Lehrstuhl für Technische Mechanik

Titel der Abschlussarbeit

Art der Arbeit

Name des Studenten
Matrikelnummer

Datum der Abgabe
Name des Betreuers

Friedrich-Alexander-Universität
Technische Fakultät

Masterarbeit

im Studiengang Maschinenbau

The title of your thesis goes here

von

Firstname Lastname

Prüfer:	Prof. Dr.-Ing. habil. P. Steinmann
Betreuer:	Dr.-Ing. S. Pfaller
Ausgabe:	DD.MM.YYYY
Abgabe:	DD.MM.YYYY

Universität Erlangen-Nürnberg
Lehrstuhl für Technische Mechanik
Prof. Dr.-Ing. habil. P. Steinmann

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Ort, Datum

Unterschrift

Contents

List of Figures

List of Tables

1 Introduction

research questions

2 Basics

2.1 MD-Model

2.2 VOCE-Model

2.3 Optimization algorithm

2.4 EasyPBC Plug-In

3 Methods and Models

3.1 Flowchart

3.2 Test cases

XXX muss vmtl eig in BAsics To evaluate the quality of the material parameters we need a possibility to investigate the material response caused by the definition of the material parameters. Then we can compare this results with the load parameters and evaluate the quality of the current material parameters. Thus we have to use a simulation program to analyse the material behaviour for every iteration of material parameter values during the optimization process. We decided to use ABAQUS as simulation software, because of the intern scripting tool. With the ABAQUS scripting tool one can run python scripts directly in ABAQUS (see chapter XX). With special ABAQUS commands one can use ABAQUS with the same opportunities as with the GUI. XXXX

Therefore we choose simple load cases, which are easy to recalculate. AS explained in the chpater XX about the mathematical problem formulation, we have to define a parameter which defines the quality of the mechanical responses calculated by ABAQUS compared to the ones from the MD-simulation. Therefore we first have to define adequate mechanical measurements which represent best the mechanical behaviour and contain information about the material parameters. Hence the stress and strain measurements in all normal and shear directions are possible quantities. Depending on the load case the measurements with the most useful information may vary.

4 Models and Methods

In the following chapter we describe the optimisation process used in this thesis. Therefore we first have a closer look on the theoretical structure of the process. Afterwards we discuss the implementation with all

4.1 Theory

The aim of the optimisation process is to find material parameters which best represent the mechanical behaviour analysed with a MD-Simulation. In the following we demonstrate the need of this process. The MD-analysis gives information about the mechanical response for an applied load case. In a MD-simulation we can apply a load case and log the stress and strain values in all directions at prescribed time steps during the loading process. In the following this data from the MD-Simulations are called load parameters. They describe the applied load (for example uniform strain up to 20%) and the corresponding reactions. In section XX we have a closer look on the structure of the data. This load parameters empirically describe the mechanical behaviour of the investigated material. However, we are interested in a deterministic description of the material behaviour. This is usually done by the definition of material parameters. Since it is not possible to specify material parameters directly in the MD-simulation, we want to build a optimization process which is able to find the best material parameters to represent the material behaviour from the load parameters. To implement this problem we have to reformulate it as a minimization problem. As minimization value we have to define a value that represents the error of mechanical response of the material. Therefore we compare the material behaviour defined through the material parameters with the one from the load parameters. If we minimise their difference, we improve the quality of the material parameter values describing the material behaviour. In our problem formulation we have now the optimisation of the material parameters controlled by the minimization of the difference between the mechanical responses. In the next step we need to define this difference between the mechanical behaviour more precisely. For the mathematical formulation only one single value is allowed for the problem formulation. For a detailed explanation we now introduce the used model, the load cases and load parameters.

As explained in chapter XX we use ABAQUS as simulation software and the ABAQUS internal scripting tool for the code.

4.2 Model, load cases and load parameters

In this section we introduce the used model and the applied load cases. As model we use a simple isotropic cube of size 1x1x1. We use it as one representative part of an infinitely extended material which is modelled through easyPBC (see chapter xx section XX). By the usage of this simple geometry we anticipate short simulation times. During the optimization process this property is important for a fast evaluation of the result quality. In addition the mechanical responses of a cube are comprehensible and it is possible to verify the results by hand computations. However, it is easy to apply different load cases on a cube. As we see in chapter XX, we have to modify the load applied by easyPBC. This is much easier at a simple geometry where we can apply uniform stresses or strains. Then it is simple to check the correctness of the load application since qualitative the mechanical responses for these load cases are known. (BEISPIEL BILD MIT LOAD CASE)

NOCH MEHR ÜBER DAS MODEL

4.2.1 Load cases and evaluation measurements

A load case defines the direction where the load is applied. Since we want reproducible and easy cases, we only allow loading in normal directions and principal shear directions. To aplly this loadings we use easyPBC. For a consistent naming, we adopt the naming from easyPBC for the load cases, which is defined in Table 4.1. To model a more complex loading situation it is possible to combine these load cases. For example it is possible to apply a tensile load in xx-direction combined with a shear strain in xz-direction. Important to notice is, that we apply load only in one direction per load case without constraining the other ones. Thus the deformation of the cube is only restricted through the periodic boundary conditions (see chapter XX) which ensure that parallel surfaces remain parallel during the deformation process. Apart from this, the material response is not restricted. After the application of a load case, we have to decide which material responses we use to compare with our load parameters. We have the possibility to read out the stress and strain values in all normal and shear directions. These values are called evaluation measurements. For a high result quality of our material parameters, we try to use evaluation measurements where we can extract the most information about our material behaviour. These measurements vary depending on the applied load case. As we see in section XX the user defines the evaluation measurements in the input file. Based on the presented load cases in the following, the meaning of the evaluation measurements becomes clear.

