

# Design of bone-like spinodoidal implants

Investigating the application of deeplearning for designing anisotropic metamaterials for bone implants

WB2332: Project Materiaalkunde

Wesley Nijhuis

Olivier Snorn

Kes Olling

Hugo Schravesande

Julen Duivesteijn Perez

# Design of bone-like spinodoidal implants

Investigating the application of deeplearning for designing anisotropic metamaterials for bone implants

by

Wesley Nijhuis  
Olivier Snorn  
Kes Olling  
Hugo Schravesande  
Julen Duivesteijn Perez

Student Name	Student Number
Wesley Nijhuis	4965590
Olivier Snorn	5645883
Kes Olling	5515637
Hugo Schravesande	4771796
Julen Duivesteijn Perez	student number not contributed

Instructor: S. Kumar  
Teaching Assistant: F. Van Raalte  
Project Duration: April, 2023 - June, 2023  
Faculty: Faculty of Mechanical Engineering, Delft

Cover: Image by freepik.com  
Style: TU Delft Report Style, with modifications by Daan Zwanenveld



# Preface

This report describes a project from the TU Delft. This project is about using machine learning to reconstruct a bone structure. It will help us learn more about the programming world and the medical field. A certain knowledge of programming and material structures is required to understand this paper. If you are interested in the code we used, we refer you to Jupyter notebook that will be delivered with this paper. If you want to take a closer look on the sources we used, you can find these in the chapter 'References'. At last we want to give a special thanks to Sid Kumar and Frank van Raalte for their inspiration and support.

*Wesley Nijhuis*

*Olivier Snorn*

*Kes Olling*

*Hugo Schravesande*

*Julen Duivesteijn Perez*

*Delft, December 2025*

# Summary

This paper examines a promising method of inversely designing materials with interesting capabilities and applications in, for example, the medical industry.

After the introduction a case study is presented where a process is outlined that could be used to generate patient-specific bone implants. These bone implants are designed using several neural networks which create a set of data that can be used to computationally construct a material with very specific stiffness properties. Such materials with a porous microstructure are called metamaterials. In this paper spinodoidal metamaterials will be used, because these are comparatively similar to actual bone structure and can be easily generated.

In order to train the neural networks, good data must be used. Therefore, a large provided dataset was thoroughly examined. A large number of experiments were executed in order to evaluate whether this dataset could be further optimized. Special attention was paid to the distribution of certain categories of data within the dataset.

The neural networks are in fact two algorithms that work together to produce design data from a design requirement. First, there is a forward neural network which predicts the behaviour of any particular design in order to evaluate whether or not a design actually meets its stated requirements. Second, an inverse neural network uses the forward neural network to train itself into creating accurate design data for each design requirement.

The duo of neural networks is capable of producing a metamaterial with uniform properties over its volume. In order to resemble actual bones more, it was chosen to explore an interpolation technique that is able to generate a single metamaterial with properties that differ for each location within the metamaterial.

This paper provides results for all experimentation and explorations in a dedicated chapter. Next, a discussion will provide all opportunities for further improvement of the results and hits for further research. Lastly, a conclusion is given.

# Contents

<b>Preface</b>	i
<b>Summary</b>	ii
<b>1 Introduction</b>	1
<b>2 A proposed method as a case study</b>	2
2.1 Data acquisition . . . . .	2
2.2 Creating local design parameters for spinodoid reconstruction . . . . .	2
2.3 Designing an implant with spatially variant topology . . . . .	2
2.4 Manufacturing the implant . . . . .	3
<b>3 Spinodoid Metamaterials</b>	4
<b>4 Data Exploration</b>	6
4.1 Provided data . . . . .	6
4.2 Examining dataset . . . . .	6
4.3 Altering dataset . . . . .	8
<b>5 Designing the Neural Network</b>	9
5.1 Forward Neural Network Design . . . . .	10
5.2 Inverse Neural Network Design . . . . .	10
5.2.1 Loss function . . . . .	10
5.2.2 Feature shifting . . . . .	11
<b>6 Spatially variant topologies</b>	12
6.1 interpolation in three dimensions . . . . .	12
<b>7 Results</b>	14
7.1 Data exploration . . . . .	14
7.2 Forward neural network . . . . .	15
7.3 Inverse neural network . . . . .	16
7.4 Spatially variant topologies . . . . .	18
7.5 Bone structure . . . . .	20
<b>8 Discussion</b>	21
8.1 Data exploration . . . . .	21
8.2 Loss function analysis . . . . .	21
8.3 Shifting of design set predictions . . . . .	21
8.4 Three-dimensional interpolation . . . . .	21
8.5 Bone structure . . . . .	22
<b>9 Conclusion</b>	23
<b>References</b>	24

# 1

## Introduction

A damaged hip joint can be very painful and reduces mobility of your body. For this problem a popular solution is hip replacement surgery.

According to Harvard Health Publishing Resources, in the US alone, surgeons perform 330,000 hip replacements each year [13]. So a small positive change per operation (materials, comfortable integration in the body) would make a big difference overall.

It is a surgical procedure in which a broken/diseased part of the hip joint is replaced with an artificial part, in our case made of a spinodoid metamaterial. The main and biggest material to focus on is the Wagner cone stem - which is conventionally made from a titanium alloy. In the hip replacement procedure, the Wagner cone stem would align with the femur or thigh bone axis and connect to the pelvis with a ball and socket connection. Using calcium phosphate cements the spinodoid could be glued or attached inside the femur.

This report will begin with information on the problem and the steps taken to get to the result of this problem. After that, a more detailed introduction will be given on spinodoid metamaterials. The next chapter will be about data exploration, to get a better idea on the data and how this might be implemented to get useful results. Chapter 5 is going to be about the used neural network, with a deeper look on the theory of it and its design. After that, there will be more information on the design set interpolation. At last, the results will be shown and there will be a discussion and conclusion.

# 2

## A proposed method as a case study

To investigate how to apply the newly developed method of designing bone-like anisotropic materials, a complete method of creating a new implant is proposed. The goal of this is not to suggest a new method of fabricating implants, but rather to identify possible obstacles in applying the design method proposed by Kumar et. al. [9].

The method that is proposed consists of three elements: data acquisition, data processing and finally the design of spatially variant topologies. The main idea of the process is to create a grid of local desired stiffness matrices, for which a grid of topologies can be designed. Interpolating between topologies will result in an implant design with a spatially variant topology that emulates a patients bone properties in terms of density and stiffness.

### 2.1. Data acquisition

The first step in the proposed method is creating a grid of local stiffness matrices. There is a plethora of ways to determine mechanical properties of bone. One way to accurately measure local mechanical properties of bone is by microbeam testing [4][14]. However, to minimize the impact of measuring the bone stiffness on the patient, a noninvasive method was chosen; micro-CT scanning.

To create the grid of local stiffness matrices a complete micro-ct scan of the to be replaced bone is made. From this complete micro-ct scan, local topology samples can be extracted by using cubic extractions from the ct-scan. It is important that these local topology samples are small enough to have an even topology, but large enough to be able to give an accurate representation of the stiffness characteristics of the topology. From these local samples, FEM analysis can be performed to obtain the local stiffness matrices [8]. To make a prediction on the time required to run these FEM analyses for every node, reported run times from Kumar et al. were taken [9]. Assuming the run time of a single FEM analysis takes around 5 minutes, retrieving the stiffness matrices for a couple of thousand nodes would take somewhere in the range of 7-14 days. Usually though, waiting times for these kinds of surgeries can take months and the proposed method will thus not create a bottle neck [3] [5].

### 2.2. Creating local design parameters for spinodoid reconstruction

For the reconstruction of the bone samples, spinodoid topologies are used (see chapter 3). These spinodoid topologies enable for the reconstruction of anisotropic stiffness matrices. Using an algorithm that could design such a spinodoid material for a given stiffness matrix, it is possible to create a designed topology for every single gridpoint. This algorithm is obtained by using a surrogate neural network model discussed in chapter 5.

