



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Empa

Materials Science and Technology

A Surrogate Modeling Approach for Cost-Efficient Mechanical Analysis of Lattice Structures

Christian Hauschel (Semester Project)
Fabian Jüngling (Bachelor Thesis)

Spring Semester 2020

Prof. Dr. Edoardo Mazza

Dr. Ehsan Hosseini
Amin Kazemi

Preface

This thesis was written in Spring Semester 2020, during which the physical access to ETH Zürich and the execution of laboratory experiments were restricted due to the COVID-19 pandemic. Therefore, the initial work plan was adjusted in agreement with the supervisor in order to adapt to this situation.

Abstract

Lattice structures are used in a variety of applications, for example as vibration absorbers in aerospace or in the medical field as part of implants. Analyzing such structures with finite elements comes with great computational cost as calculations with a huge number of degrees of freedom are necessary. Different homogenization approaches for a variety of lattice types have been developed in the past. However, most of these techniques are based on elastic models developed for specific load cases. In this project, the body-centered cubic lattice geometry is used. For this lattice, initially orthotropic material models are developed for uniaxial and shear loading conditions. The limitations of this approach are identified and discussed in this study. Furthermore, a novel modeling technique using a surrogate model trained for general load cases is presented. This model describes the mechanical behavior of the body-centered cubic lattice cell associated to a representative cube element. Preliminary results show that such a model can adequately describe the stress-strain relationship of structures made of lattice units. In the future, this model could be implemented into a finite element package as a custom material or element.

Contents

1	Introduction	1
1.1	Literature Review	1
1.2	Objective of This Project	3
2	Methods	5
2.1	Geometry, Mesh and Materials	5
2.2	Simple Boundary Condition Modeling	6
2.3	Random Boundary Condition Modeling	8
2.3.1	Lattice Cell	8
2.3.2	Cube Element 1	9
2.3.3	Cube Element 2	10
2.3.4	Surrogate Model	10
2.3.5	Reduced-Integration Modeling	11
2.3.6	Mathematical Tools for Surrogate Modeling	11
2.3.7	Code Verification	14
3	Results	17
3.1	Training and Validation Errors of the Random Boundary Condition Models	17
3.1.1	Full-Integration Model	17
3.1.2	Reduced-Integration Model	18
3.2	Model Implementation and Verification	20
3.2.1	Implementation into Abaqus	20
3.2.2	Results	21
4	Discussion	25
4.1	Simple Boundary Condition Modeling	25
4.2	Random Boundary Condition Modeling	26
4.2.1	Implementation of the Full-Integration Model into Abaqus	26

Contents

5 Conclusion	27
5.1 Outlook	27
A Appendix	29
A.1 Authors	29
A.2 Code Manual	29
Bibliography	34



Introduction

The concept of cellular structures can be seen in a variety of natural materials such as wood, bones or sponges. The high strength-to-weight ratio makes such structures ideal for various applications, for example as core in lightweight sandwich panels as Wadley [16] describes it in his study. In this project however, the focus is specifically on lattice structures. These structures are composed of periodic arrangements of single cells, which are formed by different configurations of struts.

Lattices have many potential applications in various fields of technology. For example, they are used in additively manufactured biomedical implants. In the work of Alabort et al. [1] it is shown that micro-scale lattice structures can be designed to match the mechanical properties of bone tissue. With this approach the issue of stress shielding can be prevented. Another field where these types of structures have a big potential is aerospace. For example, in the paper of Matlack et al. [11] a lattice able to absorb low-frequency vibrations is presented.

Due to the complexity of lattice structures a very fine mesh is needed to represent the geometry in a finite element analysis (FEA). For instance, for a strut diameter of 200 μm the required mesh size to get accurate results is around 20–50 μm . For bigger lattice structures, composed of hundreds of cells, a huge number of degrees of freedom is necessary. Thus, calculations on such large lattices are computationally very expensive. Consequently, fully resolved FEA of such structures are not feasible for the designer. This has led to various macroscopic descriptions of lattices.

1.1 Literature Review

In this project, the analysis is focused on a specific cell geometry: The body-centered cubic (BCC) lattice cell (see Figure 1.1). It is used here because of its geometric simplicity and also because it seems to be predominantly discussed in the literature. For example, Xu et al. [17] analyzed the Young's modulus of the

1 Introduction

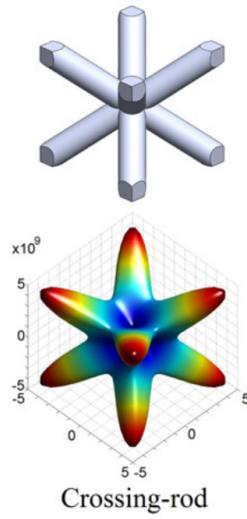


Figure 1.1: Spatial distribution of Young's modulus of the BCC cell (Xu et al. [17]).

BCC lattice cell. They found that the modulus is dependent on the direction in space. A visualization of the modulus distributed in space is shown in Figure 1.1. There it can be seen that the stiffness of the structure is symmetric. It can also be observed that the highest stiffness of the structure is in the diagonal direction parallel to the struts. For the BCC geometry, different homogenization methods are presented by various researchers. The aim of such a homogenization is to replace the lattice cell with a solid cube element that shows the same behavior as the lattice. Using this concept, the complexity of analysis can be reduced to lower-degree calculations since the lattice cell is no longer treated as a structure but rather as a material.

In the literature, many authors try to generate homogenization models with material laws similar to isotropic elastic materials. Analytical formulations are developed to correlate geometric parameters of the structure to its material moduli. For instance, Ptochos and Labeas [13] developed a model in which the Young's modulus E and the Poisson's ratio ν are made dependent on the width of the cell l and the radius of the struts r . These correlations are defined in the following way:

$$\frac{E^*}{E_s} = \frac{4\sqrt{3}}{\frac{l^2}{\pi r^2} + \frac{l^4}{2\pi r^4}} \quad (1.1)$$

and

$$\nu = \frac{-\frac{1}{\pi r^2} + \frac{l^2}{4\pi r^4}}{\frac{1}{\pi r^2} + \frac{l^2}{2\pi r^4}}. \quad (1.2)$$

Another approach is shown in the work of Ushijima et al. [15]. They analytically developed an equation for the Young's modulus dependent on the lattice dimensions under certain assumptions and boundary conditions. They validated their model using 1D beam and 3D solid elements in a FEA. Furthermore, they compared experimental data from compression tests to their calculations. For a certain ratio of strut radius to the width of the cell they achieved accurate results with their model.

Luxner et al. [6] were facing the issue from another direction by modeling the structure with Timoshenko beam elements. The quadratic beam elements allow for bending and shear deformation. Not taken into consideration were the overlapping volume at the nodes. To overcome this, the nodes were modeled with a sphere with a radius equal to the diameter of the struts.

Then, there is the correlation according to Gibson and Ashby [4]. This equation describes the dependency of the Young's modulus of a general lattice cell geometry to its density ratio. In the review article of Maconachie et al. [7] the Ashby-Gibson model is presented in more detail. There, it is formulated as follows:

$$\frac{E^*}{E_S} = C_1 \left(\frac{\rho^*}{\rho_S} \right)^n, \quad (1.3)$$

where E^* and ρ^* are the Young's modulus and the density of the lattice cell. E_S and ρ_S are the analogous quantities for the core material, respectively. C_1 is an experimentally derived constant. The exponent n is dependent on whether the lattice is stretch-dominated or bending-dominated. One can determine this behavior with the Maxwell equation. It can be formulated as follows for 3D structures:

$$M = s - 3n + 6, \quad (1.4)$$

where s is the amount of struts and n the amount of nodes. In the case of bending-dominated structures $M < 0$ while for stretch-dominated structures $M \geq 0$. For the BCC, where $s = 8$ and $n = 9$, Equation (1.4) yields $M = -13$. Therefore, the BCC behaves in bending-dominated manner. In such a case the exponent n equals 2 and Equation (1.3) becomes:

$$\frac{E^*}{E_S} = C_1 \left(\frac{\rho^*}{\rho_S} \right)^2. \quad (1.5)$$

The experimental data summarized in the study of Maconachie et al. [7] seems to approximately corroborate this correlation with this exponent.

There are also multiple numerical approaches in the literature to get to the elastic moduli. For example, Alwattar and Mian [2] used simple load cases to parameterize an orthotropic material model for a BCC lattice. They also used a neural network to find a model explaining the elastic moduli depending on the geometrical dimensions. Another example of using a numerical approach is the work of Omairey et al. [12]. For this study the authors even developed an Abaqus toolbox to homogenize general periodic cellular structures.

