtained, it might be interesting to carry out the same problem study as in this thesis work for the shell by increasing the size of the training data-set even more, in order to study deeper the inverse problem, improving the predicted results in terms of length and diameter of the shell.
The areas in which future research development from this thesis work appears interesting are: the study of shell-type structures characterised by a structure length greater than its diameter, for which the Vessel Theory no longer applies and for which it would therefore be necessary to introduce a new study using the Half-bend Theory in order to calculate stress and displacement accurately; the study and calculation of stress and displacement occurring in the bulkheads of shell-type structures.

In conclusion, finally, this thesis work has shown how the use of machine learning can guarantee the solution of structural mechanical analysis problems in a very short

time while not losing reliability in the prediction of results, potentially enabling structural and complex mechanical studies in a very short time compared to what would be required using conventional structural mechanical analysis tools such as finite elements.

Bibliography

[1] Hojjat Adeli and C Yeh. Perceptron learning in engineering design. *Computer-Aided Civil and Infrastructure Engineering*, 4(4):247–256, 1989.

[2] Jan-Hendrik Bastek and Dennis M. Kochmann. Physics-informed neural networks for shell structures. *European Journal of Mechanics - A/Solids*, 97:104849, 2023.

[3] NASA Marshall centre. Shell test tank nasa. `https://www.nasa.gov/exploration/systems/sls/shell-buckling-completed.html`, visited 03/07/2023.

[4] NASA Marshall centre. The Orbital ATK Cygnus cargo ship is seen after final approach to the International Space Station nasa. `https://www.nasa.gov/press-release/nasa-tv-to-broadcast-us-cargo-ship-departure-from-space-station-0/`, visited 03/07/2023.

[5] François Chollet. *Deep Learning with Python*. Manning, November 2017.

[6] W. Flügge. *Statik und Dynamik der Schalen, Statics and dynamics of shells*. Springer-Verlag, Berlin Heidelberg New York, 3. edition, 1981.

[7] W. Schnell G. Czerwenka. *Einführung in die Rechenmethoden des Leichtbaus II, Introduction to the computational methods of lightweight construction II*. Bibliographisches Institut, Mannheim-Wien-Zürich, 1970.

[8] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2017.

[9] K. Girkmann. *Flächentragwerke, Planar structures*. Springer-Verlag, Berlin Heidelberg New York, 6. edition, 1978.

[10] Jim Grossmann. Into the Fairing nasa. `https://www.nasa.gov/mission_pages/msl/multimedia/gallery/2011-10-25.html`, visited 03/07/2023.

[11] Yan Gu, Chuanzeng Zhang, and Mikhail V. Golub. Physics-informed neural networks for analysis of 2d thin-walled structures. *Engineering Analysis with Boundary Elements*, 145:161–172, 2022.

[12] P. Hajela and L. Berke. Neurobiological computational models in structural analysis and design. *Computers and Structures*, 41(4):657–667, 1991.

[13] S. Lenci. *Lezioni di meccanica strutturale*. Pitagora, 2009.

[14] Xin Liu, Su Tian, Fei Tao, and Wenbin Yu. A review of artificial neural networks in the constitutive modeling of composite materials. *Composites Part B: Engineering*, 224:109–152, 2021.

[15] P.A.M. Lopes, H.M. Gomes, and A.M. Awruch. Reliability analysis of laminated composite structures using finite elements and neural networks. *Composite Structures*, 92(7):1603–1613, 2010.

[16] NASA MSFC. Liquid Hydrogen Tank for the External Tank nasa. `https://www.nasa.gov/centers/marshall/history/gallery/liquid_hydrogen_tank.html`, visited 03/07/2023.

[17] Christian Emeka Okafor, Sunday Iweriolor, Okwuchukwu Innocent Ani, Shahnawaz Ahmad, Shabana Mehfuz, Godspower Onyekachukwu Ekwueme, Okechukwu Emmanuel Chukwumuanya, Sylvester Emeka Abonyi, Ignatius Echezona Ekengwu, and Okechukwu Peter Chikelu. Advances in machine learning-aided design of reinforced polymer composite and hybrid material systems. *Hybrid Advances*, 2:100026, 2023.

[18] H.-G. Prof. Dr.-Ing. Reimerdes. *Einführung in den Entwurf von Schalentragwerken, Introduction of the design of shell structures*. RWTH Aachen, Institute fur Strukturmechanik und Leichtbau, 2019-2020.

[19] Esteban Real, Chen Liang, David So, and Quoc Le. AutoML-zero: Evolving machine learning algorithms from scratch. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8007–8019. PMLR, 13–18 Jul 2020.

[20] Aniello Riccio, Salvatore Saputo, Andrea Sellitto, Angela Russo, Francesco Di Caprio, and Luigi Di Palma. An insight on the crashworthiness behavior of a full-scale composite fuselage section at different impact angles. *Aerospace*, 6(6), 2019.

[21] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[22] Leonardo FS Scabini and Odemir M Bruno. Structure and performance of fully connected neural networks: Emerging complex network properties. *Physica A: Statistical Mechanics and its Applications*, pages 128–585, 2023.

[23] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[24] Fernando V. Senhora, Heng Chi, Yuyu Zhang, Lucia Mirabella, Tsz Ling Elaine Tang, and Glaucio H. Paulino. Machine learning for topology optimization: Physics-based learning through an independent training strategy. *Computer Methods in Applied Mechanics and Engineering*, 398:115116, 2022.

[25] Jashanpreet Singh and Simranjit Singh. A review on machine learning aspect in physics and mechanics of glasses. *Materials Science and Engineering: B*, 284, 2022.

[26] A. Tesár and L. Fillo. *Transfer Matrix Method: (Enlarged and revised translation)*. Mathematics and its Applications. Springer Netherlands, 1988.

[27] Huu-Tai Thai. Machine learning for structural engineering: A state-of-the-art review. *Structures*, 38:448–491, 2022.

[28] R. Uhrig. The transfer matrix method seen as one method of structural analysis among others. *Journal of Sound and Vibration*, 4(2):136–148, 1966.

[29] Valery V. Vasiliev and Evgeny V. Morozov. Chapter 12 - composite pressure vessels. In Valery V. Vasiliev and Evgeny V. Morozov, editors, *Advanced Mechanics of Composite Materials and Structures (Fourth Edition)*, pages 787–821. Elsevier, fourth edition edition, 2018.

[30] Valery V. Vasiliev and Evgeny V. Morozov. Chapter 8 - equations of the applied theory of thin-walled composite structures. In Valery V. Vasiliev and Evgeny V. Morozov, editors, *Advanced Mechanics of Composite Materials and Structures (Fourth Edition)*, pages 575–590. Elsevier, fourth edition edition, 2018.

[31] E. Ventsel and T. Krauthammer. *Thin Plates and Shells: Theory: Analysis, and Applications*. CRC Press, 2001.