### 2.3. Designing an implant with spatially variant topology

To actually construct an implant from the constructed grid of designed topologies, a method of interpolation is needed. With the methods discussed in chapter 6 and 8, it is possible to create three-dimensional interpolations of topologies. After interpolation, a three-dimensional model of a potential topology is obtained, which accurately approximated the desired bone properties in terms of stiffness and density.

## 2.4. Manufacturing the implant

The porous nature as well as immense complexity of the resulting design necessitates the use of Additive Manufacturing (AM), generally known as 3D printing, as a production technique. Regular Subtractive Manufacturing (SM) processes cannot create the internal spaces that are crucial to the designs function. Several AM techniques have already been proven in the manufacturing of a variety of bone implants.

Once the design is realized it should be exported into a file format that is used by the software which prepares design for manufacturing. In this process, geometrical information from the model file will be converted in machining actions to be performed. Once the production machine is properly set up the manufacturing process will continue autonomously, except for the part removal and post-processing after manufacturing is complete. Total production time is generally completed within a day, and should not be considered a bottleneck in the overall process.

One of the many possible AM process for the production of medical implants is Electron-Beam Melting® (EBM®). This process usually takes in a titanium alloy such as Ti–6Al–4V to create medical-grade implants. These implants often require some post-processing steps and sterilization. [15]

To demonstrate the manufacturing process, a hip prosthesis was produced in-house using a regular FFF 3D-printer, specifically a Bambu Lab P1P. For comparison, another unit was produced with the femoral stem replaced by a porous structure of the same shape that was obtained from an interpolated topology that was created in-house.



**Figure 2.1:** A regular hip prosthesis and one with a spinodoid femoral stem

# 3

## Spinodoid Metamaterials

Topologies which are encountered in cancellous bone materials originate from spinodal decomposition. These topologies can be obtained by simulating the computationally expensive Cahn-Hilliard phase separation processes [2][9].

An alternative, spinodoid topologies can be utilized. Spinodoid materials can create topologies with anisotropic mechanical properties, without the need of performing these time-dependent Cahn-Hilliard simulations.

Spinodoid topologies can be generated using the GRF-approach (Gaussian random field), see equation 3.1. These describe how a phase field (with the three main directions:  $\theta_1, \theta_2, \theta_3$ ) can be described using a large number of standing waves sampled within cones of width  $\theta_1, \theta_2, \theta_3$ . Here,  $\phi(\underline{x})$  is the phase field where  $\underline{x}$  is the position of a phase, where  $\underline{x} \in \mathbb{R}^3$ .  $N$  is the amount of standing waves being superpositioned and  $\beta$  is the wavenumber. Gamma ( $\gamma_w \in U([0, 2\pi])$ ) is the phase angle at wave vector  $w$ . Here the  $U([0, 2\pi])$  means the continuous uniform distribution from 0 to  $2\pi$ .  $\underline{n}_w$  is the wave direction as a unit vector of length 1.

$$\phi(\underline{x}) = \sqrt{\frac{2}{N}} \sum_{w=1}^N \cos(\beta \underline{n}_w \cdot \underline{x} + \gamma_w) \quad (3.1)$$

We can indicate the presence of material at position  $\underline{x}$  in the spinodoid by using a binary function  $X$  defined in the equation directly below. This conditional equation is 1 in the case of a present material and 0 in case of a void or no present material.

$$X(\underline{x}) = \begin{cases} 1, & \text{if } \phi(\underline{x}) \leq \phi_0 \\ 0, & \text{if } \phi(\underline{x}) > \phi_0 \end{cases}$$

In the equation above,  $\phi_0$  is given, this scalar is actually dependent on an error function where a relationship with the density is observed. This function is defined in equation 3.2.

$$\phi_0 = \sqrt{2} \cdot \operatorname{erf}^{-1}(2\rho - 1) \quad (3.2)$$

These computations correspond with a density. When low density values are used, discontinuous topologies will be created. To prevent this from happening a lower bound for the density is to be placed at 0.3. For the same reason, a lower bound for the minimum cone angle is to be placed at 15 degrees. Using these bounds, continuous topologies are enforced. These design parameter bounds can also be observed in equation 5.5.

A significant advantage of spinodoid materials is that they have anisotropic stiffness, which makes its design space significantly larger than simple isotropic materials. Anisotropic stiffness is the ability of the material to have different stiffnesses in different directions. Another remarkable advantage of spinodoid materials are that they are tunable, which can make these materials behave exactly like the material you want them to mimic. That is of course, after you tune the spinodoid that desired way.



**Figure 3.1:** One of the many possible applications of spinodoid metamaterials

Due to the large design space of these spinodoid materials there are a lot of applications these materials can be assigned to: bone implants, shoes, cars, etc. The properties of each spinodoid case will change depending on which structure/mechanism it will be implemented in.

# 4

## Data Exploration

### 4.1. Provided data

This project could not be undertaken without a good dataset. Such a dataset must include a large number of datapoints consisting of design information and the resulting stiffness that will be achieved by a topology constructed with this design information. Such design information consists of the relative density of the topology as well as the maximum wave vectors along each axis, which are often called *theta*'s. This design information is used to construct topology elements which are evaluated using Finite Element Analysis (FEA) in a way that results in a symmetric stiffness matrix that consists of just six unique elements. This is the same stiffness matrix as the one described in 2.2.

A dataset such as described above was kindly provided by the project supervisor [10]. This dataset consists of 21299 datapoints and none of these appear exhibit strange behaviour. The data was nicely scaled and no datapoints overtly contradict each other.

### 4.2. Examining dataset

Before the provided dataset could be used, it was thoroughly examined so it would be possible to correctly interpret the outcomes resulting from the algorithm using this dataset. The first point of order was evaluation of all design parameters, where distribution was examined.

The distribution of relative density within the dataset appears to be rather uniform between the lower bound of 0.3 and the upper bound of 0.9. These bounds exist to ensure the resulting topology will be a single body with anisotropic stiffness properties.

The distribution of the maximum wave vectors within the dataset deserves some additional attention. It is to be expected that either 0, 1 or 2 of these '*thetas*' can be zero, which reveal the topology type of the datapoint. Cubic topologies have no maximum wave vectors of 0, columnar topologies have 1 maximum wave vector if 0 and lamellar topologies have two maximum wave vectors of 0. The original distribution of lamellar, columnar and cubic topologies is [8700, 7199, 5400], or 41%, 34% and 25% respectively. This distribution of topologies within the dataset follows no obviously discernible pattern.

A pairplot of the unique nonzero values of the stiffness matrix revealed some correlations that are to be expected given the features of the dataset that was given. No concerning behaviour was observed.