1.2 Objective of This Project

The aim of this project is to develop a novel modeling solution for lattice structures, here specifically the BCC geometry. Based on the mechanical behavior of one single lattice unit cell, the behavior of a bigger lattice is to be predicted. The first approach is the development of an orthotropic material model using simple boundary conditions, as it can be seen similarly in the literature. Then, the goal is to develop a material model using a surrogate model trained on general, randomly sampled boundary condition data. An implementation into Abaqus is discussed to validate the results of this approach for bigger clusters of lattice structures.

2

Methods

In this chapter, firstly, the lattice geometry with its mesh parameters and its material is presented (Section 2.1). Secondly, a simple approach to generate a model using a selected set of boundary conditions is discussed (Section 2.2). Lastly, the methods used to generate a surrogate model using randomly sampled boundary conditions to obtain a description of the lattice cell's mechanics for general load cases is presented (Section 2.3).

2.1 Geometry, Mesh and Materials

In this project, a lattice cell geometry as depicted in Figure 2.1 is used. It is a BCC lattice cell with a width of 1 mm and a strut diameter of 0.15 mm. More parameters and measures about the geometry are depicted in Table 2.1.

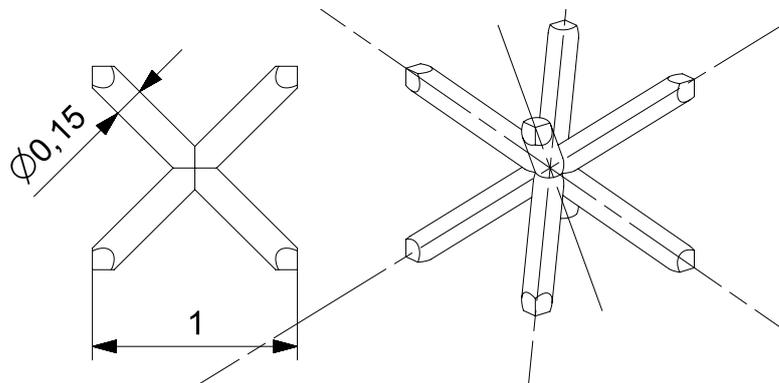


Figure 2.1: The BCC lattice cell geometry used for the simulations.

The simulations explained in the next sections are carried out using Abaqus. For this, the lattice geometry described above is meshed with quadratic tetrahedron elements with a seed size of 0.05 mm yielding 5 896 elements (see Figure 2.2).

2 Methods

Parameter	Variable	Value	Unit
Cell width	W	1	mm
Strut diameter	D	0.15	mm
Lattice volume	v	0.105 9	mm ³
Young's modulus bulk material	E_S	200 000	MPa
Poisson's ratio bulk material	ν_S	0.3	–
Density bulk material	ρ_S	0.007 85	g mm ⁻³
Cell volume	V	1	mm ³
Cell mass	m	0.000 831	g
Density lattice	ρ^*	0.000 831	g mm ⁻³

Table 2.1: The parameters of the BCC lattice used in this project.

All the lattice cell simulations carried out in this project use the same meshing parameters. Additionally, a steel material is assigned with a Young's modulus of 200 000 MPa and a Poisson's ratio of 0.3.

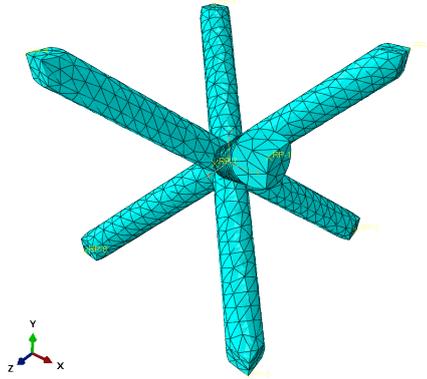


Figure 2.2: The lattice cell meshed with 5 896 quadratic tetrahedron elements.

2.2 Simple Boundary Condition Modeling

An approach to homogenize the lattice cell is the use of a selected set of non-random boundary conditions. In this project, the boundary conditions are applied according to the way Alwattar and Mian [2] describe it in their paper. The first boundary condition setup is the normal displacement load case (Figure 2.3a), in which four surfaces of one side are displaced in z-direction. On the opposite side of the cell, the surfaces are fixed in z-direction. Lateral contraction is allowed for the surfaces. The second boundary condition setup (Figure 2.3b) is the shear displacement load case. In this configuration, four surfaces on one side are displaced in x-direction, the surfaces on the opposite side are fixed in all three directions.

2.2 Simple Boundary Condition Modeling

The resulting deformation of the lattice for these two load cases is visualized in Figure 2.4.

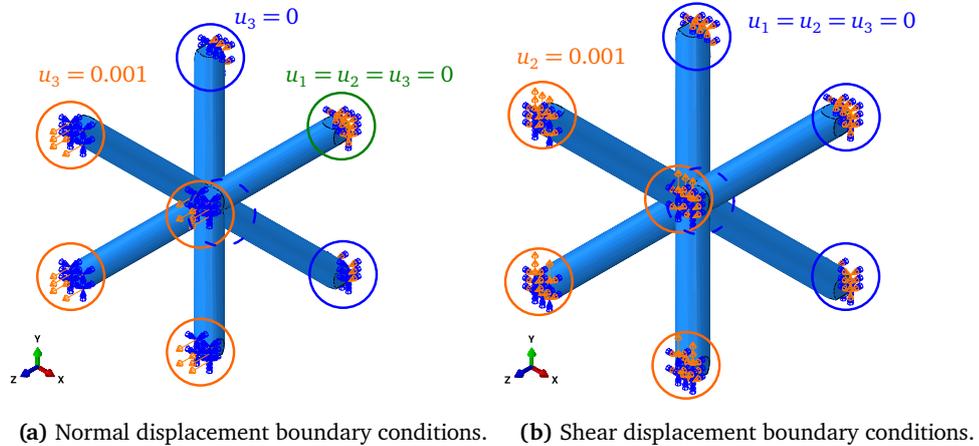


Figure 2.3: The two different boundary condition configurations.

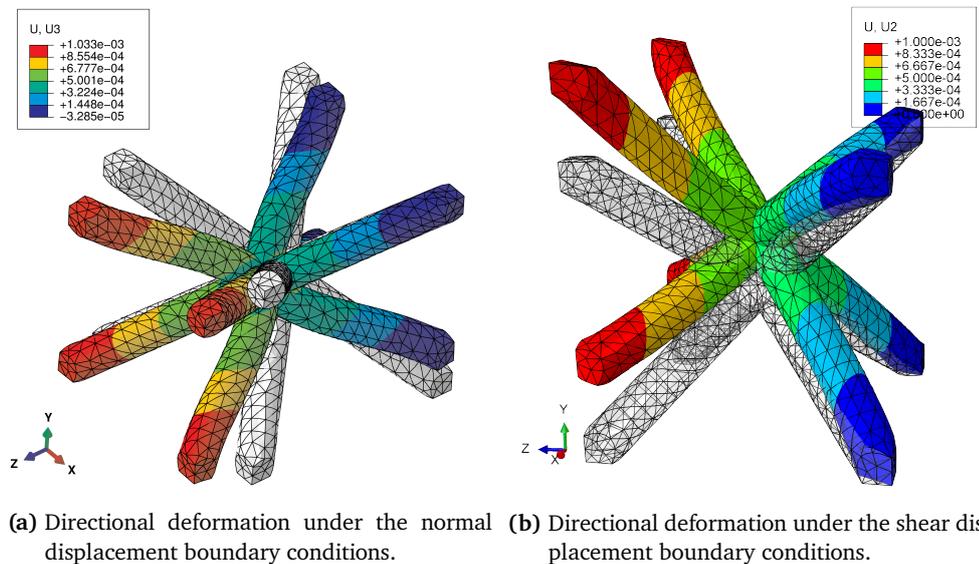


Figure 2.4: The deformation for the two different boundary condition configurations.

Using these boundary conditions, the parameters for a material model are calculated. The material is defined to be orthotropic but because of the symmetry of the BCC cell, the material is here called “quasi-isotropic”. By this term it is meant that it behaves like an elastic isotropic material. However, the shear modulus is defined to be independent of the Young’s modulus and the Poisson’s ratio. The model therefore has three independent parameters instead of only two. In mathematical

2 Methods

terms, the material law can be expressed as follows:

$$\begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{23} \\ \varepsilon_{13} \\ \varepsilon_{12} \end{pmatrix} = \begin{pmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2G} \end{pmatrix} \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix}. \quad (2.1)$$

The Young's modulus E is defined as

$$E = \frac{\sigma_{33}}{\varepsilon_{33}}, \quad (2.2)$$

the Poisson's ratio ν as

$$\nu = -\frac{\varepsilon_{11}}{\varepsilon_{33}}, \quad (2.3)$$

and the shear modulus G as

$$G = \frac{\sigma_{12}}{2\varepsilon_{12}}. \quad (2.4)$$

By measuring the corresponding reaction forces and lateral displacements in Abaqus, the three parameters defined in the Equations (2.2–2.4) can be calculated. It is to note that due to the symmetry of the material definitions and the BCC cell itself, other directions of stress and strain might also be used to define the parameters.