Load direction	**Load case**
xx	E11
yy	E22
zz	E33
xy	G12
yz	G23
xz	G13

Table 4.1: Load case naming.

For the verification of the code we use a simple tensile load case E11. In all other directions we do not apply any constraints except the periodic boundary conditions. As evaluation measurements we use the stress in x-direction and the lateral strains in normal y- and z-direction. The normal stress in x-direction contains information about the Young's modulus and the plastic parameters. For the Poisson ratio the lateral strains are necessary. Since we do not constrain the motion of the cube, the y- and z-dimension will decrease, to balance the strain in x-direction. This reaction is necessary to keep a state of minimum stress. Simultaneously this means that the lateral stresses do not contain any useful information in this load case, because they are numerically zero. The validation study is done with the same load case. However, we use different mixing ratios as materials which show different mechanical behaviours. Similar to the verification study we need the stresses in x-direction and the lateral strains to extract information about all material parameters. In the next step we apply a combination of a tensile load case and a simple shear test. For additional information we apply a simple shear strain as load case G12. Then we measure the stresses in this direction. Through the additional obtained information we try to improve the uniqueness of the determined material parameter values. As a last study we investigate the application of cyclic loading as E11 load case. We perform this study with varying load parameters (see subsection 4.2.2).

Noch was über dei zyklische Art der Strains. Bild mit load case und gemessenen stress/strain directions.

4.2.2 Load parameters

NUR EIN WERT IM ELASTUSCHEN BEREICH

Test

We apply a linear strain up to 20% in normal x-direction. Here we apply a tensile strain following a sinus function over time with a maximum amplitude of 15 % strain. We only consider the first quarter of a period up to the maximum value as a preparation for studies with cyclic loading. In this preparing study we want to investigate the handling with non-linear loadings.We apply the same loading as in normal direction. Now we use a full period of a sinusoidal loading for a tensile load case in x-direction. We use amplitudes of 1%, 5% and 8%. The varying proportions in the elastic and plastic domain of the material should

4.2.3 error calculation

The stress and strain measurements we do for the introduced load cases need to be reduced to one single value per load case for the mathematical solution of the optimization problem. In this section we explain how we do this reduction. First we define the measurements we want to use in the current load case. Then we read out this measurements from the ABAQUS-analysis and from the MD-analysis. In the next step we have to build the difference between them. For the tensile load case for example we have to do this for multiple measurements (stress xx, strain yy, strain zz). Additional it is not enough to consider only the stress and strain values at the end of the applied load (at 20 percent). If we want to adequately describe the material behaviour it is sufficient to regard the measurement during the complete loading process. From the MD-simulation certain strain-values are given at which stresses and strain in other directions are measured. We use this strain-values as steps in the ABAQUS-simulation to read the measurements there too. Thus we are able to directly compare the evolution of the measurements during the loading process. Since we now have multiple measurements to evaluate for multiple strain steps we need a mathematical expression to reduce all this values into one. For a representative value we decide to build a root-mean-squared error (RMSE). NOCH MEHR DAZU THEORIE. TO build this value we first compute the squared difference between the ABAQUS measurement value and the MD measurement value at every strain step for all necessary measurements. The result is one array with squared differences for every measurement quantity used in the load case. As described in chapter XX the distribution of the data points is unfavourable for the determination of the elastic parameters because only one data point is included in the elastic domain of the material. To support the algorithm to find the elastic parameters anyway we applied a weight of 100 at the data point in the elastic domain for every measurement quantity. In the next

step we build the mean value out of the weighted arrays separate for every measurement quantity. Now we have mean squared errors for all measurements of the current load case. If we now build one value out of this mean squared errors we have to ensure a common scale of the values. Otherwise their influence on the overall error may vary significantly. Therefore it is necessary to apply weights to the mean squared errors of certain measurement quantities. The exact weights depend on the load case and the used data set. In general, the mean squared errors of strain measurements are much smaller than the ones from stress measurements, such that a weight of around 10e4 is necessary for the strain measurements. After that we sum up the weighted mean squared errors and build again a mean value. From this value we build then the square root. Since our code is able to process multiple load cases for one data set in one optimisation process we can calculate the RMSE for every load case and apply weights depending on the load case. Then we sum up all these weighted RMSE values and use it as return value for our minimization function. Additional, multiple data sets can be processed which leads to a repetition of the described procedure for every data set. Therefore we can calculate the return value for multiple load cases for multiple data sets. Then we can apply weights for every data set and sum it again to return a single value. This value is the value we return our minimization function. Through the adaption of the material values this value should be minimized.
IN the following sections we have a closer look on the implementation of this minimization process.