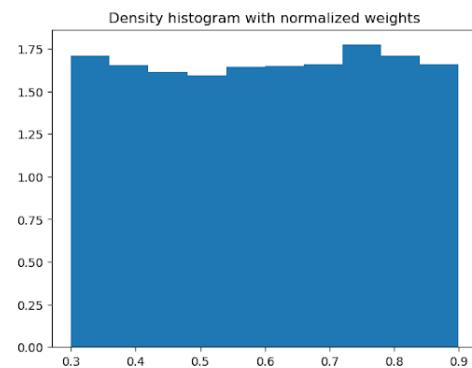


Figure 4.1: Relative density histogram from original dataset

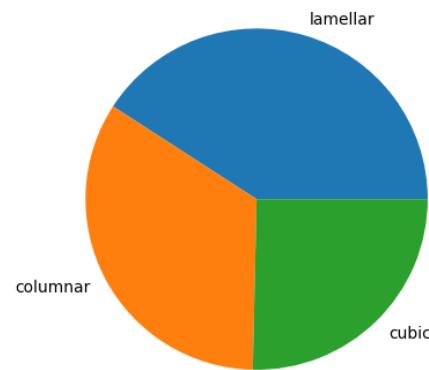


Figure 4.2: Topology distribution from original dataset

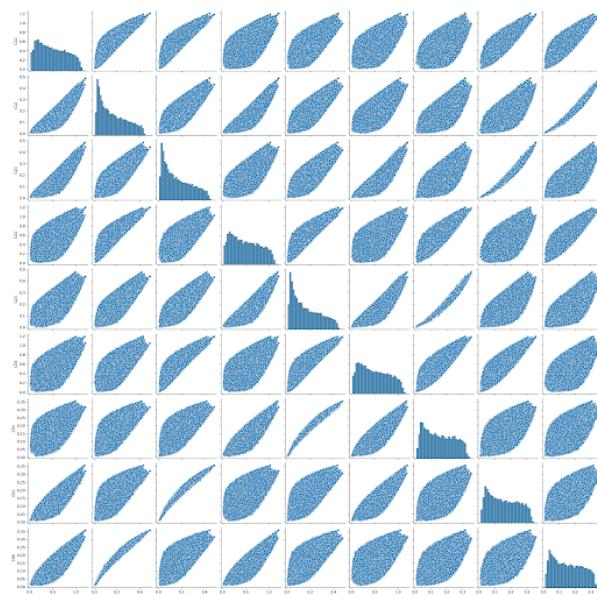


Figure 4.3: Pairplot of unique stiffness matrix entries from original dataset

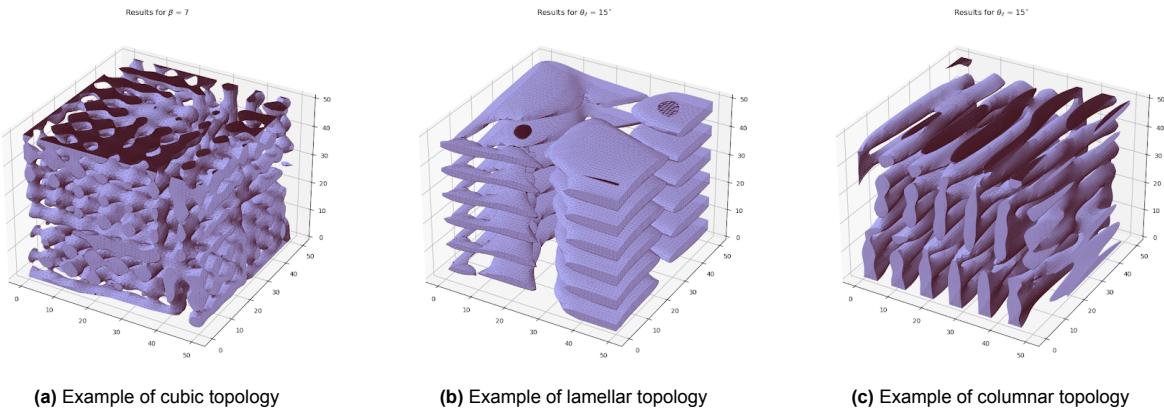


Figure 4.4: An example for each topology type

### 4.3. Altering dataset

The provided dataset yielded good results at initial testing. At that point, it was unclear whether the chosen distribution between topologies had a significant effect on the accuracy of the predicted stiffness matrix. This was examined by testing several simple distributions that each could hypothetically improve the final average  $R^2$ .

To accomplish this, first a subset of the original dataset was made to act as testdata. This data remained unmodified for each test case.

Second, the trainingsdata was made by creating multiple subsets of the remaining data. These subsets all follow specific distributions that are of interest and are each the same size. This size is smaller than the amount of remaining data, because for each distribution that is different from the original distribution some datapoints need to be discarded. The distributions are chosen in such a way that the size of the trainingsdata was never less than 75% of the size of the amount of data that is left after defining the testdata. The testdata was randomized and all data was normalized before entering the neural networks.

Third, the algorithm was adapted so that each neural network was redefined for each new distribution to be tested. These changes made it possible to test a list of possible distributions against each other automatically. The final average  $R^2$  and minimum  $R^2$  were listed for each distribution and this was used to determine accuracy.

A couple of simple distributions were hypothesised to yield a better accuracy. These were [1, 1, 1], [6, 4, 3] and [6, 3, 2] for lamellar, columnar and cubic topologies. The first distribution might work well because too much focus on a single distribution might reduce the overall accuracy. The second distribution might improve accuracy by compensating for the zero values in maximum wave vectors. The third distribution takes this a step further by also compensating for the density, so that the total amount of datapoints per topology in each subset was identical.

Also, 10 variations on the original subset were constructed. This was accomplished by multiplication of each item in the original distribution, being [8700, 7199, 5400], with different randomized numbers in between 0.8 and 1.2. Then, if this distribution leads to an amount of topologies that is higher than the amount of topologies that is available, the maximum number of available topologies is used. Some of these 10 variations yielded better accuracy than the original distribution, and so the experiment was expanded to add 100 variations to the distributions to be tested and compared.

# 5

## Designing the Neural Network

After retrieving the data on the mechanical properties of the to be replaced bone, topologies can be designed that accurately emulate mechanical properties of bone in terms of stiffness and density. As discussed in chapter 3, we will create these topologies by generating Gaussian Random Fields. An important aspect of this is that it is computationally inexpensive to create these spinodoid topologies with respect to simulating Cahn-Hilliard phase separation processes[9]. This makes creating a large data set for the training of neural network feasible.

The application of a neural network is to enable the creation of the inverse designs of these topologies. A neural network that could construct a set of design parameters  $\underline{\Theta}_i$  from a given local stiffness matrix  $S_i$  can be used to create the local topologies that can recreate the aforementioned mechanical properties of the patients bone.

Choosing a design parameter set that reconstructs the desired stiffness as accurately as possible presents an optimization problem. Furthermore, an initial naive attempt of minimizing the MSE between the inverse neural network predictions for a given stiffness matrix  $S_i$  and the associated design parameter set  $\underline{\Theta}_i$  does not yield a solution. This problem is ill-posed: multiple design parameter sets can achieve the same stiffness matrix. Such an ill-posed optimization problem is given in equation ??, where  $\mathcal{F}$  is the forward neural network.

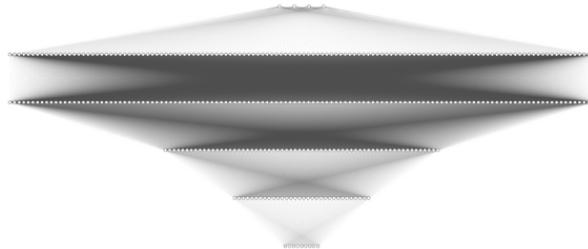
The solution to this problem of ill-posedness lies in a forward prediction. Going from a design parameter set  $\underline{\Theta}_i$  to an associated stiffness matrix  $S_i$  is a well-posed problem, as every design parameter set has a unique stiffness matrix associated with the generated topology. It is therefore possible to create a well-posed problem for the inverse design of design parameters by coupling the inverse neural network to a forward neural network. This is known as a surrogate model and bypasses computationally expensive FEM analyses[16][9][17]. This is in practice done by using a pre-trained forward neural network for the loss optimization of the inverse neural network. In the rest of this chapter, the design of both the forward neural network and the inverse neural network will be discussed as will this before mentioned loss optimization.

### Choice of optimizer

Before using an optimization algorithm, literature on the optimizers was studied to make a well based choice for this project. The goal of an optimization algorithm is to find the optimal solution that minimizes or maximizes a given objective function. Three optimizers of interest were Momentum, RMSProp and Adam. Momentum is an optimizer based on the physics phenomenon, so momentum/kinetic energy is accumulated going down a slope of a function and this momentum can be enough to escape local minima. Escaping these local minima can be beneficial for finding one global minimum. RMSProp is an optimization algorithm where the gradients at a given point of the function are summed and a decay factor is added, as to slow down at a minimum. The optimizer Adam is a combination of the two optimizers mentioned above, taking the best of both worlds of not being trapped in local minima and coming at a halt at a global minimum. Adam is a popular choice in deep-learning algorithms in recent years and achieves accuracy fast. For this project, Adam was chosen as the optimizer for the neural network.

## 5.1. Forward Neural Network Design

By training a forward neural network, we can accurately predict the stiffness vectors with the design vector as input. The training data contains 85% of the given dataset and the other 15% is used as test data. First, a neural network was made. The parameters that were used were mainly taken from the paper of Kumar et al. [9]. To optimize the neural network, we experimented with the learning rate and the number of epochs. These different outcomes can be seen in chapter 7. With our optimal architecture overfitting or underfitting does not occur, since both the test loss history and the training loss history are quite low. The running time is also very low. The whole architecture of the forward neural network can be seen in figure 5.1.