Furthermore, for reasons of comparison, an additional model is developed using Ansys' Material Designer (see also Ansys [3]). This is a toolbox which allows to homogenize custom materials in Ansys. For the BCC cell it generates a quasi-isotropic material as well.

2.3 Random Boundary Condition Modeling

In this section, the idea to generate a surrogate model able to describe the mechanical behavior of the BCC lattice cell under general deformations is discussed. The modeling procedure is illustrated in Figure 2.5. The approach consists of three simulations all carried out in Abaqus:

- lattice cell,
- cube element 1 and
- cube element 2.

2.3.1 Lattice Cell

The first simulation is the lattice cell calculation. At the corner nodes of the lattice cell, eight reference points are defined. The three surfaces connected with each

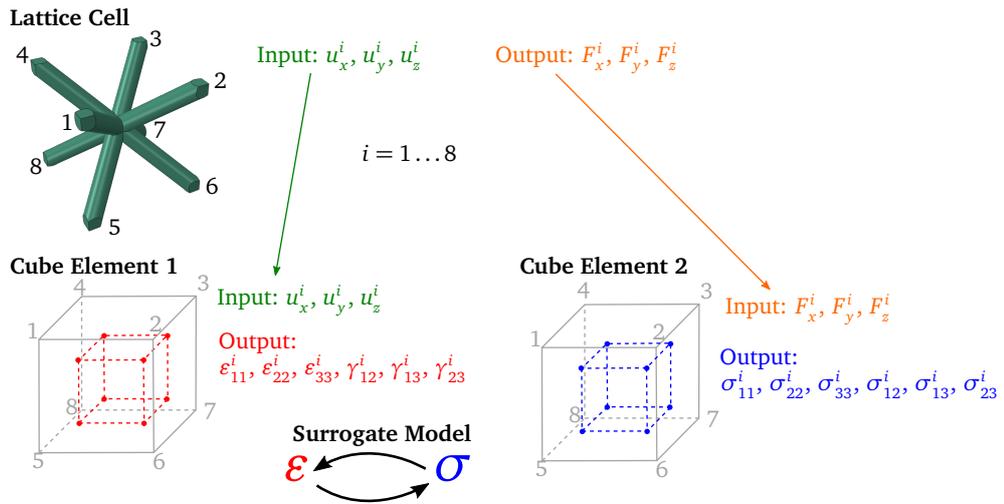


Figure 2.5: Modeling approach with the full-integration representative cube elements.

corner node are coupled with these reference points. At the reference points, displacement boundary conditions are applied with random sampling of the displacement in all three directions of space. This is depicted in Figure 2.6. Additionally, the rotational displacements of the reference points are set to be free. After the calculation in Abaqus is done, the reaction forces at the same reference points are read out. These forces are denoted as outputs of the simulation.

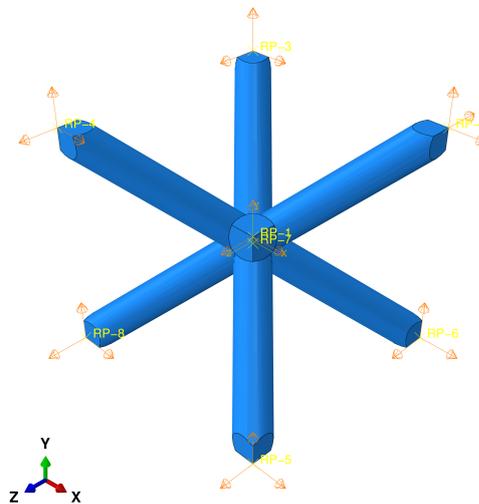


Figure 2.6: The displacement boundary conditions in all three directions assigned to the eight reference points at the corners of the cell. The rotational displacement is set to be free.

2.3.2 Cube Element 1

The next simulation carried out in Abaqus is the representative cube element 1 (see Figure 2.5). The geometry of this simulation is a cube with a width of 1 mm.

It is meshed with one linear full-integration element. The material assigned to the cube is the same as for the lattice simulation. For the calculation, the displacement values generated in the lattice cell simulation are used as inputs and applied to the corner nodes of the cube. After the simulation, the strain values at the element's integration points are read out. In this case, six strain components exist for each integration point yielding a total of a 48-dimensional output. An example of a deformed lattice and a deformed cube element where the same displacement boundary conditions are applied can be seen in the Figures 2.7a and 2.7b.

2.3.3 Cube Element 2

The last simulation to be discussed in Figure 2.5 is the cube element 2. The geometry, mesh and material of the simulation is the same as in the cube element 1. The boundary conditions are now different however. Here, the reaction forces obtained in the lattice cell simulation are applied to the corner nodes as concentrated forces. As outputs of the calculation, the stress values at the element's integration points are read out. A lattice cell's reaction forces and its application to a cube element as concentrated forces at the corners is shown in Figures 2.7c and 2.7d.

2.3.4 Surrogate Model

The three simulations described before are executed for a set of random displacement boundary conditions applied in the lattice cell.¹ A MATLAB code produces Abaqus input files with the sampling data for the three different simulations for a defined number of sampling points. These input files are then executed using the Abaqus command line tools. Various Python scripts read out the output data from the Abaqus result files. The outputs of these simulations are the strain and stress values at the cube element's integration points. The idea is to model the relationship between the strain and stress data using data-driven “black box” modeling. This modeling is carried out in MATLAB with the help of the uncertainty calculation and metamodeling framework UQLab (see Marelli and Sudret [9]).

Eventually, the model should define the constitutive behavior of the lattice cell for the representative cube element. By applying the boundary conditions randomly, the idea is to have a reduced model able to describe a multitude of different complex load cases. For big lattice structures, the single lattice cells with their detailed geometries can then be replaced by simple cube elements with a constitutive behavior described by the surrogate model. Instead of running a full finite element calculation, only an evaluation of the surrogate model is necessary for each representative cube. Theoretically, it should be possible to implement the model as a custom material law or a custom element in Abaqus. However, as it is discussed later in Section 4.2.1, there seem to exist difficulties to implement such a model.

¹ The generation of such sets of random points in the “displacement hyperspace” (a space of 48 dimensions for the full-integration element) is shown in Section 2.3.6.

