

Almasa Alomerovic, BSc

Model Order Reduction for a Prediction of Short Circuits of Lithium-ion Batteries Induced by Deformation

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieurin

Master's degree programme:
Mathematics

submitted to
Graz University of Technology

Supervisor

Assoc. Prof. Dr. G. Of
Institute of Applied Mathematics

Graz, January 2024

Acknowledgments

This master's thesis was written in close cooperation with the VIRTUAL VEHICLE Research Center Competence Center. I would like to take this opportunity to express my thanks to all the people involved, especially Dipl. Ing. Stefan Kirschbichler, Dr. Christoph Breitfuß and Dipl.-Ing. Julia Eichinger. Their generous support, effective cooperation and joint discussions have contributed significantly to the success of this work.

Special thanks go to Prof. Dr. Of for his valuable suggestions and careful support in the preparation of this Master's thesis. His generosity with his time deserves special thanks from me.

The unconditional support of my family has made my way through my studies and Master's thesis particularly valuable. I extend my sincere gratitude to my parents, Hamed and Mag. Amira, for their patience, loving guidance, and unwavering support throughout my entire academic journey. I also want to express my appreciation to my brothers, BSc. Amar and Bakir, for their motivational support.

I would also like to thank my friends Mag. Barbara Hierzer and Bsc. Marwin Strutz, who supported me throughout my studies as well as during the writing of my Master's thesis and always had time for me. Finally, I would like to thank my work colleagues and friends BSc. Alem Miralem and Dipl. Ing. Srdjan Tododrovic for their valuable support.

Abstract

The main objective of this master's thesis was to develop a reduced model to make short-circuit prediction for lithium-ion batteries based on failure criteria more efficient. Since the model was built in LS-DYNA and the finite element simulation was also performed there, we first dealt with the mathematical methods used by LS-DYNA. The central methodology of the construction of the reduced model was the application of the Empirical Interpolation Method. In the process of the research work, this method was specifically adapted to our problem and combined with other approaches in order to achieve the desired approximations and reductions of the data set. In particular we applied local and two-stage approximation methods. The subsequent evaluation of the failure criteria after the model reduction showed that we were able to achieve accurate predictions for a considerable number of scenarios.

Contents

Introduction	9
1 Description and modeling of lithium-ion batteries	13
1.1 Meso and macro model	15
1.2 Load cases	17
2 Mathematical description of the model	19
2.1 Weak formulation	20
2.2 Semidiscretization	21
2.3 Time-stepping	24
2.4 Elastic-plastic behavior	26
2.5 Modified Honeycomb Material	28
2.5.1 Stress Update	32
2.6 Volumetric strain as failure criterion	33
3 Empirical Interpolation Method	35
3.1 Basic Method	35
3.2 Implementation of the Empirical Interpolation Method	38
4 Model Order Reduction by the Empirical Interpolation Method	43
4.1 Numerical Model of the Load Cases	43
4.2 Application of the Empirical Interpolation Method	45
4.3 Basic Test Case	46
4.4 Approximation by the Empirical Interpolation Method	47
4.4.1 Comparing the Errors of Coarse and Fine Meshes	48
4.4.2 Low-Rank Approximation in the Entire Parameter Set	49
4.5 Local Approximation	51
4.5.1 Pure Splitting of the Parameter Set	52
4.5.2 Pure Splitting with Larger Margins	55
4.5.3 Two-Step Approximation at the Margins	57
4.6 Training on the Half Parameter Set	60
4.7 Translation in x - and z -direction	62
4.7.1 Global Approximation	63
4.7.2 Local Approximation	64
4.7.3 Refined Local Approximation at a Corner	67

4.7.4	Local Two-step Approximation at the Margins	69
4.8	Evaluation of the failure criteria	72
5	Conclusion	75
	Bibliography	77

Introduction

Batteries play a fundamental role in our modern lives, as they are used as energy storage devices in a variety of applications. These devices efficiently store and release electrical energy when needed. From small button cells in watches to large batteries in electric vehicles, it is impossible to imagine our everyday lives without batteries.