Hyperparameter	Design choice
Hidden layer dimensions	128,128,64,64,32,32
Optimization algorithm	Adam
Batch size	32
Numbers of epochs	200
Learning rate	0.0001
Loss function	MSE

Figure 5.1: Architecture f-NN

## 5.2. Inverse Neural Network Design

As discussed in the beginning of this chapter, the goal of the inverse neural network is to create a design parameter set  $\Theta_i$  for each and every node in the generated grid, using the calculated stiffness matrices  $S_i$ . To deal with the fact that minimizing the difference between the reconstructed design parameters and the given design parameters is an ill-posed problem, a forward neural network was introduced in section 2.1.

The architecture used was largely directly adapted from the paper of Kumar et al. [9]. The used architecture of the inverse neural network is illustrated in figure 5.2. The main things that were changed were the implementation of the loss function and the feature scaling and shifting.

Further literature studies into

### 5.2.1. Loss function

As mentioned in the beginning of this chapter, the forward and inverse neural networks are coupled by implementing the forward neural network in the loss function of the inverse neural network. Such a loss function can be given as:

$$\text{Loss} = |\mathcal{F}(\mathcal{I}(S_i)) - S_i|^2 \quad (5.1)$$

Here  $\mathcal{I}$  is the inverse neural network. Further investigation into the lost function via literature studies yielded new insights into the loss function. Literature suggested that for these kinds of inverse problems, an extra penalty term can be added to improve performance [9][7][12]. For this project it was decided to add the ill-posed loss function as a penalty term. This term acts in the way of a regularization term. Through experimentation of the implementation of this term in was found that activating this is the later stages of training aided in the convergence of the neural network. We hypothesise that this is achieved

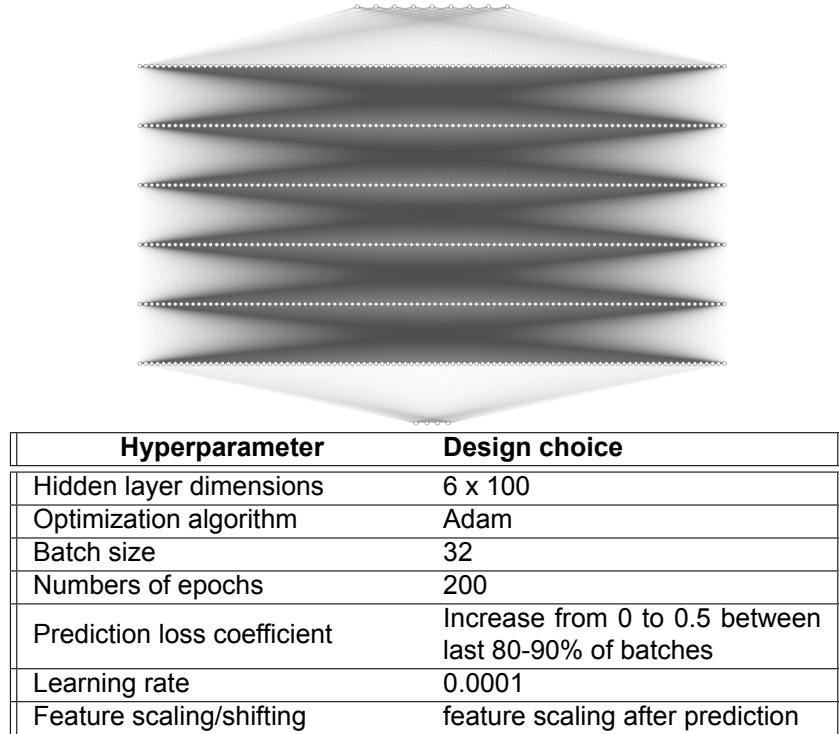


Figure 5.2: Inverse neural network architecture

by the ill-posed loss term by forcing the neural network to find different solution sets, some of which can result in better reconstruction. The final loss function is defined as following:

$$Loss = ||\mathcal{F}(\mathcal{I}(S_i)) - S_i||^2 + \lambda ||\mathcal{I}(S_i) - \underline{\Theta}_i||^2 \quad (5.2)$$

with

$$\lambda = 0 + \frac{1}{2}(3 - p + \frac{pn}{N}) \quad (5.3)$$

for

$$\frac{N(p-3)}{p} \leq n \leq \frac{N(p-2)}{p} \quad (5.4)$$

where  $N$  is to total number of batches,  $n$  is the batch number and  $p$  is a chosen parameter that influences where the penalty term will be activated en how long the penalty term will stay activated.

In practice this will result in a penalty term that will discretely linearly increase in a tunable interval.

### 5.2.2. Feature shifting

To make sure that only realistically producible topologies were generated, Kumer et al. 2020[9] proposed the following restrictions on the design paramaters:

$$\rho_i = \begin{cases} 0.3, & \text{if } \rho < 0.3 \\ 0.9, & \text{if } \rho > 0.9 \end{cases}, \quad \Theta_{i,k} = \begin{cases} 0^\circ, & \text{if } \Theta_{i,k} \leq 7.5^\circ \\ 15^\circ, & \text{if } \Theta_{i,k} \in (7.5^\circ, 15^\circ) \\ 90^\circ, & \text{if } \Theta_{i,k} > 90^\circ \end{cases}, \quad (5.5)$$

For this project, these restrictions where enforced by shifting unsatisfactory values via a simple restriction enforcing algorithm in post processing. This means the shifting of features was done after the trained neural network made a prediction of a set of design parameters, but before the values would be returned to the user.

# 6

## Spatially variant topologies

Being able to smoothly spatially vary the topology of the architecture is crucial, as this will avoid any regions of stress concentration. As discussed in chapter 3, spinodoid topologies are generated using Gaussian Random Fields. We can use these GRF to generate topologies with desired mechanical properties by using design parameters yielded from the inverse neural network. Spatially varying topologies can be created by using a superposition of multiple GRFs [17]:

$$\tilde{\phi}(\underline{x}) = \sum_{i=1}^k \mathcal{A}_i(\underline{x}) \phi_i(\underline{x}) \quad (6.1)$$

where  $\mathcal{A}_i$  is determined by an interpolation function.

There are many possible ways of interpolating these GRFs. For a one-dimensional case,  $\mathcal{A}_i$  could be the Lagrange basis terms. This would yield linear interpolation between two data points:

$$\tilde{\phi}(\underline{x}) = L_0(\underline{x})\phi_0(\underline{x}) + L_1(\underline{x})\phi_1(\underline{x}) \leftarrow \begin{cases} L_0(\underline{x}) = \frac{\underline{x}-x_1}{x_0-x_1} \\ L_1(\underline{x}) = \frac{\underline{x}-x_0}{x_1-x_0} \end{cases} \quad (6.2)$$

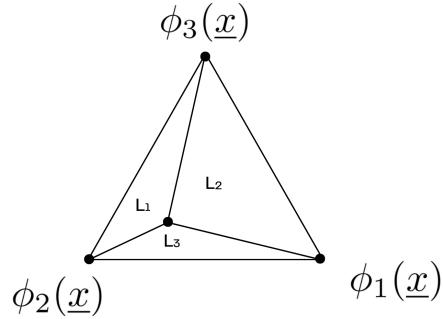
Here  $\phi_0(\underline{x})$  and  $\phi_1(\underline{x})$  are GRFs at adjacent grid points. An important characteristic of GRFs is that any well behaved superposition of GRFs create smooth transitions between grid points.

### 6.1. interpolation in three dimensions

To design a topologically varying implant, a method is needed that can interpolate the Gaussian Random Fields between the grid points in three dimensions. To generate spatially varying topologies in 3-D many methods can be used such as triangulation based methods or radial basis function methods [1][9]. In this section, a linear three-dimensional interpolation method based on barycentric coordinates is proposed for generating these sparially varying topologies.