2.3 Random Boundary Condition Modeling

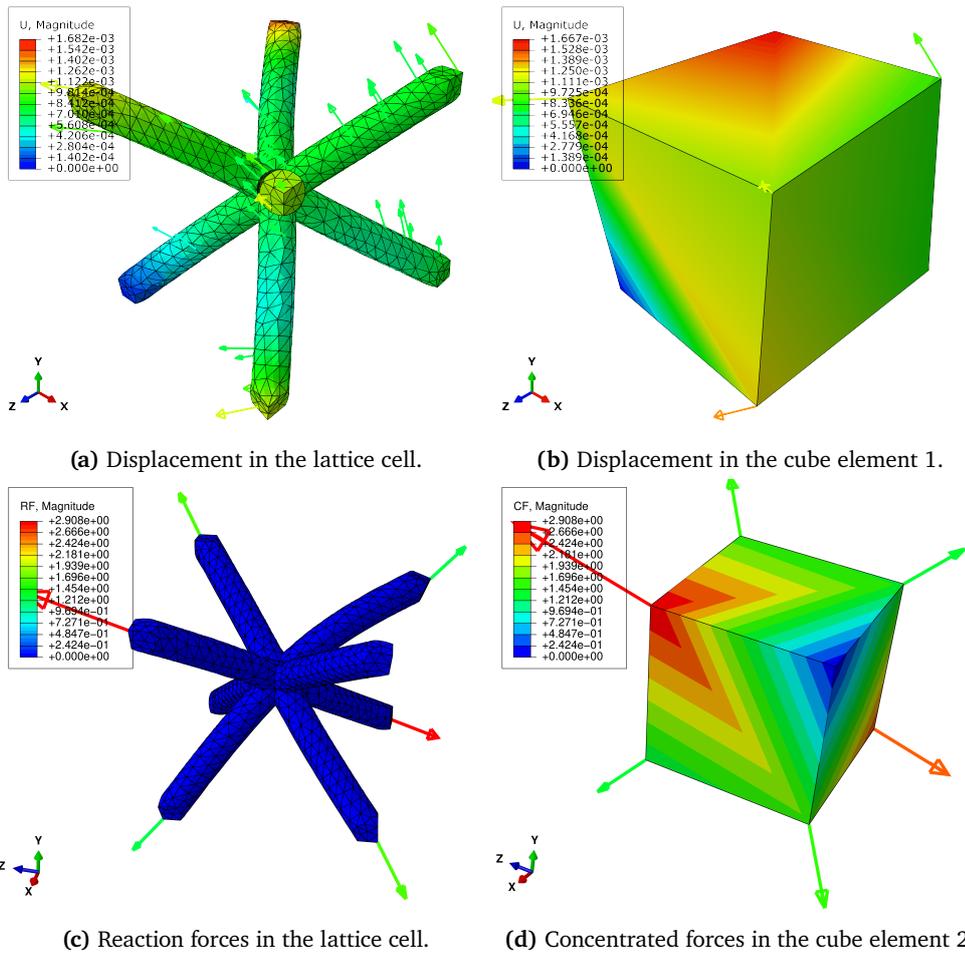


Figure 2.7: Depicted here are all the three simulation types for the random boundary condition modeling approach. It can be seen how the displacement from the lattice cell (2.7a) is applied to the cube element 1 (2.7b). Additionally, it is visible how the reaction forces of the lattice cell (2.7c) are transferred to the cube element 2 as concentrated forces (2.7d).

2.3.5 Reduced-Integration Modeling

A variant of the above described modeling approach is the use of reduced-integration elements for the cube 1 and cube 2 simulations instead of the fully-integrated elements. This modified procedure is illustrated in Figure 2.8. This new approach yields only six strain and stress values as in- and outputs for the surrogate model. The number of dimensions is therefore greatly reduced. As this modeling formulation has only one integration point per element, a possible implementation into a custom material in Abaqus (with FORTRAN and UMAT) is made more convenient than with the eight integration points of the full-integration elements.

2.3.6 Mathematical Tools for Surrogate Modeling

In the following, the different mathematical tools for sampling, fitting and error calculation of surrogate models used in this study are presented. Additionally, their implementation in MATLAB is shortly described.

2 Methods

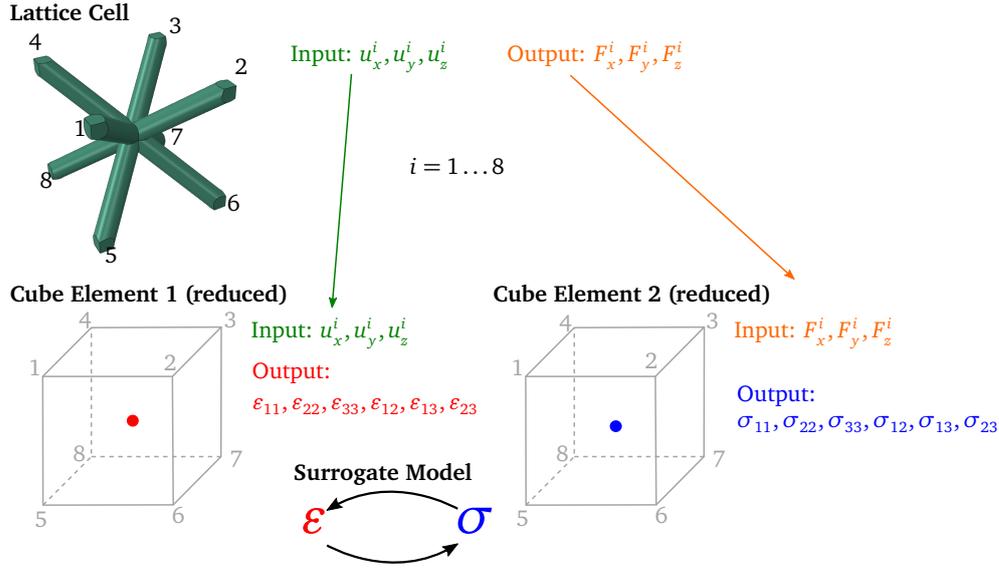


Figure 2.8: Modeling approach with the reduced-integration representative cube elements.

Multivariate Linear Regression (MLR)

The MLR model is implemented in MATLAB using the `mvregress` function. According to the MATLAB documentation (MathWorks [10]), this model is defined as follows:

$$y_i = X_i \beta + e_i, \quad i = 1, \dots, n, \quad (2.5)$$

where y_i is a d -dimensional vector of responses, X_i is a design matrix of predictor variables, β is a matrix of regression coefficients and e_i is a d -dimensional vector of error terms with multivariate normal distribution. In this study, X_i is directly filled with the strain values, it is therefore a vector. y_i consists of the stress values at all the integration points. n describes the number of sampled points.

Elastic Material Laws

The following types of elastic material models are tested as simple surrogate models:

- isotropic (with two degrees of freedoms),
- orthotropic (with nine degrees of freedoms) and
- anisotropic (with 36 degrees of freedoms).

In this study, the parameters for these three models are found using MATLAB's function `fminunc`. `fminunc` is a nonlinear programming solver for unconstrained multivariable functions. Here the objective function to be minimized is defined as sum of the squared residuals.

Kriging

Kriging is a metamodeling method originally developed for the interpolation of spatially-related data in geostatistics. The algorithm assumes a model output to

be a realization of a Gaussian process. A Kriging metamodel provides the prediction $\mathcal{M}^K(x)$ of a new point x , given the observed input and the corresponding response data: $\mathcal{X} = \{x^{(1)}, \dots, x^{(N)}\}$ and $\mathcal{Y} = \{\mathcal{M}(x^{(1)}), \dots, \mathcal{M}(x^{(N)})\}$. For noise-free responses, the Kriging algorithm performs an interpolation. The UQLab framework allows also for regression of noisy data. A detailed documentation of Kriging is found in the manual of UQLab's Kriging module (Lataniotis et al. [5]).

Polynomial Chaos Expansion

Polynomial chaos expansion (PCE) is a surrogate modeling technique to functionally approximate a computational model through its spectral representation using basis polynomials. These polynomials are chosen in accordance to the input distribution of the computational model. In this project the input distributions are statistically inferred from the data with the help of UQLab's inference module (see Torre et al. [14]). More details about PCE can be found in the PCE module manual (Marelli and Sudret [8]).

Random Sampling

Different methods can be used to generate near-random samples of parameter values from multidimensional distributions. In this work, mainly the Latin Hypercube Sampling (LHS) and also the Halton sampling methods are used. For the implementation in MATLAB, the functions `lhsdesign` and `haltonset` are used, respectively. Examples of the two sampling methods are shown in Figure 2.9.

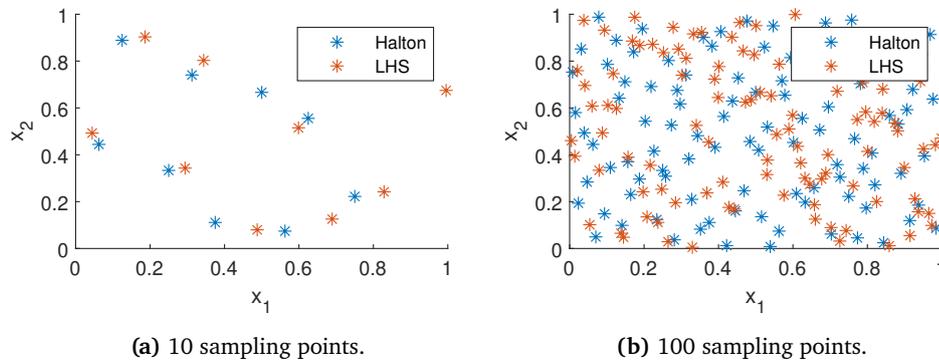


Figure 2.9: Comparison of the LHS and Halton sampling methods using a different number of points.

Error Measures

The following *a posteriori* error estimates are used in order to predict the generalization ability of the trained model.

Training Error. For the Kriging and PCE modeling approaches, the leave-one-out (LOO) cross-validation (CV) error is calculated. Lataniotis et al. [5] formulate the LOO CV error in case of Kriging as in Equation 2.6. A similar definition can be

2 Methods

derived for PCE.

$$\epsilon_{\text{LOO}} = \frac{1}{N} \left[\frac{\sum_{i=1}^N (\mathcal{M}(\mathbf{x}_i) - \mu_{\hat{Y},(-i)}(\mathbf{x}_i))^2}{\text{Var}[\mathcal{Y}]} \right], \quad (2.6)$$

where $\mu_{\hat{Y},(-i)}(\mathbf{x}_i)$ denotes the Kriging model gained from using all the points of the initial experimental design \mathcal{X} , except \mathbf{x}_i . $\mathcal{Y} = \mathcal{M}(\mathcal{X})$ is the computational model's response to the initial input \mathcal{X} .