In recent years, e-mobility has gained importance as the number of electric vehicles on the road has increased. In light of this development, it is crucial to examine the safety aspects associated with the batteries used in electric vehicles more closely. Special attention is dedicated to mitigating risks associated with potential issues like electrical short circuits. To ensure the safety of electric vehicles, crash simulations and extensive testing are being carried out to gain a detailed understanding of what happens in the event of a short circuit.

The finite element method, in particular, has proven to be highly valuable simulation tool in the automotive sector. This method allows for highly accurate simulations of deformation processes ranging from quasi-static to highly dynamic. In addition, the cost and effort involved are usually significantly lower compared to real tests. This helps to increase vehicle safety in connection with electrical short circuits and other safety aspects. To predict the occurrence of a short circuit, we resort to different short circuit criteria based on strains and stresses. These criteria are used to identify potential break points or failure points in the system and to evaluate the safety of the structure. In this work, we specifically use a short circuit criterion based on the volumetric strain. This criterion allows us to predict the probability of a short circuit in the battery cell and develop appropriate safety measures. If the volumetric strain ε_{vol} exceeds the critical value ε_c somewhere we predict a short circuit. This criterion was established and validated in [33]. Since in our case the y -component of the strain has the largest values, we will use the strains in the y -direction as a good approximation to the volumetric strain, since the volumetric strain is the sum of the strains in the x , y and z directions.

Battery safety simulations are performed using various finite element solvers, including LS-DYNA, Abaqus, Nastran and others. Large models are often required to best represent the realistic situation. However, these large models can require long computation times and significant memory, making practical realization difficult. For this reason, various model reduction methods are developed and applied in many cases. These techniques make it possible to reduce the complexity of large models without significantly affecting the accuracy of the results.

Nowadays, machine learning and artificial intelligence are widely used in everyday

life as well as in science, although they are relatively new. The methods of machine learning are often used for model reduction. In the paper [12], sPGD (Sparse Proper Generalized Decomposition) and iDMD (Incremental Dynamic Model Decomposition) are specifically used as techniques for AI-based reduced models.

The most used battery models are macro and meso models, see [3]. In this work, the simulations and calculations focus on a macro model of a lithium-ion battery. Here, the battery cell is fixed on a plate and loaded with a sphere or a cylinder. These simulations are performed using the explicit finite element solver LS-DYNA. The simulation in LS-DYNA is performed by solving the Navier–Cauchy equation with the corresponding boundary and initial conditions, see [17, 18].

The strains and stresses computed during the simulation are of particular importance for the application of failure criteria. Some of these quantities can also serve as input parameters for model reduction methods. In using volumetric strains in our reduction methods, our goal was to reduce the model such that after the reduction we still find the elements in the model that represent failure in the system. This helps to increase the efficiency of our models while retaining the relevant physical properties that are important for a failure scenario analysis.

The Empirical Interpolation Method is a data-based model reduction method, see [14, 26]. The method extracts a set of basis functions from the available data that represent the most important variations in the system. These basis functions are used as the basis for the reduced model.

This thesis is divided into four chapters. In the first chapter, we provide a general description of a lithium-ion battery, with reference to sources such as [23]. In addition, we elaborated on the macro and meso models, using [3] as a source of information. Finally, potential loading scenarios of the battery are presented.

In the second chapter we deal with the mathematical description of the model. We consider an initial boundary value of the Navier-Cauchy equation and try to explain which mathematical methods are used by LS-DYNA in our finite element simulation. The time discretization is based on the modified central difference method. In Sections 2.1, the weak formulation of the Navier-Cauchy equation is derived with initial and boundary conditions suitable for our model. As a next step, we describe the semidiscretization of the weak formulation by a Finite Element Method in Section 2.2. As a result of the semidiscretization we get a system of equations which will be solved in LS-DYNA with modified central difference methods, see Section 2.3. Since we perform a crash simulation, we describe the elasto-plastic behavior of the material in Section 2.4. In our model we have to consider orthotropic material behavior, which will be model in LS-DYNA by Modified Honeycomb, which we described in Section 2.5. Finally in Section 2.6, we describe the considered failure criterion, motivated by [33].