Characteristics of the chosen interpolation method can have a significant impact on the quality of the designed implant. In the design of implants it is crucial that topologies vary smoothly to avoid any stress concentrations. Normally this would exclude linear interpolation methods for the design of spatially variant topologies. However, since superpositions of GRFs are used to generate the topologies, linear interpolation methods actually generate smooth transitions in topologies as seen in 7.10.

As discussed in 2.1, calculation time of the FEM analysis of the bone do not have an impact on the time it takes to implement the implants, due to long waiting times for ORs and such [check dit terug met 2.1]. Real life implementation of the proposed method of designing implants is not very limited by the amount of data points. It is therefore very plausible that a linear interpolation method will yield a high accuracy, while being less prone to artifacts compared to other interpolation methods[1]. With the assumption that a large set of data points on the stiffness of the to be replaced bone will be available, the following interpolation method is proposed.



**Figure 6.1:** Barycentric coördinates in two dimensions

### Barycentric interpolation

Barycentric coördinates are used extensively in computer graphics and computational mechanics. It enables for easy interpolation between grid points in a mesh of triangles or tetrahedrons through coördinate transformation. In Barycentric coördinate systems, the coördinates are actually the areas of volumes between grid points in these discrete triangular or tetrahedron elements [6]. For the interpolation between GRFs, the interpolation factor are chosen as the Barycentric coördinates. A visualisation of the method is given in figure 6.1. An elementwise interpolation method is now established for an  $j^{th}$  element.

$$\tilde{\phi}_j(\underline{x}) = \sum_{i=1}^k \mathcal{A}_i(\underline{x}) \phi_i(\underline{x}) \longrightarrow \sum_{i=1}^{d+1} L_i(\underline{x}) \phi_{j,i}(\underline{x}) \quad (6.3)$$

where  $d$  is the number of dimensions of the element (2 for triangles, 3 for tetrahedrons).

# 7

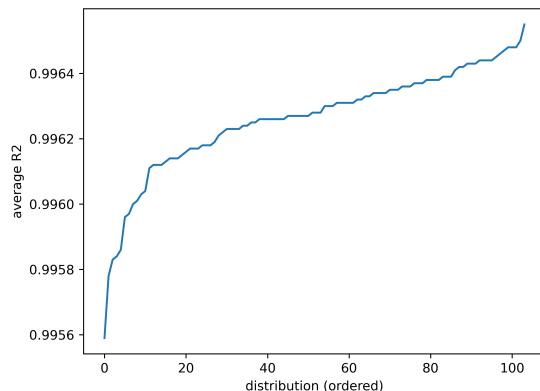
# Results

## 7.1. Data exploration

Our proposed simple distributions performed very poorly compared to the original distribution, which scored above average on both average and minimum  $R^2$  value of the stiffness matrix. Surprisingly, about one-third of variations outperformed the original distribution. This improvement was however minimal and did not significantly affect the overall accuracy of the algorithm.

No discernible pattern was found in the distributions that scored best on  $R^2$  values for the stiffness matrix. For this reason, it is impossible to substantiate the reason for some adapted distributions to outperform the original distribution. There was no incontrovertibly best distribution found but it is certain a better distribution exists because the large amount of tested distributions suggests the result do not follow the behaviour expected from random errors.

In the below figure the average  $R^2$  for each tested distribution is given, ordered by  $R^2$ . The worst distribution, measured by average  $R^2$ , is to the left and the best is to the right. The original distribution is number 65, when ordered by average  $R^2$ . Please note that these average  $R^2$  values result from a subset of all data and therefore is generally lower than the  $R^2$  values mentioned in other parts of this report, where all data is used.

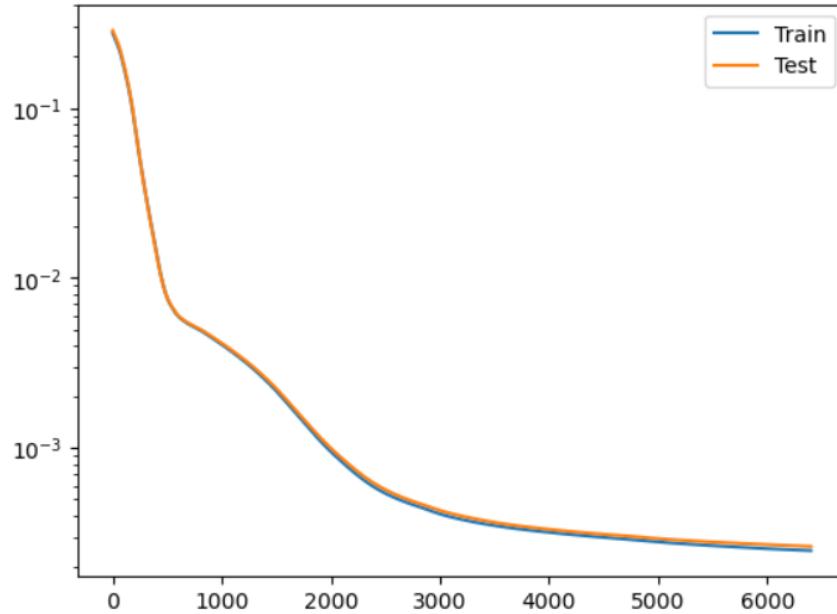


**Figure 7.1:** average  $R^2$  (ordered) for each tested distribution

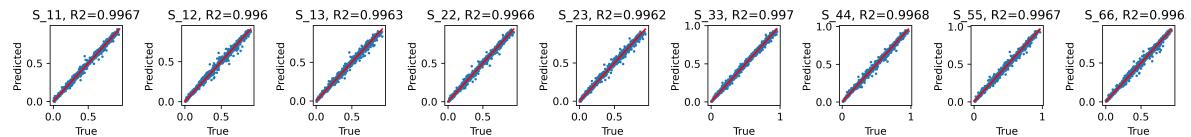
There is some improvement in accuracy to be had in selecting data distribution. However this requires much time, effort and computing power and these resources are likely better spend on improving other parts of the process if improvement of accuracy is the main goal.

## 7.2. Forward neural network

In figure 7.2, the final loss history can be seen. In figure 7.3 the  $R^2$  values are shown with an average  $R^2$  value of 0.9965.



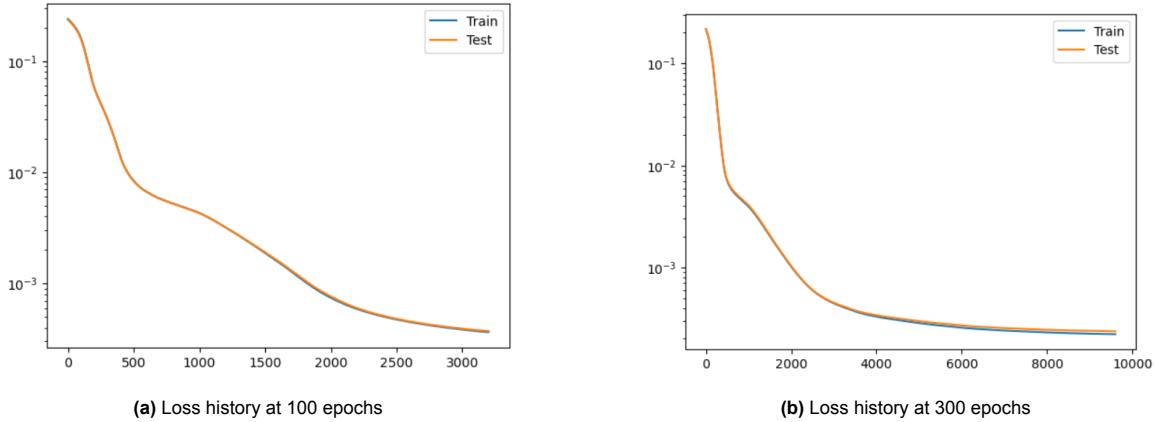
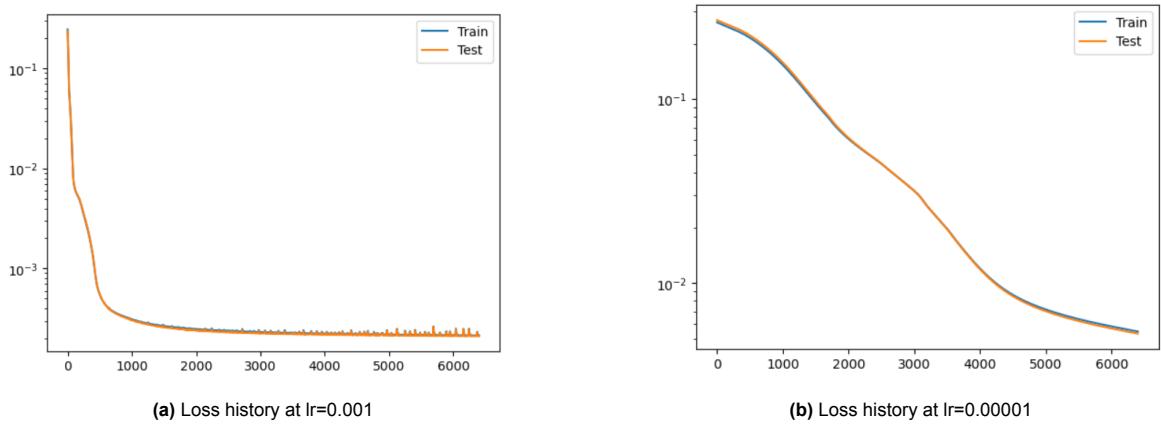
**Figure 7.2:** Loss history f-NN