Validation Error. According to Lataniotis et al. [5] the validation error is calculated as a relative generalization error on an independent data set

$$\mathcal{D}_{\text{val}} = \{(\mathbf{x}_{\text{val}}^{(i)}, y_{\text{val}}^{(i)}), i = 1, \dots, N_{\text{val}}\} \quad (2.7)$$

as follows:

$$\epsilon_{\text{val}} = \frac{N_{\text{val}} - 1}{N_{\text{val}}} \left[\frac{\sum_{i=1}^{N_{\text{val}}} (\mathcal{M}(\mathbf{x}_{\text{val}}^{(i)}) - \mathcal{M}^{\text{Meta}}(\mathbf{x}_{\text{val}}^{(i)}))^2}{\sum_{i=1}^{N_{\text{val}}} (\mathcal{M}(\mathbf{x}_{\text{val}}^{(i)}) - \hat{\mu}_{Y_{\text{val}}})^2} \right], \quad (2.8)$$

where $\mathcal{M}^{\text{Meta}}(x)$ denotes the metamodel's and $\mathcal{M}(x)$ the computational model's response to x . $\hat{\mu}_{Y_{\text{val}}} = \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \mathcal{M}(\mathbf{x}_{\text{val}}^{(i)})$ is the sample mean of the validation set responses. With this error measure, the performance of different surrogate models tested on the same validation data set can be compared.

2.3.7 Code Verification

In order to verify the whole code framework described in Section 2.3.4 using Abaqus for simulation, its Python API to read out the data and MATLAB to generate, process and model the data, a simple test is carried out. In this test, the lattice cell geometry is replaced by a simple cube with an elastic orthotropic material assigned (see Table 2.2). The geometry is meshed with one single linear full-integration element. As boundary conditions the random sampling scheme is used as described before. A number of 25 sampling points are generated as training data and 10 as validation data. The type of surrogate model used to fit to the training data is the orthotropic model described in Section 2.3.6.

D1111	D1122	D2222	D1133	D2233	D3333	D1212	D1313	D2323
200 000	150 000	180 000	220 000	190 000	170 000	160 000	250 000	140 000

Table 2.2: Parameters for the elastic orthotropic material according to the nomenclature used in Abaqus.

Results

As it can be seen in Figure 2.10, the validation errors for the orthotropic model on the 10 validation points (calculated as defined in Section 2.3.6) are in the order of 10^{-12} . Thus, the model fits the validation data perfectly. The code integrity is therefore ensured.

2.3 Random Boundary Condition Modeling

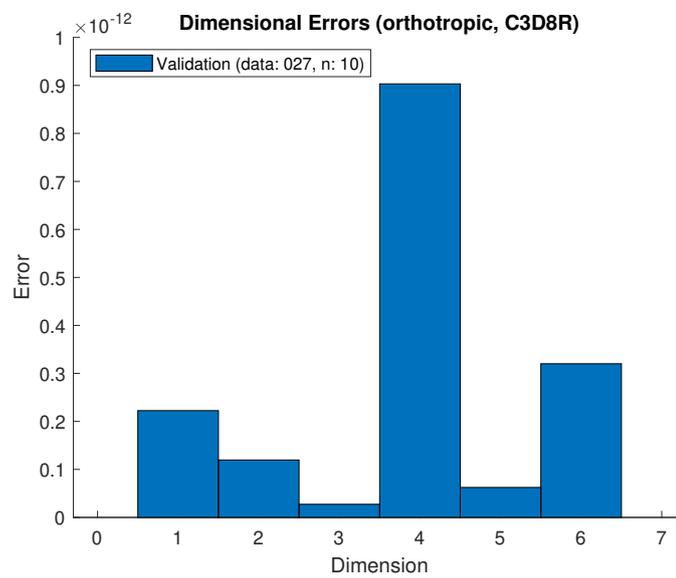


Figure 2.10: Validation error for all six output dimensions (all six stress values) of the orthotropic model used for the code verification.

3

Results

The following chapter shows the results of the developed models. Firstly, the data fitting results are shown for the surrogate models trained on the randomly sampled data. In this context, the training and validation error values from the full-integration model with eight integration points and the reduced-integration model are presented. Secondly, the implementations into Abaqus of the models developed on simple boundary conditions and the reduced-integration model for random boundary conditions are explained. The results of these implementations are presented and the different model performances are compared with each other.

3.1 Training and Validation Errors of the Random Boundary Condition Models

The performances of the surrogate models in predicting data are presented here. Two types of errors are shown respectively: The training LOO CV error according to Equation (2.6) and the validation error according to Equation (2.8). The training error is calculated for Kriging and PCE on a modified training data set according to the LOO CV procedure. The validation error is calculated on a new data set. This error measures the generalization ability of the model. The error values are calculated on the outputs of the models, i.e. the stress values. Therefore, the multidimensional error values are depicted with column charts.

3.1.1 Full-Integration Model

The full-integration model results are depicted for two model types here. Figure 3.1 shows the training and the validation errors of a Kriging model. Figure 3.2 presents the validation results of a MLR model.¹ The modeling errors are

¹ For the MLR model, no training error is calculated. This is due to its implementation in the code.

3 Results

48-dimensional as the modeling has eight integration points each with six stress values. Both the training and validation error values of these two models are in the magnitude of 10^{-12} . The two model structures are thus able to describe the strain-stress relationship very accurately. The differences in the errors between the Kriging and the MLR Method are so small that the more complex Kriging approach with many more parameters seems to be too cumbersome in comparison.

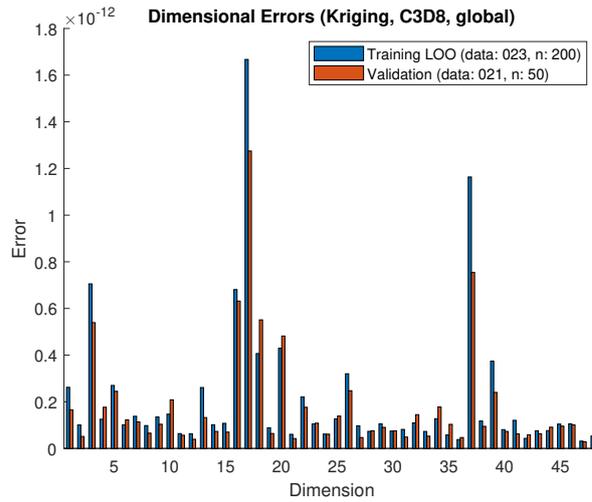


Figure 3.1: Training LOO CV and validation error for the Kriging model.

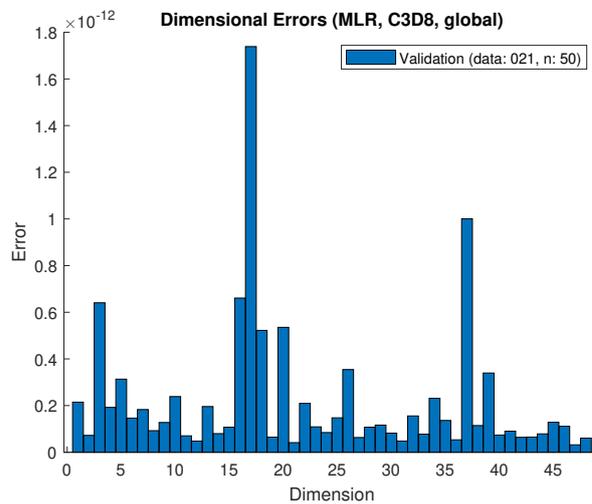


Figure 3.2: Validation error for the MLR model.

3.1.2 Reduced-Integration Model

The reduced-integration model results are displayed for three model types here. Figure 3.3 shows the Kriging training and validation errors, Figure 3.4 presents the errors of a PCE model and Figure 3.5 depicts the errors of an orthotropic

3.1 Training and Validation Errors of the Random Boundary Condition Models

model². As the reduced-integration element has only one integration point, the errors displayed here have only six dimensions. It can be seen that the errors of the three modeling types all follow a similar pattern: The first three dimensional errors are relatively small whereas the last three error components are relatively big (in the range of 15–60%). The first three components denote the normal stresses and the last three the shear stresses. These big errors in the shear components cannot be diminished even if big data sets are used for the training (here, a training data set of 200 points is used). It can therefore be concluded that the modeling procedures with the reduced-integration elements are not capable to capture the underlying mechanics of the lattice cell in a satisfactory manner.