The third chapter deals with the Empirical Interpolation Method [14, 26] and is mainly based on [29]. It consists of two parts: The first part deals with the theoretical background of this method, while the second part describes the implementation.

In the fourth chapter, we present our model reduction approach for our model and an

evaluation of the related approximations. In Section 4.1, the macro model used for the calculations is explained in detail. Our models represent load cases of the battery with a sphere and a cylinder. In Section 4.2, we described the application of the Empirical Interpolation method. Assuming that the strains in the y -direction remain relatively unchanged when the impactors are shifted, the strains for different parameters are overlaid by corresponding shifts. Then we apply the Empirical Interpolation Method to the overlaid strains. The subsequent back transformation ensures that the reduced approximation is transferred back to the original position. Due to the loss of data caused by the shift, we consider two approaches. In the first approach, the missing data is set to zero. The second approach involves completing the data by reflecting it with respect to the center of contact in each plane. In Section 4.3, we explain the parameterization of our model and an initial test setting, where only several positions of the impactor (sphere, cylinder) are considered. In Section 4.4, we present first test results. We first compare simulations of a coarse and a fine mesh and then investigated the low rank approximation for the entire parameter set. To obtain better results, we perform local approximations as described in Section 4.5. Such concepts are well known in model order reduction methods, see [2, Sec. 4.6] and [20, Sec. 4.3]. In addition to subdividing the parameter set for local approximation in the subsets, we apply a two-step approximation. In Section 4.6, we train our reduced model on half of the data and checked the quality of the approximation for the other half of the data. In Section 4.7, we consider a larger parameter set, varying the initial position of the impactor in the x - and z -directions. We apply the methods we developed for varying position in the z -direction here as well, with certain adjustments. In this data set, there is the case where the sphere exerts pressure at the corners, which is a new situation. For this reason, we extend our previous method to refined local approximation at the corners, see Section 4.7.3, to obtain more precise results. In Section 4.8, we evaluate the failure criterion from Section 2.6 on our two-dimensional data set. Lastly, in Chapter 5, we summarize the obtained results and present potential opportunities for improvement of the method.

1 Description and modeling of lithium-ion batteries

This chapter focuses on the lithium-ion battery as a crucial component of modern energy solutions. We start with a brief overview of the historical development and the basic structure of a lithium-ion battery. Then we discuss the description in more details by looking at macro and meso models of a lithium-ion battery. We will not only examine the structures of these models, but also explain the possible load cases.

We describe the lithium-ion battery with the help of information from [23], and [27]. A battery is a storage device for electrical energy. In general, it consists of a cell or a set of cells. The cells consist of two electrodes, an electrolyte, a separator and a cell frame. Alessandro Volta developed the first battery in 1800. He discovered that two different metals generate electricity when a weakly conducting, non-metallic medium connects them. More information about the history of the battery and electricity can be found in [27].

More than 100 years later, in the 1970s, the principle of the lithium-ion battery was developed. It first appeared on the market in a Sony Hi-8 video camera. Lithium-ion batteries have numerous applications, such as mobile phones, e-bikes and e-cars.