**Figure 7.3:**  $R^2$  values f-NN

The graph converges to a global minimum and there is no overfitting or underfitting. In figure 7.4a, however, the number of epochs was set to 100 and in figure 7.4b to 300. With the number of epochs at 100, the line is still declining, so it has not reached the global minimum yet. But with the number of epochs at 300, the global minimum is reached long before the end, which means the code takes an unnecessarily long time to run. This made us come to the conclusion that the number of epochs is optimal at 200.

In figure 7.5b a learning rate of 0.00001 is used. As can be seen, it takes very long for the line to converge, since it takes smaller steps to look for a minimum. But in figure 7.5a the learning rate is equal to 0.001. Here the line converges very quickly. This means that we might skip over the optimal solution. The optimal learning rate lies in between, so this was chosen as 0.0001.

**Figure 7.4:** f-NN loss history with varying number of epochs**Figure 7.5:** f-NN loss history with varying learning rates

### 7.3. Inverse neural network

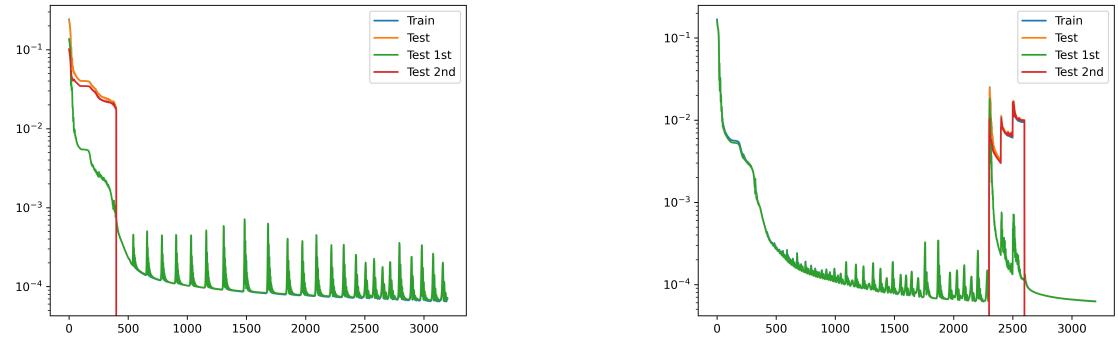
For the implementation of  $\lambda$  in the loss function, a value of  $p = 10$  seemed to give the best results. This means that  $\lambda$  will linearly increase for a number of batches between the last 80-90% of the training. Implementation at later stages seemed to interfere with the ability of the neural network to predict values of  $\Theta$  that accurately reconstruct the desired stiffness matrix. This way of implementing the ill-post lost function also seemed to produce better results than implementing a constant value for  $\lambda$  at the first  $\frac{1}{p}^{th}$  part of the training, similar to the implementation seen in the paper of Kumar et al. [9]. A comparison of these two strategies can be seen in figures 7.6 and 7.7.

From the resulting predictions the average value of  $R^2$  values was calculated. These predictions are illustrated in figure 7.7. The early adoption of the penalty term yielded an average  $R^2$  value of 0.9988. The later adoption yielded an average  $R^2$  value of 0.9990. This gains are incremental, but seem to be consistent when the training is rerun.

Also notable is the stabilisation of the loss function after the implementation of the penalty term in the later stages. This is discussed further in the discussions 8.

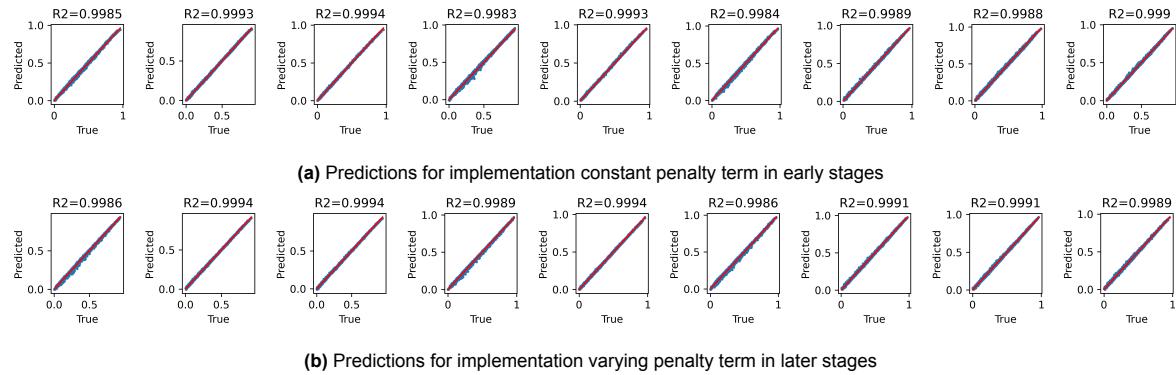
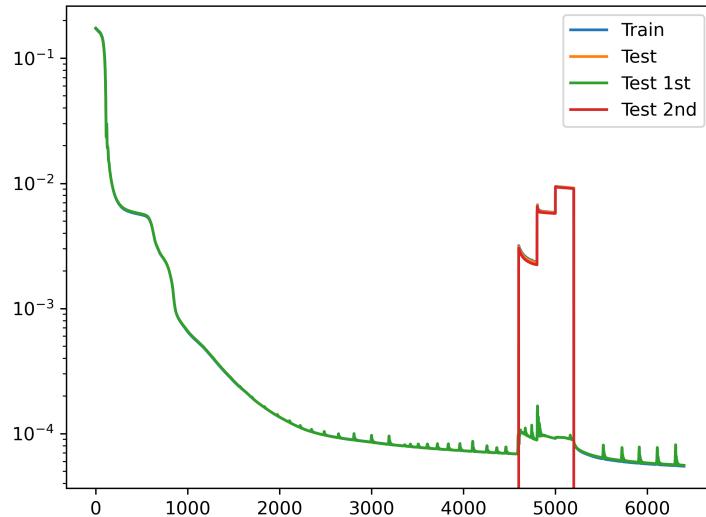
In figures 7.8 and 7.9 the final loss history and predictions are illustrated respectively. Though amplified by the log scale, it can be seen the right term seems to converge relatively quickly. During the batches the right term was activated, some interesting behaviour can be observed. This could be explained by the hypothesis from chapter 5. It was proposed that the ill-posed term would force the neural network to find some new previously undiscovered paths to better solutions.