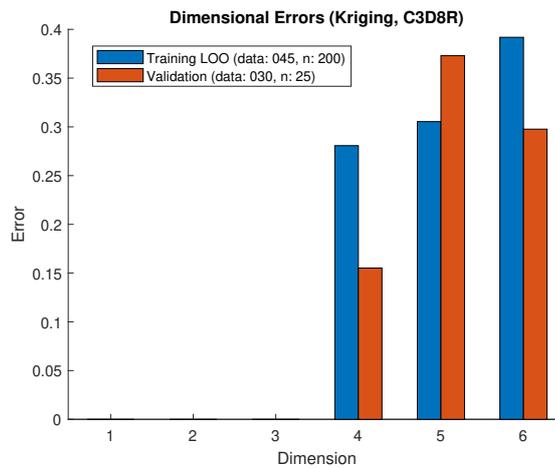


Figure 3.3: Training LOO CV and validation error for the Kriging model.

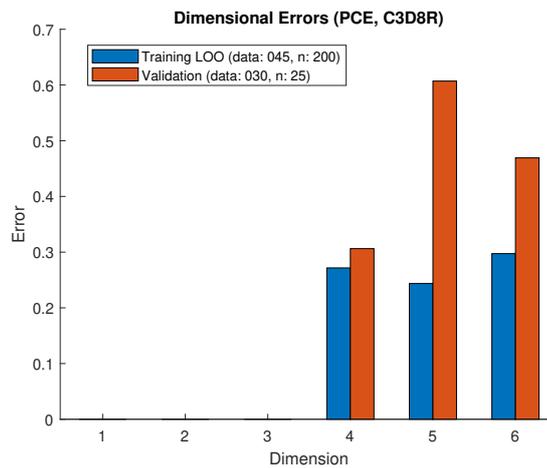


Figure 3.4: Training LOO CV and validation error for the PCE model.

² An orthotropic model formulation is used here instead of the MLR formulation because it is observed that using the MLR formulation usually yields approximately an orthotropic material matrix.

3 Results

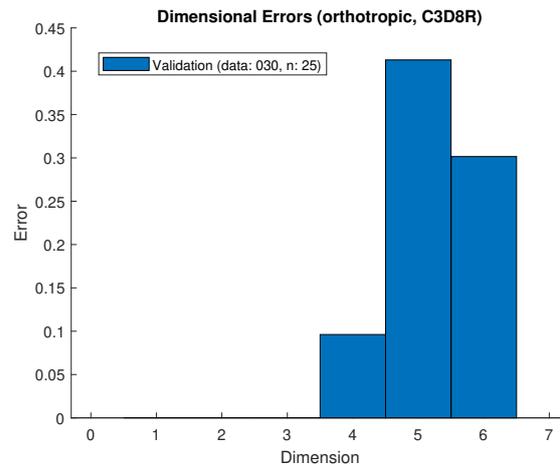


Figure 3.5: Validation error for the orthotropic model.

3.2 Model Implementation and Verification

In this section the implementation of different models into Abaqus is explained. Furthermore, the results of the implemented models tested on lattice clusters subjected to certain boundary conditions are presented and compared.

3.2.1 Implementation into Abaqus

In order to ensure that the models developed under certain boundary conditions describe the “real”³ mechanics of the larger cluster of lattices accurately, the models have to be implemented into Abaqus. The models implemented are the reduced-integration, orthotropic surrogate model (termed “random”, see Figure 3.5), the model trained according to the boundary conditions in Section 2.2 (“non-random”) and the Ansys Material Designer model (“Ansys”) also mentioned in Section 2.2. The employment into Abaqus of these three models is straightforwardly done using the predefined Abaqus orthotropic material option. Unfortunately, the full-integration model cannot yet be implemented as it is further discussed in Section 4.2.1.

The three different models are tested against finite element simulations of the full lattice structures. For that, the material models are applied to clusters of cubic reduced-integration elements. The following selection of clusters is used: 2x2x2, 4x4x4 and 7x7x7. For comparison, finely meshed⁴ finite element simulations of the lattice structures clustered in the same manner are carried out.

The setup for the boundary conditions of the clusters can be seen in Figure 3.6. On two opposite sides of the lattice and cube clusters all the free surfaces are coupled to a reference point (RP) by a kinematic (rigid-body) coupling. The displacement of RP_1 is fixed in all three spacial and rotational directions. On the opposite side, RP_2 is displaced in a chosen direction, to either generate normal or

³ Modeled here with a highly resolved FEA.

⁴ The meshing parameters are analogous to the ones described in Section 2.1 for the single lattice cell.

3.2 Model Implementation and Verification

shear deformation of the structure. The reaction forces at the RP are then extracted from Abaqus and used for comparison. For a precisely working homogenization of the lattice the reaction forces should be the same for both models. The error values are calculated according to the following equation:

$$\text{Absolute Error} = \left| \frac{\text{RF}_{\text{Lattice}} - \text{RF}_{\text{Cube}}}{\text{RF}_{\text{Lattice}}} \right|, \quad (3.1)$$

where RF denotes the reaction forces. In Figures 3.7 and 3.8 the displacement is visualized for a 7x7x7 cluster in the case of normal and shear displacement, respectively.

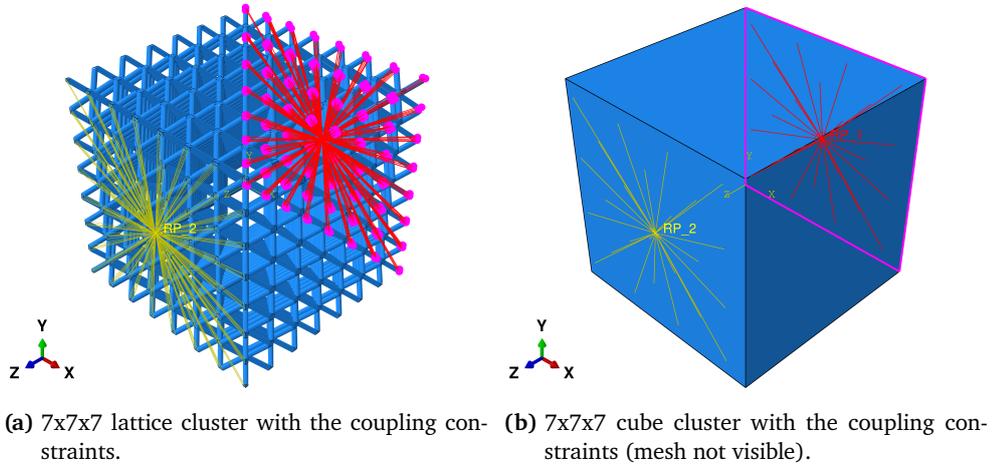


Figure 3.6: Setup for verification of reaction forces.

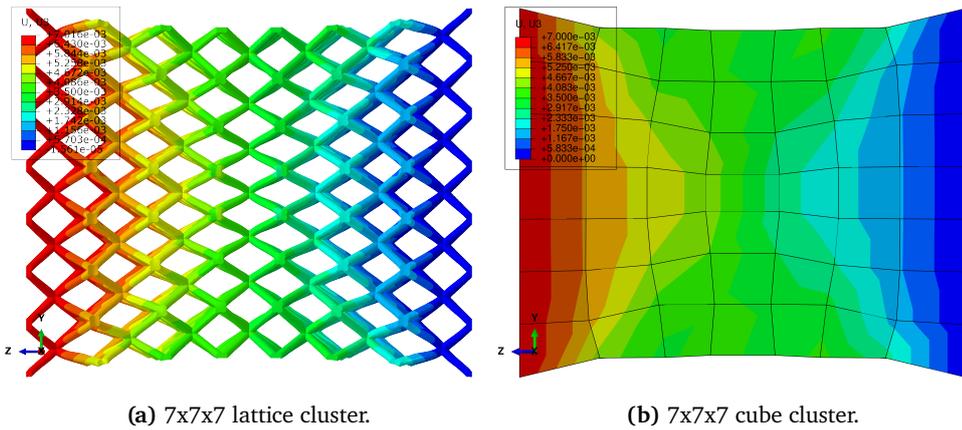


Figure 3.7: Comparison of a lattice and cube cluster under normal deformation.

3.2.2 Results

Here the results of the three different models (“random”, “non-random” and “Ansys”) are compared for the three different cluster sizes and the two different displacement boundary conditions.