The lithium-ion batteries are the most popular choice for many applications due to their superior performance, see [10]. One of the great benefits of lithium-ion batteries is their energy efficiency, which allows them to store large amounts of energy in a compact and lightweight form. It is ideal for portable electronics and other devices where space is limited. Another advantage of lithium-ion batteries is their low self-discharge rate, which means they lose less energy when not in use compared to other types of rechargeable batteries. It is ideal for applications where batteries need to be stored for long periods without use. In addition, lithium-ion batteries do not have a "memory effect", which is a phenomenon where the battery's capacity is reduced if it is not fully discharged before being recharged. This is not an issue with lithium-ion batteries, which can be charged and discharged as often as needed without impacting their performance. Lithium-ion batteries are also known for their long life cycle, meaning they can be recharged and used many times before their capacity significantly decreases. This makes them a cost-effective choice for applications where the battery needs to be replaced less frequently. Finally, lithium-ion batteries are ideal for use over a wide temperature range, making them ideal for a variety of applications and environments. Lithium-ion batteries are designed to be efficient and reliable in hot and cold climates.

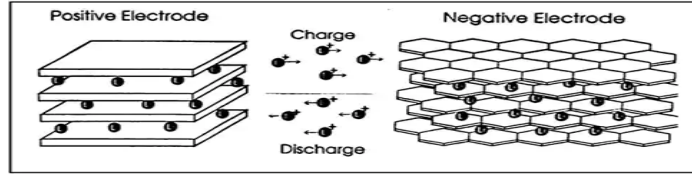


Figure 1.1: Charging and discharging processes of a lithium-ion battery, taken from [10]

Lithium-ion battery cells are built from two differently charged electrodes Fig. 1.1. During charging, the positively charged electrode is usually called the cathode, and the negatively charged electrode is called the anode. The electrodes are composed of a metal current conductor and an active material. The negative electrode consists of a copper foil and a layer of carbon or silicon compounds. The carbon compound used is natural or artificial graphite. It has a lower electrode potential and exhibits lower volume expansion during charging and discharging. During charging, lithium ions are reduced and stored in graphite layers. The positive electrode consists of mixed oxides deposited on an aluminum collector. The applied mixed oxides serve to store the lithium ions during discharge. The separator serves as a physical separation of the electrodes while allowing the flow of electrons, see Fig. 1.2. This prevents a short circuit caused by direct contact between the anode and cathode. Polyethylen, polypropylen and ceramic are usually used materials for building a separator. In addition, between the electrodes is the ion-conducting electrolyte, which enables the exchange of ions. There are three types of electrolytes: liquid, polymer and solid. In a commercial Lithium-ion battery, the liquid electrolyte is used. During discharge, the Lithium ions migrate from the negatively charged electrodes through the separator and electrolyte to the positively charged electrodes. The oxidation process releases the electrons. They then flow from the anode to the cathode in the external electric circuit. A reduction process occurs at the cathode, enabling electrons to be absorbed. When charging, precisely the opposite process will happen. The charging and discharging process is

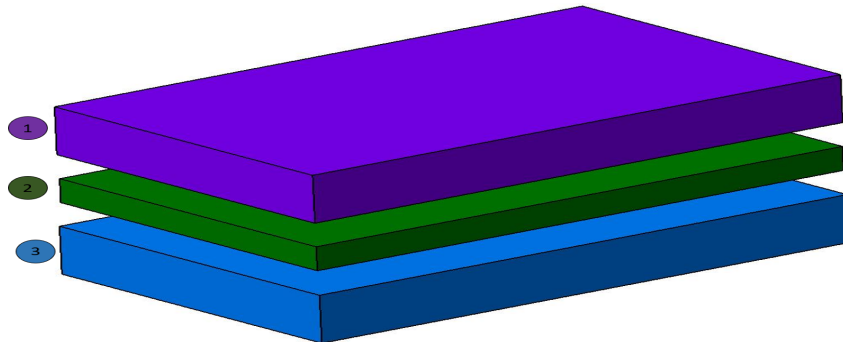


Figure 1.2: Structure of a lithium-ion battery; 1. anode, 2. separator, 3. cathode