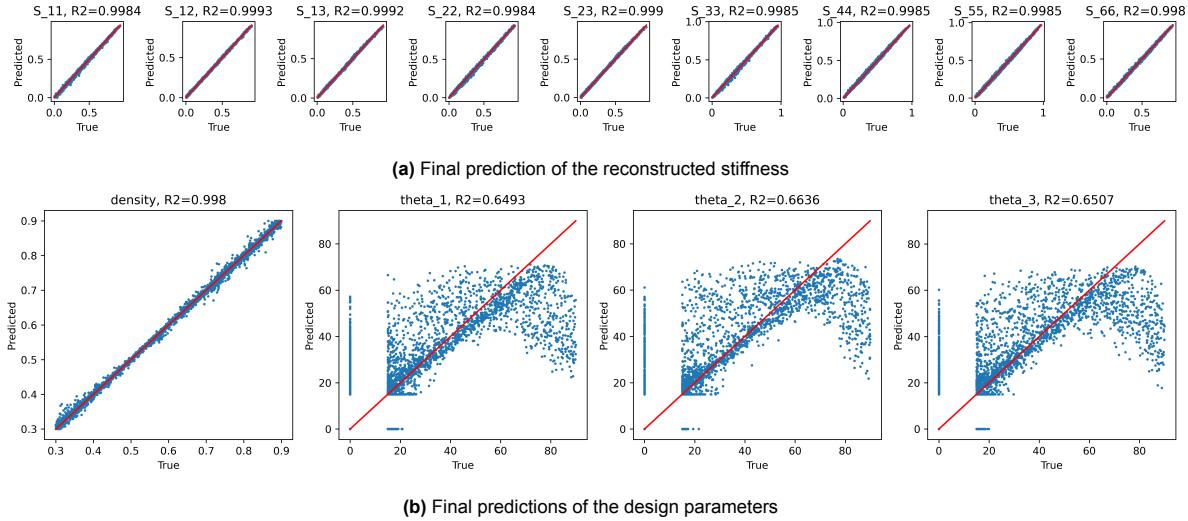
From the  $R^2$  values found in figure 7.9, an average  $R^2$  value of 0.999 was obtained. Also the  $R^2$  values for the density are satisfactory at 0.998. Looking further into the results it can be seen that the predictions for  $\Theta_i$  are not accurate at all. This supports the initial claim that there exist multiple sets of design parameters which result in the same stiffness matrix.



(a) Loss history for implementation constant penalty term in early stages

(b) Loss history for implementation varying penalty term in later stages

**Figure 7.6:** Comparison of the loss history of two different implementations of the penalty term**Figure 7.7:** Comparison of the predictions of two different implementations of the penalty term**Figure 7.8:** Loss history of the final design

**Figure 7.9:** Results for the final inverse neural network

One might notice that in figure 7.9b, some nonzero values of  $\Theta_{i,k}$  were predicted. The assumption was made that the neural network would not predict any design sets containing all zeros, as the training set did not contain such a design set. To make the process more rigorous, a function was added into the code that would check if this was the case. If the neural network did predict a design set with all zero entries, the design set would not be returned to the user and an error message would be displayed. In addition this code also checked if there were any predictions above  $90^\circ$ . Through multiple iterations and training sessions on thousands of features it never occurred that this was the case, given a reasonable amount of epochs.

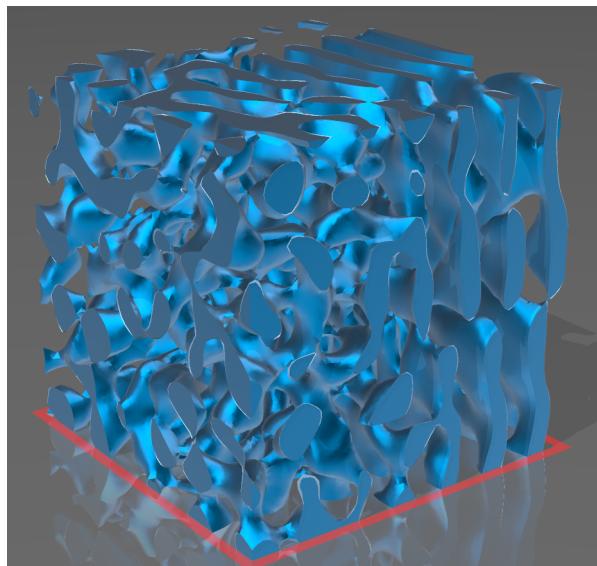
## 7.4. Spatially variant topologies

In this section the results of the one-dimensional linear interpolation method discussed in chapter 6 are presented. In figure 7.10 a spatially varying topology was generated using such a linear interpolation. This topology was created by varying the topology between two randomly generated test samples using the interpolation method from equation 6.2. Not only does this demonstrate the fact that the neural network can predict valid topologies, it also demonstrates a proof of concept for the proposed linear interpolation method. The design set and predicted stiffness matrix of the random samples is given below.

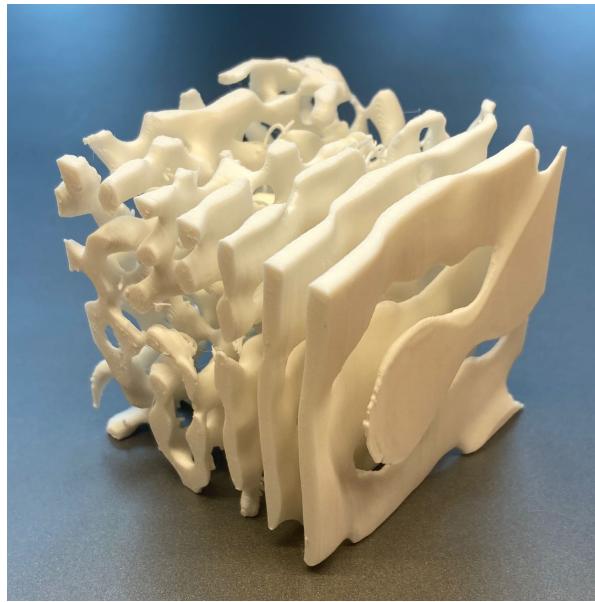
Design parameter	Design 1	Design 2
$\rho$	0.6110	0.4207
$\theta_1$	44.1333	24.9648
$\theta_2$	63.1852	0.0000
$\theta_3$	46.5428	0.0000

**Table 7.1:** Predicted design parameters

Scalar	Design 1	Design 2
$S_{11}$	8.0310	4.1981
$S_{12}$	7.4154	2.3143
$S_{13}$	7.6163	2.3937
$S_{22}$	6.5573	1.5835
$S_{23}$	5.1647	2.0720
$S_{33}$	7.1438	0.3093
$S_{44}$	6.2619	0.0229
$S_{55}$	5.3795	0.0229
$S_{66}$	3.7290	0.9582

**Table 7.2:** Reconstructed stiffness matrices**Figure 7.10:** Spatially Varying Spinodoid Topology

From table 7.1 and figure 7.10 a transition from a cubic/isotropic topology to a lamellar topology can be seen.



**Figure 7.11:** 3D Printed demonstration model

## 7.5. Bone structure

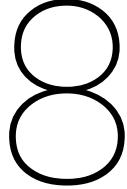
It is clear the constructed stiffness does not correlate as well to the given stiffness as could be expected from the accuracy information provided before. Also, it is evident that such a stiffness matrix simply cannot exist as demonstrated by the very problematic reconstruction. In 'Discussion', this will be explained further.

Scalar	Given stiffness	Constructed stiffness
$S_{11}$	0.390831	0.3026
$S_{12}$	0.058835	0.0948
$S_{13}$	0.061290	0.1173
$S_{22}$	0.160911	0.0933
$S_{23}$	0.031779	0.0110
$S_{33}$	0.155180	0.1311
$S_{44}$	0.039351	0.0214
$S_{55}$	0.081009	0.2090
$S_{66}$	0.073581	0.1559

**Table 7.3:** Reconstructed bone stiffness

Motivation for inverse designing metamaterials to match the anisotropic stiffness of a bone specimen.

Human bone is a material with anisotropic properties. This means that every design vector has a unique stiffness matrix, but every stiffness matrix has multiple corresponding design vectors. With use of an inverse neural network and a forward neural network, a unique design vector of a given stiffness matrix can be found. This is very useful for bone implants, since the body might react better to an implant that has the same design as the original bone. Moreover, inverse designing is less time consuming than doing FEM analysis and it can be more cost efficient.