$$\mathrm{mse}_\sigma = \frac{\sum_{i=1}^{N_\mathrm{frame}} w_i(\sigma_{odb,i} - \sigma_{md,i})^2}{\sum_{i=1}^{N_\mathrm{frame}} w_i} \qquad \mathrm{mse}_\varepsilon = \frac{\sum_{i=1}^{N_\mathrm{frame}} w_i(\varepsilon_{odb,i} - \varepsilon_{md,i})^2}{\sum_{i=1}^{N_\mathrm{frame}} w_i} \tag{4.1}$$

$$\mathrm{rmse} = \sqrt{\frac{\sum_{j=xx}^{N_\mathrm{SED}} w_\sigma \cdot \mathrm{mse}_{\sigma,j} + \sum_{j=1}^{N_\mathrm{EED}} w_\varepsilon \cdot \mathrm{mse}_{\varepsilon,j}}{N_\mathrm{SED} + N_\mathrm{EED}}} \tag{4.2}$$

$$\mathrm{error} = \sum_{k=1}^{N_\mathrm{LC}} \sum_{l=1}^{N_\mathrm{LP}} w_k w_l \cdot \mathrm{rmse}_{k,l} \tag{4.3}$$

$$N_\mathrm{inc} : \text{Number of frames} \tag{4.4}$$

$$N_\mathrm{SED} : \text{Number of stress evaluation directions} \qquad N_\mathrm{LC} : \text{Number of load cases} \tag{4.5}$$

$$N_\mathrm{EED} : \text{Number of strain evaluation directions} \qquad N_\mathrm{LP} : \text{Number of load parameters} \tag{4.6}$$

BILD MIT FORMEL

4.3 general structure of the code

- object oriented programming

4.4 Preprocessing

Input parameter	Directions	Category	Data format	Unit
Youngs modulus	–	value	array	MPa
	–	minimum	scalar	MPa
	–	maximum	scalar	MPa
Poisson ratio	–	value	array	–
	–	minimum	scalar	–
	–	maximum	scalar	–
Plastic Yield	–	value	array	MPa
	–	minimum	scalar	MPa
	–	maximum	scalar	MPa
Alpha, beta, gamma	–	value	array	–
	–	minimum	scalar	–
	–	maximum	scalar	–
Load parameters	–	filename	string	–
	–	weight	scalar	–
Load case	E11, E22, E33,	active	0/1	–
	G12, G23, G13	weight	scalar	–
Stress evaluation	xx, yy, zz,	active	0/1	–
	xy, yz, xz	weight	scalar	–
Strain evaluation	xx, yy, zz,	active	0/1	–
	xy, yz, xz	weight	scalar	–
Load weighting	normal stress, normal strain, shear stress, shear strain	weight	scalar	–

Table 4.2: Input parameters.

Before starting with the optimization process, we need some preprocessing steps to prepare a working ABQUS model with the required properties. In picture XX the complete structure of the code is depicted. The upper part belongs to the preprocessing. In the first step the code extracts the values from the input file. Table Table 4.2 lists the input parameters relevant to the optimization process. The whole input file is included in the Attachment XX. Here the user has multiple options to specify the optimization process. It is possible to test multiple initial values for the material parameters calling the script once. This function is important to verify the optimization results with varying input values. For every material parameter we can write an array of initial values. Then the code loops over all array entries at a time to extract one initial value for each parameter (see Table 4.3). As a consequence all arrays need to be of same length. For all the created initial value combinations the code creates a new ModeldataBase (mdb) in ABAQUS and a new folder structure to set the working directory and store the results.

Material Parameter	Combination 1	2	3	4	5
E	E_1	E_2	E_3	E_4	E_5
ν	ν_1	ν_2	ν_3	ν_4	ν_5
σ_{pl}	σ_{pl_1}	σ_{pl_2}	σ_{pl_3}	σ_{pl_4}	σ_{pl_5}
α	α_1	α_2	α_3	α_4	α_5
β	β_1	β_2	β_3	β_4	β_5
γ	γ_1	γ_2	γ_3	γ_4	γ_5

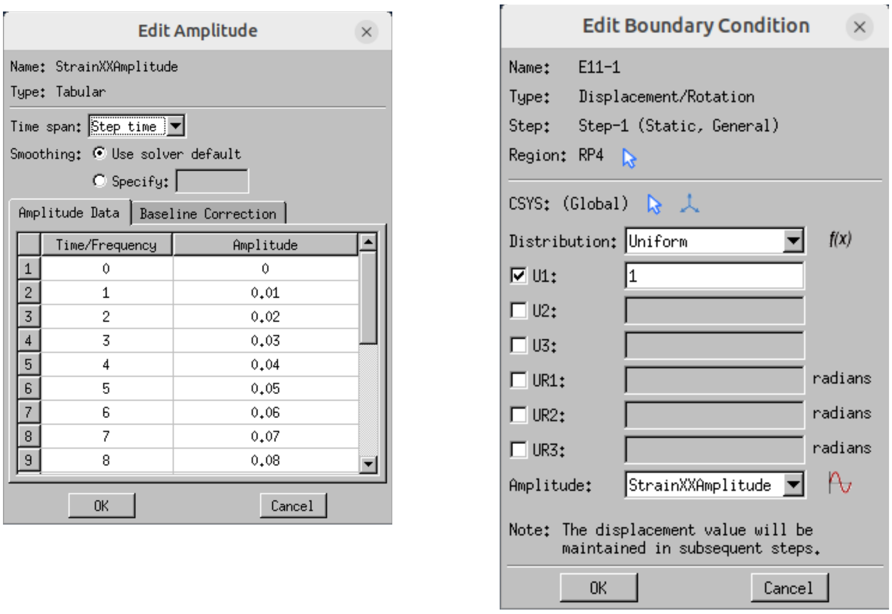
(a) Arrangement of initial value combination of material parameters.