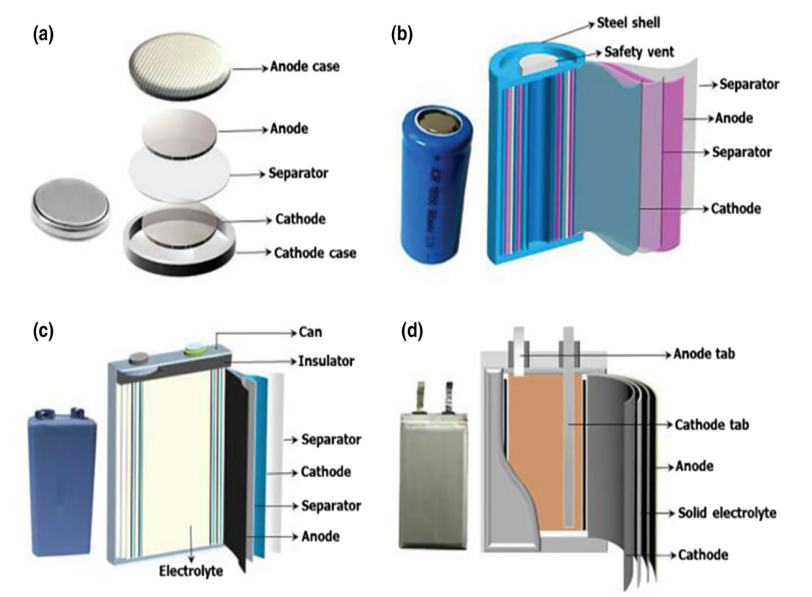


Figure 1.3: Types of the lithium-ion battery (a) coin-type, (b) cylindrical-type, (c) prismatic-type, (d) pouch-type , taken from [7]

shown in Fig. 1.1 shown and the construction of different Lithium-ion batteries in Figure 1.3. More information about the structure and production process, as well as the properties of a Lithium-ion battery, can be found in [11] and [19].

1.1 Meso and macro model

In order to increase the safety of a vehicle, we would like to check the mechanical behaviour of a battery under different loads in numerical simulation. For this purpose, various models are created for the software such as LS-DYNA. Generally, meso and macro models are used to understand the phenomena by looking at processes at different levels.

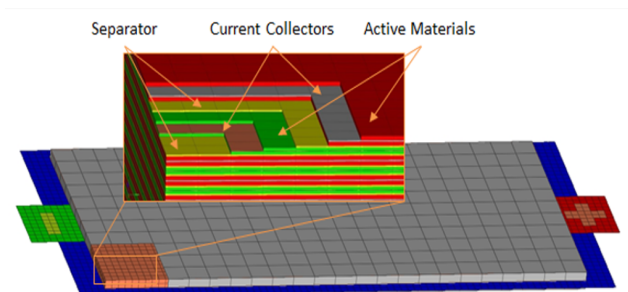


Figure 1.4: Meso model; taken from [12]

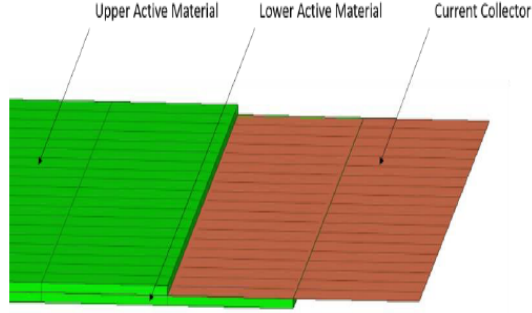


Figure 1.5: Meso model; taken from [12]

The meso model is used for detailed analysis to understand the phenomena. On the other hand, a macro model is used to predict short circuits using a short circuit criterion in a larger model (e.g. vehicle). Short circuit criteria are based on different physical quantities, mostly stress and strain. For this section, [3] and [12] are used as a sources.