# Discussion

## 8.1. Data exploration

The experimentation with data distribution selection did yield interesting insights in the value of good data and its affect on the accuracy of our algorithm. However, some questions still remain about our implementation and the results.

First, it is unclear to what extent random error has shaped our results. A large number of distributions is tested to reduce the influence of random behaviour on the results, but to what extent this was successful is still not evident. Constraints in time and computing capabilities prevented further research into this topic.

Second, the selection of dataset distribution might prove to be worthwhile for datasets of lower quality and/or quantity. This could be interesting to explore in other testcases where good datasets are not readily available and there is more room for improvement upon the current methods.

## 8.2. Loss function analysis

When testing the late stage implementation of the penalty term, it would regularly happen that the loss function seemed to become more stable with respect to spikes in loss, see figure 7.6b. This is an interesting result, as the spikes in loss are mainly due to the stochastics involved in the Adam optimizer. Loss spikes are a well known and studied limitation of the Adam optimizer [11]. It would therefore be interesting to see why the implementation of the penalty term mitigated this undesirable effect. Attempts to understand the phenomena to where to no avail. Time constraints played a role in this.

## 8.3. Shifting of design set predictions

In chapter sections 5.2.2 and 7.3, it was discussed how design constraints were imposed up on the method. The choice was made to implement a feature shifting algorithm just before the predictions were returned to the user. However, there are more possibilities when it comes to imposing the design constraints. It could be possible to integrate a feature shifting/filtering algorithm in the training of the neural network. If this would be done, activation functions would have to be used in the shifting/filtering of the features to maintain the gradients for optimization (for example a ReLU activation function for  $\Theta_{i,k}$  shifting). For this project, the deliberate choice was made of implementing the algorithm after training. The reason for this is that the distribution of zero-value containing design parameter sets was more similar to that of the generated data set. The rational is that this would enable for a more accurate representation of all topologies in the predicted design parameter sets.

## 8.4. Three-dimensional interpolation

In chapter 6, a proposition for a three-dimensional interpolation method for GRFs was proposed. Due to time constraints, an implementation of this method remains for a future study. If one would implement such a function, it would be wise to look into optimizing the proposed method in terms of computational expenses. The linear, one-dimensional spatially variant topology in figure 7.10 was created by using

two GRFs. Because the interpolation of two GRFs in a linear one-dimensional fashion was not going to be a computational challenge, the GRF values were computed for the complete cube. When this method would be scaled to an implant-sized, three-dimensional grid containing thousands of nodes, this method would impose serious computational expenses. A possible solution is to create a mesh containing tetrahedrons. When the GRFs are computed, only the space in between the surrounding nodes have to be taken into account. This mesh can then be used for the interpolation method.

## 8.5. Bone structure

Unfortunately the provided stiffness matrix differs from the constructed stiffness vector. This is the result of a code error that was discovered very late, during the final stages of the writing process of this report. It was hypothesised that this was probably due to some normalization error, as the results for the test data (which was similar to the bone specimen stiffness) was very accurate. Sadly, in the process of troubleshooting this, an error in the code rendered the neural network completely useless. This meant the neural network had to be trained again before the bug could be fixed. This process takes hours, there was simply not sufficient time for this to take place before the report had to be handed in. Also, visualisations of the stiffness matrices corresponding to generic cubic, lamellar and columnar topologies will be provided as these sadly could not be included for the same reason.

# 9

## Conclusion

In the first few chapters we talked a lot about the theory of our networks and topologies. In chapter 7 we can compare our results to our assumptions and draw conclusions.

We have four important subsections with results. First the data exploration, our proposed simple distributions performed very poorly compared to the original distribution. One-third of variations outperformed the original distribution but we could not find a discernible pattern in the distributions that scored best on  $R^2$ . Because of this we can not conclude which of the distributions would be ideal for our algorithm.

The forward neural network performed really well with a great value for  $R^2$ . The graph converges to a global minimum without overfitting or underfitting. The graph showed us the best results with the loss history at 200 epochs. This gives the best view of the global minimum without an unnecessary long running time.

We used an increasing  $\lambda$  for the implementation of the loss function between the last 80-90% of the training. At later stages an increasing  $\lambda$  would cause problems with the prediction of values of  $\theta$ . It also provided better results than a constant  $\lambda$  at the first  $p = 10$  of the training. The average  $R^2$  values we got are very high. The predictions for  $\theta_i$  are not accurate at all. Our results support our predictions that there exist multiple sets of design parameters which result in the same stiffness.

For the last section, the spatially variant topologies we chose linear interpolation. Our results demonstrate that the neural network can predict valid topologies. It also demonstrates a proof of concept for the proposed linear interpolation method.

# References

- [1] Isaac Amidror. "Scattered data interpolation methods for electronic imaging systems: a survey". In: *Journal of Electronic Imaging* 11.2 (2002), pp. 157–176. DOI: 10.1117/1.1455013. URL: <https://doi.org/10.1117/1.1455013>.
- [2] John W Cahn. "On spinodal decomposition". In: *Acta metallurgica* 9.9 (1961), pp. 795–801.
- [3] Aileen M Davis et al. "Waiting for hip revision surgery: the impact on patient disability". In: *Can. J. Surg.* 51.2 (Apr. 2008), pp. 92–96.
- [4] Eve Donnelly. "Methods for assessing bone quality: a review". In: *Clinical Orthopaedics and Related Research®* 469 (2011), pp. 2128–2138.
- [5] Marie-Claire Gaudet et al. "The wait for total hip replacement in patients with osteoarthritis". In: *Canadian Journal of Surgery* 50.2 (2007), p. 101.
- [6] Kai Hormann and N Sukumar. *Generalized barycentric coordinates in computer graphics and computational mechanics*. CRC press, 2017.
- [7] Shima Kamyab et al. "Deep learning methods for inverse problems". In: *PeerJ Computer Science* 8 (2022), e951.
- [8] Eva Klintström et al. "Predicting trabecular bone stiffness from clinical cone-beam CT and HR-pQCT data; an in vitro study using finite element analysis". In: *PLoS One* 11.8 (2016), e0161101.
- [9] Siddhant Kumar et al. "Inverse-designed spinodoid metamaterials". In: *npj Computational Materials* 6.1 (2020), p. 73.
- [10] Kumar, Siddhant. *data.csv*. [Online; accessed 12-May-2023]. 2023.
- [11] Igor Molybog et al. "A theory on adam instability in large-scale machine learning". In: *arXiv preprint arXiv:2304.09871* (2023).
- [12] Jon Ander Rivera, David Pardo, and Elisabete Alberdi. "Design of loss functions for solving inverse problems using deep learning". In: *Computational Science—ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part III 20*. Springer. 2020, pp. 158–171.
- [13] Robert H. Shmerling. "How long will my hip or knee replacement last?" In: (2021). [Online; accessed 22-June-2023]. URL: <https://www.health.harvard.edu/blog/how-long-will-my-hip-or-knee-replacement-last-2018071914272#:~:text=For%20people%20considering%20hip%20or%20knee%20replacements%20each%20year>.
- [14] "The design and in vivo testing of a locally stiffness-matched porous scaffold". In: *Applied Materials Today* 15 (2019), pp. 377–388. ISSN: 2352-9407. DOI: <https://doi.org/10.1016/j.apmt.2019.02.017>. URL: <https://www.sciencedirect.com/science/article/pii/S2352940718307182>.
- [15] Vladimir V. Popov Jr. et al. "Design and 3D-printing of titanium bone implants: brief review of approach and clinical cases". In: *Biomedical Engineering Letters* (2018). URL: <https://doi.org/10.1007/s13534-018-0080-5>.
- [16] Daniel A White et al. "Multiscale topology optimization using neural network surrogate models". In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 1118–1135.
- [17] Li Zheng, Siddhant Kumar, and Dennis M Kochmann. "Data-driven topology optimization of spinodoid metamaterials with seamlessly tunable anisotropy". In: *Computer Methods in Applied Mechanics and Engineering* 383 (2021), p. 113894.