Model	Load case	Load parameters
Model 0	E11	Data set 1
Model 1	E11	Data set 2
Model 2	E11	Data set 3
Model 3	G12	Data set 1
Model 4	G12	Data set 2
Model 5	G12	Data set 3

(b) Model creation for load case and parameter combinations.

Table 4.3: Loop conditions in preprocessing.

Afterwards we start to construct the model. As discussed in section XX we use a cube with size 1x1x1 as model. We mesh the cube with 6x6x6 XXXX elements. The number of elements is a compromise between a coarse mesh for fast computation and a minimum number to avoid convergence errors. Although we use the hyperelastic material law in our optimization process, we first have to build the model with elastic material. This is necessary for the usage of easyPBC in the next step. Now we use the load case defined in the input file to create a easyPBC job to apply it on the cube. As discussed in chapter XXX we use easyPBC for the automatic construction of periodic boundary conditions. We use this set up to simulate a small detail of a infinitely large area (SCHLECHT FROMULIERT). Aside from that we use the generated boundary conditions for the applied load case. Since they act at a reference point a homogeneous load contribution is ensured. –> vlt auch in basics teil.. However the settings from easyPBC contain some default values, we have to adjust. EasyPBC applies for every load case a uniform displacement with a standard fixed value. In our optimization we want to study cases with applied stresses as well. Additional we have to adapt the value of the load. For a correct comparison of the ABAQUS data with the MD-data we have to create evaluation points at similar load increments. In ABAQUS we can solve this issue by creating an amplitude. We register the evaluation points from the MD-data as steps and use this amplitude to apply the load. The value of the load is then set to 1 because it only defines the factor the amplitude is multiplied (see Figure 4.1). Afterwards we modify the increment settings. EasyPBC automatically creates increments with fixed size and without non-linear geometry effects. In order not to run into convergence errors we use automatic incrementation. Especially in the first load steps we observe large deformations. If we try to resolve such large deformations in one incrementation step ABAQUS cannot resolve the step. With automatic incrementation ABAQUS can adapt the number of increments per load step dynamically. The non-linear geometry effects have to be considered because of the material properties we use. As described before we build elastic material for the usage of easyPBC. In the following the code removes this material and substitutes it with a hyper-elastic material which is suitable for high non-linear deformation. By using this material, ABAQUS demands the inclusion of non-linear geometry effects. In the last step of preprocessing we store the model in a dictionary. We use this dictionary later to call the models for the optimization. We perform the preprocessing for all prescribed load cases. This means for example if we define E11, G12, and G23 as load cases, we create one model for each load case in the previous described manner. Furthermore, we loop over the load parameters and create separate models. In the end we have then for each combination of load case and load parameter one model.



(a) Definition of load amplitude in ABAQUS.

(b) Boundary condition menu in ABAQUS.

Figure 4.1: Loop conditions in preprocessing.

4.5 Optimisation process

Parameter	Content	Data format	Explanation
Objective function	Optimization function	–	Function whose scalar value should be minimized
Initial guess	Material values	array	Initial (scaled) values for the optimization parameters
Additional arguments	Cube parameters	object	Model information from input file
	Load parameters	dictionary	Load parameters from MD-simulations
	Work directory	string	Path to store results
	Evaluation counter	scalar	Counter for the performed function evaltions
method	Nelder mead	–	Mathematical algorithm to perform minimization
bounds	Minimal and maximal material values	array	Limits possible range of optimization parameters
maxiter	Number of maximum iterations	scalar	Limit for optimization iterations

Table 4.4: Input parameters for SciPy minimize function.