In a meso model (Fig. 1.4), we model the following parts separately: electrodes, separator, pouch foil, tabs and electrolytes. The electrodes are built in three parts, see Fig. 1.5. In the middle, there is a current metal collector. It is coated on both sides with active materials. The current collector is made of copper for the anode and aluminum for the cathode. The negative and positive tabs (see Fig. 1.4) are made of the same materials, i.e. the positive is made of aluminum and the negative of copper. For the mechanical modeling, an elastoplastic material model is used.

A separator separates the electrodes. The separator has an orthotropic mechanical behaviour. Therefore, it is modeled in LS-DYNA with material type 126 Modified Honeycomb. We will deal with the material in more detail in Section 2.4. The electrolytes are modeled by adding mass to the nodes of the separator. The reason for this is that the electrolyte is located in the pores of the separator.

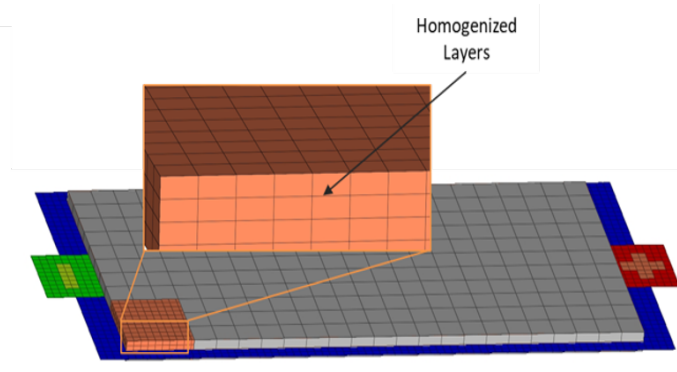


Figure 1.6: Cross section of the macro model; taken from [12]

In our macro model (Fig. 1.6), the mechanical behaviour of all individual layers (separator and electrodes) is combined to form a homogeneous material. We refer this part as jellyroll, and it is modeled in the same way as a separator in our meso model. Since the separator no longer exists as a separate part, the electrolytes are considered by adjusting the density of the jellyroll, and the material parameters are modified. The remaining parts of the model are taken from the meso model.

For both models approximate solutions are computed by LS-DYNA, where LS-DYNA is a finite element solver. In Tables 1.1a and 1.1b more detailed information about the models is given. The characteristic length of our models is given as the element volume divided by the surface of the largest side of the element, see [17, Ch. 22]. Furthermore, we see that the number of elements in the meso model is much larger than in the macro model. Therefore, the simulation with a macro model is much faster.

Name	Variable
Char.length	0.015 mm
Nodes	23707397
Shells	822610
Solid	12131168
Shell Parts	72
Solid Parts	190

(a) Data for the meso model

Name	Variable
Char.length	1.17 mm
Nodes	59228
Shells	16012
Solid	37128
Shell Parts	8
Solid Parts	1

(b) Data for the macro model

Table 1.1: Data for meso and macro model

1.2 Load cases

In this section, we discuss possible load cases of a battery. There are several load cases for a battery but the following three fit best for our situation. Figure 1.7 shows the 3-point bending, where the blue cylinders represent the support, the red cylinder represent the impactor, and the green body represents the battery. A 3-point-bending is suitable for describing a bending of cells. This is an essential feature of the car model. In this case, a short circuit should not be expected.

In contrast, the cylindrical (Fig. 1.9) and spherical (Fig. 1.8) indentation are likely to cause a short circuit. In Figures 1.9 and 1.8, the blue part represents a plate, and the green part represents the battery. The battery is fixed on the plate and is pressed with an impactor. The impactor is the red cylinder in Fig. 1.9 and the red sphere in Fig. 1.8. The contact area between the sphere and the battery is smaller than

between the cylinder and the battery. Therefore, with the same force greater pressure is generated at the spherical indentation.