3 Results

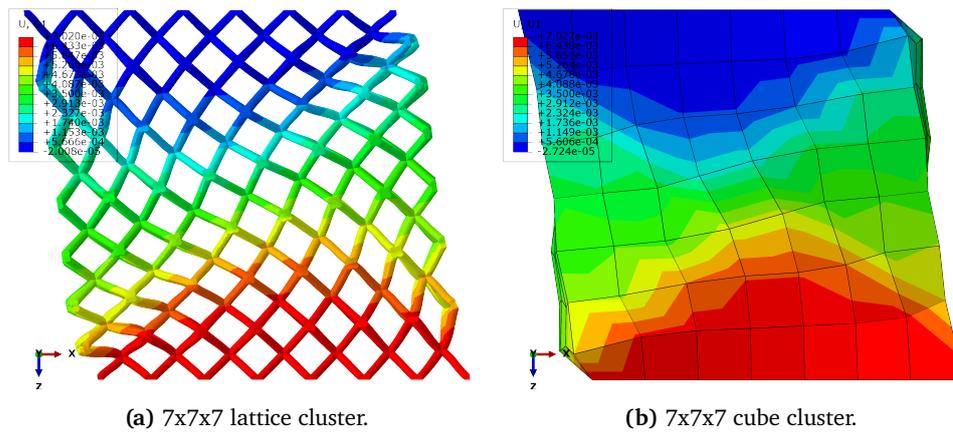


Figure 3.8: Comparison of a lattice and cube cluster under shear deformation.

The results for models with normal displacement can be seen in Figure 3.9. The error values of the random model are 18.2% for the 2x2x2 cluster, 43.6% for the 4x4x4 cluster and 56.3% for the 7x7x7 cluster. It can be seen that the errors increase with increasing size of the structure. In contrary, the error values for the non-random model decrease with increasing size, being 22.8% for the 2x2x2 structure 12.2% for the 4x4x4 structure and 9.1% for the 7x7x7 structure. The Ansys model has the lowest error values with 0.3% for the 2x2x2 structure, 7.8% for the 4x4x4 structure and 8.5% for the 7x7x7 structure.

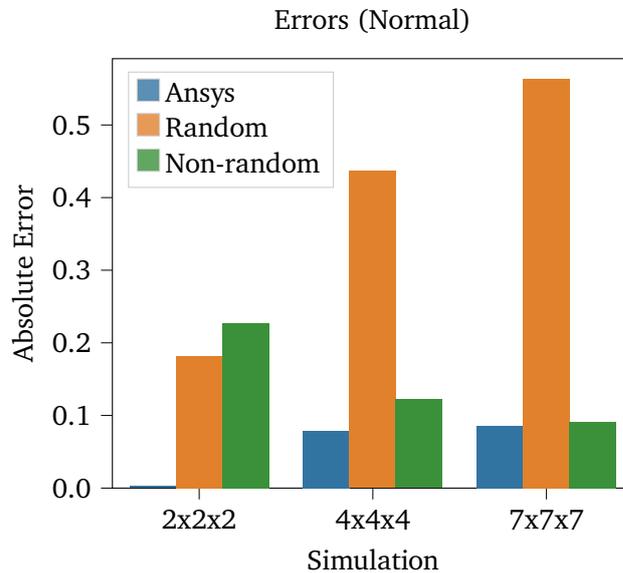


Figure 3.9: The modeling errors of three different models with normal displacement boundary conditions.

The results for the shear displacement are shown in Figure 3.10. The random model has errors of 5.0% for the 2x2x2 structure, 26.5% for the 4x4x4 structure and 39.5% for the 7x7x7 structure. The non-random model shows the same decreasing error values with increasing lattice size with errors of 68.4% for the 2x2x2 structure, 21.5% for the 4x4x4 structure and 5.5% for the 7x7x7 structure.

3.2 Model Implementation and Verification

For these two models the trends with increasing cluster size are similar to ones for the normal deformation configuration. Lastly, the Ansys material model has again a low error for all cluster sizes with errors of 3.8 % for the 2x2x2 structure, 12.2 % for the 4x4x4 structure and 10.6 % for the 7x7x7 structure.

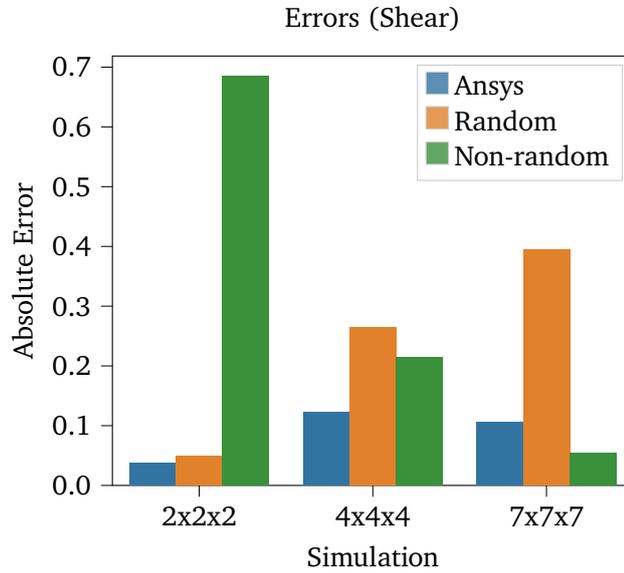


Figure 3.10: The modeling errors of three different models with shear displacement boundary conditions.

4

Discussion

In the following chapter, a discussion of the results is presented. It begins with the models calculated for the simple boundary conditions and then proceeds with the models trained on the random boundary conditions.

4.1 Simple Boundary Condition Modeling

For simple specific load cases, a homogeneous deformation exists in the lattice cells (see Figure 2.4). Therefore, this allows the use of simple quasi-isotropic models for homogenization. However, the results using these models show that this modeling approach has its limits. What could be the reasons for the high implementation errors especially for the “non-random” model¹ trained according to the boundary conditions defined in Section 2.2? One issue seems to be that the material parameters are determined with simple load cases. The application of these simple quasi-isotropic models to large lattices assumes that the load cases can be approximated by a superposition of the simple training load cases. However, the conditions in the big lattice structures are (locally) more complex. Additionally, there are boundary condition related effects, e.g. the effect of the kinematic coupling of the lattice in the normal displacement case. Last but not least, the training boundary conditions itself have a big impact on the parameters gained. For instance, in the normal load case, if the cell was trained without allowing lateral movement, the cell would behave stiffer as the struts would undergo less bending but more stretching.

¹ The Ansys model performs better in the tested lattices, especially for the small number of cells. However, there are still errors up to 12.2% with this model.

4.2 Random Boundary Condition Modeling

For random sampling of the displacement boundary conditions, it is clear that the lattice can be under a heterogeneous deformation. This observation is confirmed in Figure 2.7 where a lattice structure under random boundary conditions is depicted. As presented in Section 3.1.2, the reduced-integration surrogate model with the six strain inputs cannot adequately predict the strain-stress relationship under this heterogeneous deformation. The effect of implementing this deficient surrogate model in the representative cubes is clearly visible in the Figures 3.9 and 3.10, where it is shown that the errors of the reduced randomly sampled model are relatively high (up to 56.3 %).

On the other hand, the full-integration model using the information from all eight integration points can perfectly explain the relationship of the data. This seems to suggest that we need a higher order element model, i.e. not a reduced element but at least a fully integrated linear element to capture the mechanics properly.

4.2.1 Implementation of the Full-Integration Model into Abaqus

The implementation of the quasi-isotropic material law and also of the reduced-integration surrogate models is straightforward in Abaqus. The quasi-isotropic, isotropic, orthotropic and anisotropic material laws are already available in Abaqus by default. In the hypothetical case in which a multivariate linear model using six strain inputs should be employed, a relatively accessible implementation would also be possible. This could be achieved by using the FORTRAN subroutine functionality of Abaqus. In this project this was not necessary as the matrices obtained with a MLR model formulation typically motivated the utilization of anisotropic or orthotropic material matrices.

The implementation of the full-integration model into Abaqus seems to be more complicated because this model formulation uses the information from all eight integration points. Possible ways to implement these models could be the use of common blocks in UMAT FORTRAN code or user-defined elements. However, it is decided that such an implementation is out of the scope of this project. Therefore, unfortunately, no validation of the randomly sampled full-integration models could be made on a lattice with multiple cells.

If the full-integration model capable of explaining the stress-strain relationship perfectly was implemented, there could still occur issues and deviations from the full finite element calculation. For instance, there could be bending moments at the lattice corners. How big the effect of such moments would be has to be further investigated. In this project, the moments are simply neglected.

5

Conclusion

In summary, multiple models for the BCC lattice geometry are developed in this project. On the one hand, there is the quasi-isotropic model developed using simple boundary conditions. This type of modeling is also used in the literature. The implementation into Abaqus and the validation tests on bigger lattice structures show that such a model has its limits.