In the following section we describe the optimisation process. We start the processing by calling the scipy-minimize function. We pass this function various parameters to perform the optimization listed in table XX. (MEHR ÜBER INPUT PARAMETER INKL TABLEE) The minimize function itself calls our self-written optimization function, where the evaluation takes place. As described before we create a model with a corresponding job for every combination of load case and load parameters and store them in a dictionary. Now we pass the whole dictionary to the optimization function. All of the models describe now different test cases for the same material. We want to use them all to get as much information as possible, such that we have to include all of them into the optimization process. To do so we have to calculate an error expression from all these analysis as described in section XX. We start the process with rewriting the material values in all models. Since they all describe the same material we write the same values for every model. For the minimization computation optimization parameters are scaled in the bounds 0-1, that we have to rescale them first. Then we can use the rescaled parameters to compute the values for the plastic stress function with the formula for VOCE-hardening. Now we can update all material values. in the next step we handle the models successively. We start a job to perform the ABAQUS analysis and open the resulting output-data base. Afterwards we read the stresses from the odb. We do this by reading the fieldOutput variable 'S' and write the data in a stress directory. Since we need the stress-values at all the defined strain steps we read out every frame. One frame corresponds with one strain step. Additionally we loop over all directions (xx, yy, zz, xy, yz, xz). The same procedure is done for the strain values. Here it is important to read out the correct strain variable 'NE' (nominal strain). For hyperelastic materials ABAQUS uses as standard value the logarithmic strain ('LE'), which gives incorrect values in our studies. Then we store all values for all frames and directions in a dictionary again. Now we collected all required data to calculate the error. We do this in the way described in section XX. For a better structure of the code this part is outsourced in a separate function. We call this function and pass the stress and strain directory as well as the corresponding load parameters from the md-analysis. Then the computation runs and the function returns the rmse value for this job. Multiplied with its corresponding weights for load case and load parameters we add this value to the total error value. Now we restart the result reading and error computation for the next job. When all jobs are processed we have a total error value containing information from all jobs about the quality of the current material values. This value is the one we return our minimization function. It uses this values to compute internal the next material value combination to reduce the returned error value. Now one optimization iteration is performed. The new combination of material values is passed to our optimization function and it starts again. This process will run till our defined number of maximum iterations is reached.

1. call scipy minimize with all necessary arguments 1. function which shuold be minimized 2. list with optimisation parameters 3. additional arguments: data which are called in the minimized function: cubeparameter- object, mddaa dictionary, work directory, evaluation counter 4. minimization method: nelder mead 5. boundaries for optimization parameters 6. options: return all (return function value from every iteration), maxiter (use max iteeration value as termination criterion)

optimization function:

1. rescale material parameters
2. compute plastic stresses
3. delete elastic material
4. write hyperelasitc material and update plastic material

5. delete lock-files for job: they prevent the start of a job where a obd is already written
6. list jobs in current work directory/ mdb
7. iterate over jobs
8. start job
9. open odb
10. create stress directory: for tensile load cases with normal direction, for shear load with shear direction
11. loop over all directions
12. loop over all frames from last step
13. read fieldoutput 'S' from element set in current direction
14. write it in stress directory
15. create strain directory
16. loop over all directions
17. loop over all frames from last step
18. read fieldoutput 'NE' from element set in current direction
19. write it in strain directory
20. call calculate rmse function
21. call multiple save functions to store results
22. when all jobs are finished:
23. multiple rmse from jobs with weights from parameter input (different weighting of load cases or md data sets)
24. sum weighted rmse
25. save material parameters, rmse
26. return total rmse

calculate rmse function:

- loop over directions in stress directory if the direction should be anaylsed (parameter input)
- read out md data for this direction
- check if md data and odb data have the same length
- calculate squared differences for every step (array)

- weight array –> first data point is weighted with 100, because it is the only point in elastic domain
- compute mean from array: sum over all weighted squared differences/ sum of all weights
- differentiate between tensile and shear load case: - use correlated weight
- append to mse array
- same procedure for strain directory
- sum up all mse values, divide by number of values (build mean value)
- wurzel ziehen
- return rmse

- formel wir sich RMSE zusamensetzt - welche schleifen gibt es?
test cases: - linearere Zug - Zug und shear –> mit sinusform –> warum? - cyclic tensile

5 Results

In this chapter the results from different test cases presented in section XX will be evaluated. First, we discuss the verification results to understand the general behavior of the optimization process. In the next step we validate the optimization performance using different data sets. Finally, we present the results of the cyclic load cases.

5.1 Verification

In this section, we discuss the results of the data set used for the verification of our optimization process. To evaluate the quality of the optimization process, we define characteristic quantities and investigate their evolution. Noch den load case kurz aufschreiben.

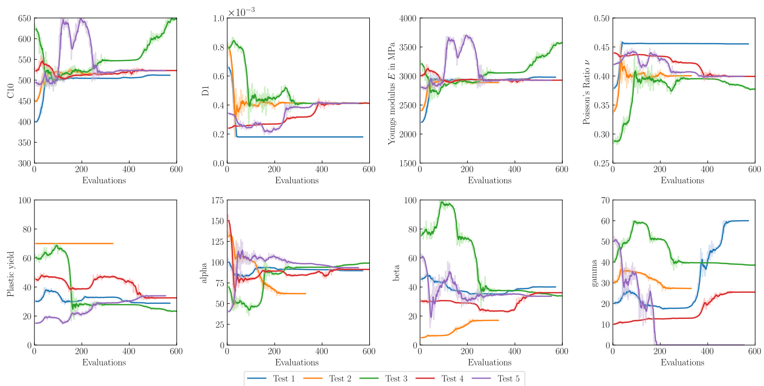
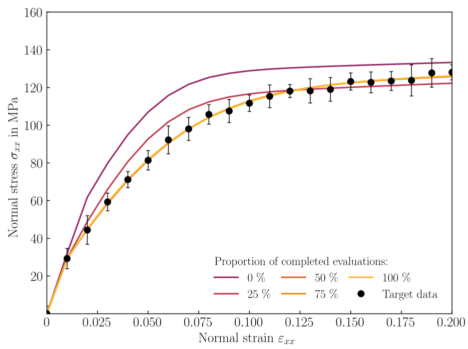


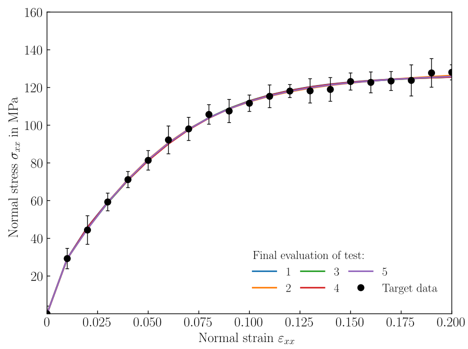
Figure 5.1: optimization progress of material parameters.

As process quantities we log the evolution of the material parameters during the optimization. Since these are our input parameters we performed multiple tests with varying parameter combinations to evaluate the stability of our program. The results are presented in 5.1. In the first row the elastic material parameters are presented. For a better understanding, we transformed the hyper-elastic parameters $C10$ and $D1$ into Young's modulus E and Poisson ratio ν. In the second row the plastic material parameters are presented. For elastic and plastic parameters we can observe convergence for every parameter combination. However, the converged solutions differ from each other for most of the combinations. For a detailed discussion of possible reasons we separate between elastic and plastic material behavior. Since our tested material shows an elastic response only up to the second data point, there might be not enough optimization points to find an unique solution for the elastic material parameters. Nevertheless, we have to investigate other characteristic quantities to ensure our assumption and understand the influence of the plastic parameters. In 5.2(a) the progress of the stress-strain curve in normal direction during one exemplary test together with the target curve from the MD simulation is presented. The optimized curve matches the target curve after only 25%

of the evaluations. Since the stress-strain curve is one of our target data this progress indicates a correct optimization behavior of our algorithm. To support this assumption the final stress-strain curves of the test series are depicted in 5.2(b). Despite the high variance of the final material parameters, the stress-strain curves all matches the target data for the stress-strain curve in normal direction. Because of this deviation we depicted the influence of the parameters on the trend of the VOCE-hardening curve shown in Figure 5.3. The parameter alpha has the greatest impact on the shape of the curve, while a variety of 50% in the parameter gamma has hardly any visible effect. This leads to a high flexibility in adjusting the shape of the curve and as a consequence to multiple possible parameter combinations to fit the target curve. To support our assumption we check the quality of the optimization result by plotting the progress of the root mean squared error (RSME) for all the tests in Figure 5.4. We observe that a common minimum RMSE value is reached in all optimization runs, indicating that the results are of equivalent quality. The results of this verification tests lead to some states about the quality of the optimization algorithm. The results of the stress-strain data show a good match of the optimized curve with the target data for every test which is confirmed by the RMSE. In contrast we observe a high variance in final the material parameters found by the algorithm. These results suggest that the algorithm can generally find parameter values to match the target data. However, the variance in these optimal parameters shows that multiple parameter combinations lead to the same quality of result. This behavior may be due to the relatively large number of optimization parameters compared to the dimension of the target data. To verify this assumption we reduced the number of material parameters. Since only the first point of the target data lies in the elastic domain, we fixed the elastic material behavior and computed them directly from the data of the first point. Then we tested this new configuration of the algorithm with the current target data and two other data sets.



(a) evolution of stress-strain curve



(b) final stress-strain curves

Figure 5.2: a) evolution of the stress-strain curves during optimization for exemplary test, b) final stress-strain curves for complete test study.

PICTURES with Krümmung und Steigung –> parameters to characterize curve

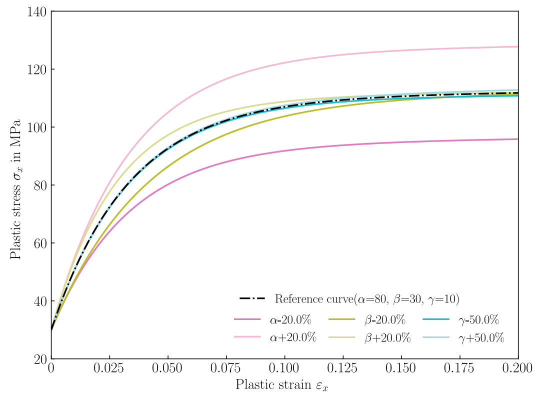


Figure 5.3: parameter influence on voce-hardening curve.