On the other hand, a new technique is developed which tries to generate surrogate models of the lattice structure able to predict more general load cases. This is done by randomly sampling the boundary conditions and then training a meta-model on such generated data. For this approach, two types of representative cube elements are tested. Firstly, the full-integration elements and secondly, the reduced-integration elements. The errors of the full-integration model (calculated on validation data sets) suggest that the model can explain the stress-strain relationship perfectly. The reduced-integration model cannot do that in a satisfactory manner. The implementation of the full-integration model and therefore its validation on big lattices exceeds the scope of this project and is therefore not presented in this work. The implementation of the reduced-integration models is carried out in Abaqus. However, as already visible in its inability of explaining the stress-strain data, the errors of its implementation are also high.

5.1 Outlook

In the future, the implementation of the full-integration model into a finite element package should be considered. By doing so, a verification of this modeling technique could finally be made. Also, possible issues of this approach for example with the setting of the boundary conditions could then be further investigated. A different approach to generate a (maybe even better) metamodel could also be to not only train the model on one lattice cell but for example on a $2 \times 2 \times 2$ cluster of cells. In this case the boundary effects between the lattice cells could probably be incorporated into the model.

5 Conclusion

Once a successful implementation is made possible, one could even think of applying a metamodel not only for a single BCC geometry but also to explain the variations of different parameters of the structure. Also, the application to completely different kinds of lattice structures can be imagined. Challenging problems could arise for lattice cells with not only corners but also edges on the boundaries of the cell.

Additionally, it could also be feasible to incorporate geometrical or material non-linearity into the surrogate model. In theory, the machine learning approach used in this project allows for high flexibility, given enough data is generated in a mechanically compatible manner.

A

Appendix

A.1 Authors

Chapter	Name
1 Introduction	Fabian Jüngling
2 Methods	Christian Hauschel
3 Results	Fabian Jüngling
4 Discussion	Christian Hauschel
5 Conclusion	Christian Hauschel & Fabian Jüngling

Table A.1: Contribution of the authors to the different parts of this report.

A.2 Code Manual

See the next pages.

Code Manual

Folder & File Structure

[Code](#)

[Temp Folder](#)

Installation

Usage

[Data Generation](#)

[Surrogate Modeling](#)

Folder & File Structure

Code

```
code
├── main_data_generation.m # generates the input data for modeling
├── main_evaluate_model.m # load and evaluate a model on data
├── main_surrogate_modeling.m # calculates surrogate model on data
├── data_generation
│   ├── cleanDir.m # cleans the specified directory of all files
│   ├── cube_rf.m # run cube force BC simulation
│   ├── cube_u.m # run cube displacement BC simulation
│   ├── lattice.m # run lattice simulation
│   ├── parameters.m # set here the parameters of the simulations
│   ├── results_cube_rf.py # extracts data from cube_rf odb
│   ├── results_cube_u.py # extracts data from cube_u odb
│   └── results_lattice.py # extracts data from lattice odb
├── surrogate_modeling
│   ├── deStandardize.m # reverses standardization by the function "standardize"
│   ├── elastic_model.m # fits an elastic model (iso-, ortho-, anisotropic) to the data.
│   ├── kriging.m # kriging modeling settings
│   ├── linear_model.m # calculates the stress according to the linear isotropic
│   ├── lra.m # LRA modeling settings
│   ├── mises.m # calculates equivalent measures according to Mises
│   ├── mlr.m # calculate multivariate linear regression (MLR) model
│   ├── mlr_evaluate.m # evaluate MLR model
│   ├── myDenormalize.m # denormalizes data normalized by "myNormalize".
│   ├── myNormalize.m # function to the normalize data
│   ├── nn.m # (experimental) neural network modeling settings
│   ├── pce.m # PCE modeling settings
│   ├── pck.m # PCK modeling settings
│   ├── simple_model.m # simple custom model for reduced elements.
│   ├── standardize.m # function to the standardize data
│   ├── svr.m # SVR modeling settings
│   └── validation_error.m # calculates validation error
└── uqLab
    # etc.
```

Temp Folder

In this directory, all the Abaqus input files are located. Here, the simulation and output files get stored.

```
| cube_rf_000.inp # cube input file with force BC
# etc.: other variants...
| cube_u_000.inp # cube input file with displacement BC
# etc.: other variants...
| tetra_002.inp # lattice input file
# etc.: other variants...
```

Installation

1. Put folder the `temp` to `C:\` or change the `PATH` variable in the script `code\data_generation\parameters.m`
2. Do not change the relative folder structure of the other directories

Usage

Data Generation

1. Change parameters in `code\data_generation\parameters.m`
2. Run `code\main_data_generation.m`

Surrogate Modeling

1. Change parameters in the parameters section in `code\main_surrogate_modeling.m` and also in the corresponding file for the chosen modeling type (e.g. for PCE: `code\surrogate_modeling\pce.m`)
2. Run `code\main_surrogate_modeling.m`

Bibliography

- [1] E. Alabort, D. Barba, and R. C. Reed. “Design of Metallic Bone by Additive Manufacturing.” In: *Scripta Materialia* 164 (Apr. 15, 2019), pp. 110–114.
- [2] T. Alwattar and A. Mian. “Development of an Elastic Material Model for BCC Lattice Cell Structures Using Finite Element Analysis and Neural Networks Approaches.” In: *Journal of Composites Science* 3.2 (Apr. 1, 2019), p. 33.
- [3] Ansys. *How to Simulate and Design the Microstructures of Composites and Other Complex Materials*. 2019. URL: <https://www.ansys.com/blog/how-to-simulate-microstructures-composites> (visited on 06/23/2020).
- [4] L. J. Gibson and M. F. Ashby. *Cellular Solids: Structure and Properties*. 2nd ed. Cambridge Solid State Science Series. Cambridge: Cambridge University Press, 1997.
- [5] C. Lataniotis, D. Wicaksono, S. Marelli, and B. Sudret. *UQLab User Manual – Kriging (Gaussian Process Modeling)*. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019, p. 74.
- [6] M. H. Luxner, J. Stampfl, and H. E. Pettermann. “Finite Element Modeling Concepts and Linear Analyses of 3D Regular Open Cell Structures.” In: *Journal of Materials Science* 40.22 (Nov. 1, 2005), pp. 5859–5866.
- [7] T. Maconachie et al. “SLM Lattice Structures: Properties, Performance, Applications and Challenges.” In: *Materials & Design* 183 (Dec. 5, 2019), p. 108137.
- [8] S. Marelli and B. Sudret. *UQLab User Manual – Polynomial Chaos Expansions*. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019, p. 66.
- [9] S. Marelli and B. Sudret. “UQLab: A Framework for Uncertainty Quantification in MATLAB.” In: *The 2nd International Conference on Vulnerability and Risk Analysis and Management (ICVRAM 2014)*. University of Liverpool, United Kingdom, 2014, pp. 2554–2563.

Bibliography

- [10] MathWorks. *Mvregress – MATLAB Documentation*. URL: <https://ch.mathworks.com/help/stats/mvregress.html> (visited on 06/19/2020).
- [11] K. H. Matlack, A. Bauhofer, S. Krödel, A. Palermo, and C. Daraio. “Composite 3D-Printed Metastructures for Low-Frequency and Broadband Vibration Absorption.” In: *Proceedings of the National Academy of Sciences* 113.30 (July 26, 2016), pp. 8386–8390.
- [12] S. L. Omairey, P. D. Dunning, and S. Sriramula. “Development of an ABAQUS Plugin Tool for Periodic RVE Homogenisation.” In: *Engineering with Computers* 35.2 (Apr. 2019), pp. 567–577.
- [13] E. Ptochos and G. Labeas. “Elastic Modulus and Poisson’s Ratio Determination of Micro-Lattice Cellular Structures by Analytical, Numerical and Homogenisation Methods.” In: *Journal of Sandwich Structures & Materials* 14.5 (Sept. 1, 2012), pp. 597–626.
- [14] E. Torre, S. Marelli, and B. Sudret. *UQLab User Manual – Statistical Inference*. Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2019, p. 36.
- [15] K. Ushijima, W. Cantwell, R. Mines, S. Tsopanos, and M. Smith. “An Investigation into the Compressive Properties of Stainless Steel Micro-Lattice Structures.” In: *Journal of Sandwich Structures & Materials* 13.3 (May 1, 2011), pp. 303–329.
- [16] H. N. Wadley. “Multifunctional Periodic Cellular Metals.” In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 364.1838 (Jan. 15, 2006), pp. 31–68.
- [17] S. Xu, J. Shen, S. Zhou, X. Huang, and Y. M. Xie. “Design of Lattice Structures with Controlled Anisotropy.” In: *Materials & Design* 93 (Mar. 5, 2016), pp. 443–447.



Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

A Surrogate Modeling Approach for Cost-Efficient Mechanical Analysis of Lattice Structures

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):

Hauschel

Jüngling

First name(s):

Christian

Fabian

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date

Zürich, 03.07.2020

Signature(s)

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.