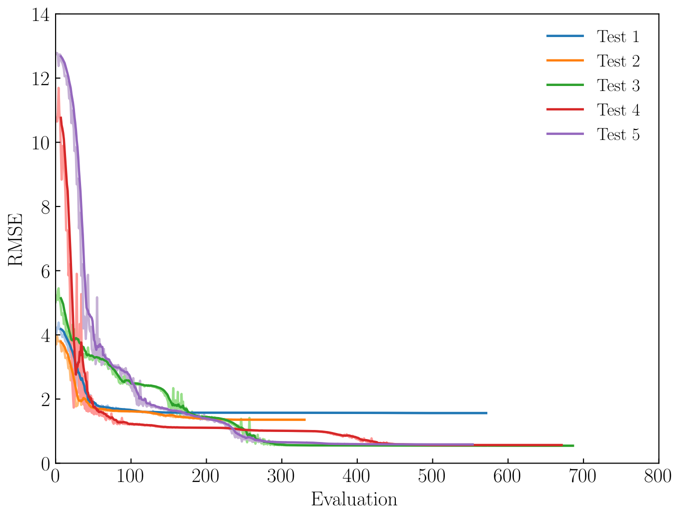


Figure 5.4: rmse for multiple tests.

5.2 Validation

To improve our algorithm we reduced the number of optimization parameters by fixing the elastic material parameters. The results of this implementation for three different data sets will be discussed in this section.

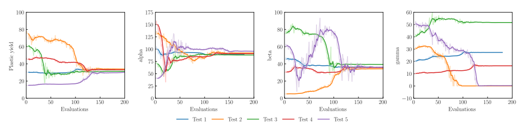
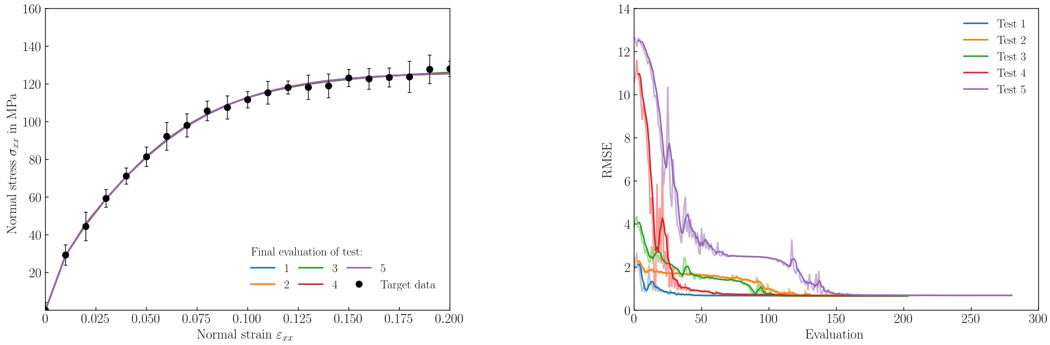


Figure 5.5: progress of material parameters for validation tests.

In the first step we used the same data set as in section 5.1 to test the modified algorithm. The optimized plastic material parameters are shown in Figure 5.5. In all tests, the values of the plastic yield, alpha and beta demonstrate a converging trend



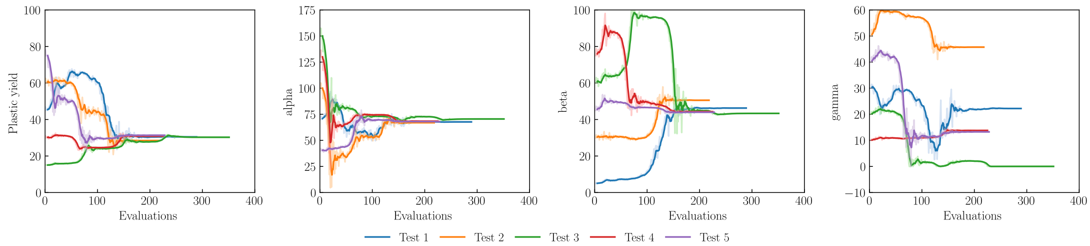
(a) Final stress-strain curves.

(b) RMSE evolution.

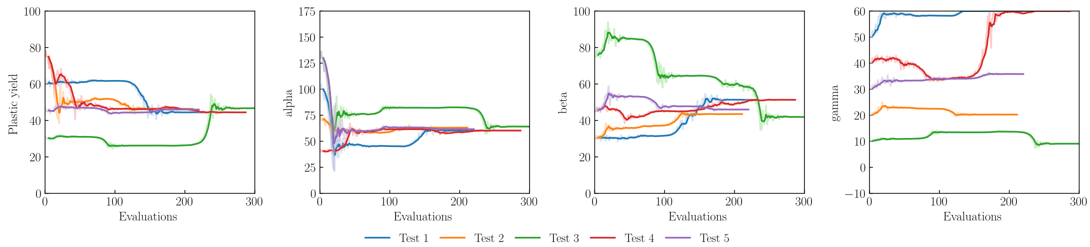
Figure 5.6: Results of validation tests with 6to3 dataset.

towards a singular solution. The only exception to this is gamma whose converged values vary for each individual test. As was outlined in the preceding discussion, gamma exerts minimal influence on the trend of the hardening curve. Consequently, the focus shall be directed towards the plastic yield, alpha and beta, which indicate an improvement in their optimization behavior. The quality of this optimized parameters is ensured through the match of the stress-strain curves with the target data. As demonstrated in Figure **??** the final stress-strain curves exhibit a strong correlation with the target data. The evolution of the RMSE XXX supports this results with small values for every test. A comparison of the results of the present study with those of the verification study reveals an equivalent level of optimization quality. Additionally, the results of the material parameters indicate a positive impact of the algorithm modification, showing a unique solution for the important parameters.

To verify our algorithm independent of the used target data set, we made the same tests with two additional data sets. In the following we present the results of studies for mixing ratio 4:3 and 8:3.



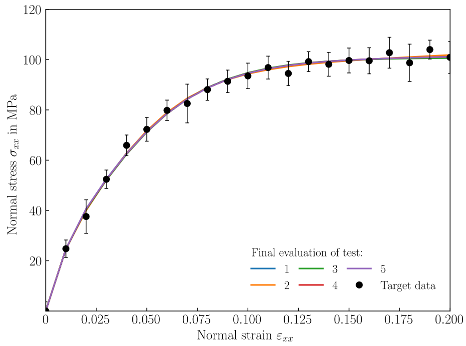
(a) evolution of material 4:3



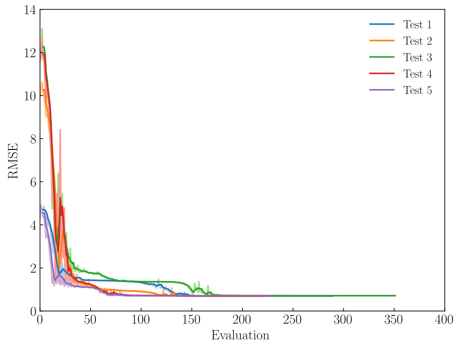
(b) evolution of 8:4

Figure 5.7: Evolution of material parameters for a) mixing ratio 4:3 and b) mixing ratio 8:3.

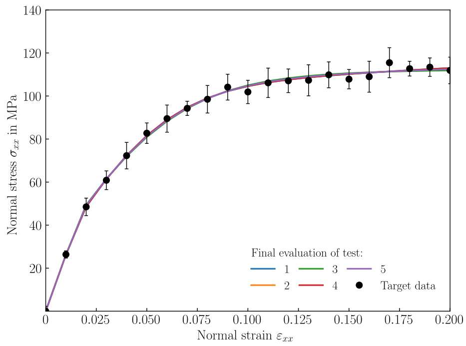
The evolution of the plastic material parameters for the mixing ratios 4:3 and 8:3 in plotted in Figure 5.7. We can observe a similar convergence behaviour as in the validation study with mixing ratio 6:3. Only test case 3 for mixing ratio 8:3 shows a deviation provided that all values converge quite late. Since we chose the initial values randomly this might occur through an unfavourable combination of values. In Figure 5.8 we represent the optimized stress-strain curves and the evolution of the RMSE. For all tests the stress-strain values correlate with the target data. The equivalent level of the RMSE for the converged solutions indicate a similar quality of the optimization result for all tests. These results indicate an improvement of the solution results through determining the elastic parameters.



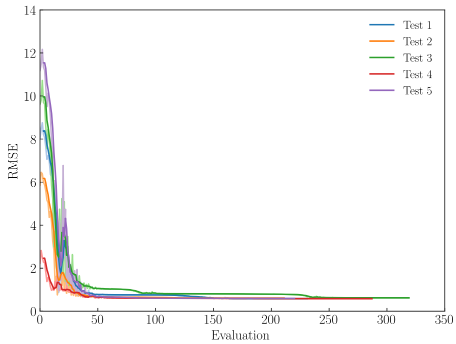
(a) stress-strain curves



(b) RMSE evolution



(c) Final stress-strain curves



(d) RMSE evolution

Figure 5.8: Results of validation tests with mixing ratios 4:3 and 8:3: a) final stress-strain curves for mixing ratio 4:3, b) RMSE for mixing ratio 8:3, c) final stress-strain curves for mixing ratio 4:3, d) RMSE for mixing ratio 8:3.

Overall these results demonstrate the reliability of the optimization algorithm for the load case of a single tensile strain in one direction with fixed elastic parameters. The specification of the elastic parameter values improves the optimization performance in a way that for the plastic yield, alpha and beta independent of the initial values a singular solution can be found. However, the manual specification of Youngs modulus and Poisson ratio is only possible for target data sets with exactly one data point in the elastic domain of the material. For data sets with multiple points in the elastic domain a manual specification becomes complicated quite fast. Additional, for materials with completely unknown material behaviour, the point of transition between elastic and plastic behaviour is still unknown. In order to process such data sets too, we need to integrate the elastic parameters in the optimization process. In doing so the singularity of the solution should be maintained. Therefore the algorithm needs additional information about the mechanical material behaviour. In the following step we tried to do so through the combination of two load cases. Additional to the already tested tensile strain we applied a shear strain. As described in section XX the shear modulus contains information about Youngs modulus and Poisson ratio what might improve the performance of the optimizatoin process. The information contained within the shear modulus about Youngs modulus and Poisson ratio might enclose the necessary restrictions to reduce the solution variability.

Warum sinusförmige belastung? Jz schon mit zyklischen versuchen kommen?

5.3 shear and normal strain tests combined

5.4 cyclic tests

6 Conclusion

A Appendix

A.1 Section

B Appendix