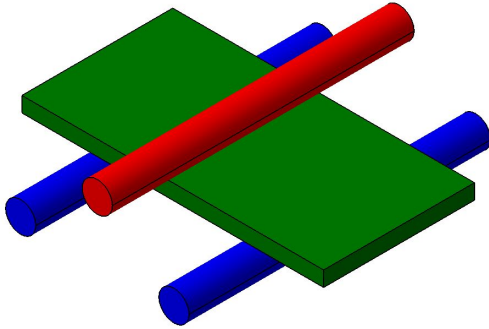


Figure 1.7: Load case; 3-point-bending

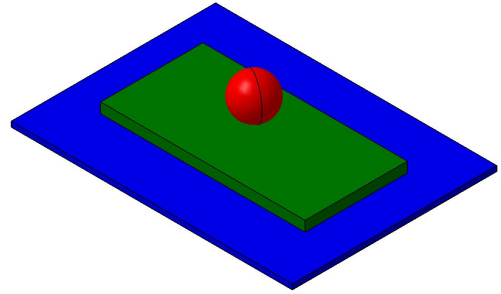


Figure 1.8: Load case; Spherical indentation

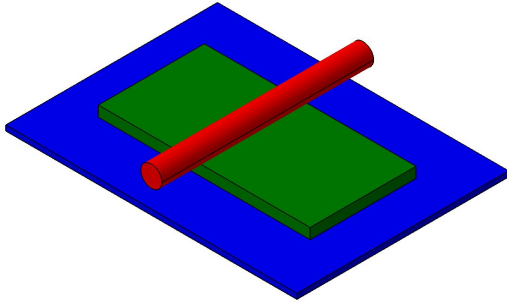


Figure 1.9: Load case; Cylindrical indentation

2 Mathematical description of the model

In this chapter, we will mathematically describe the mechanical processes in a battery cell. Like many other processes, this one is described with a partial differential equation. The mathematical derivations are based on ideas from [22] and [30]. Before we give the partial differential equation, we establish the notation.

Let $\Omega \subset \mathbb{R}^3$ be a bounded domain with the boundary $\Gamma = \bar{\Gamma}_D \cup \bar{\Gamma}_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Further on, Γ_D denotes the Dirichlet and Γ_N the Neumann boundary. Let $\underline{x} \in \mathbb{R}^3$ and time $t \in \mathbb{R}_+$. As we mentioned in Chapter 1, we are interested in the mechanical processes in a battery cell under load. With \underline{f} we will denote the volume forces which act on the battery and cause some displacement $\underline{u}(\underline{x}, t)$ and ρ describes the mass density. The stresses $\underline{\sigma} \in \mathbb{R}^{3 \times 3}$ are induced by the displacements. At first, we assumed a linear stress-strain relation, e.g. Hook's law. In Section 2.5, we will enhance the model by some nonlinear stress-strain relation. Since we describe a dynamic process, we will also have to take into account the acceleration $\underline{\ddot{u}}(\underline{x}, t)$. Furthermore, γ_0^{int} denotes the inner Dirichlet trace, and \underline{n} is the exterior unit normal vector. We consider the Navier–Cauchy equation, which can be found in [17, Ch.2, p.2.1],

$$\rho \underline{\ddot{u}}(\underline{x}, t) - \operatorname{div}(\underline{\sigma}(\underline{u}(\underline{x}, t), \underline{x})) = \rho \underline{f}(\underline{x}, t) \quad \text{for all } (\underline{x}, t) \in \Omega \times (0, T). \quad (2.1)$$

In our case, the boundary conditions are as follows

- the displacement boundary condition

$$\gamma_0^{int} \underline{u}(\underline{x}, t) = \underline{g}_D(\underline{x}, t) \quad \text{for all } (\underline{x}, t) \in \Gamma_D \times (0, T),$$

- the traction boundary conditions

$$(\underline{\sigma}(\underline{u}(\underline{x}, t), \underline{x})) \underline{n}(\underline{x}) = \underline{g}_N(\underline{x}, t) \quad \text{for all } (\underline{x}, t) \in \Gamma_N \times (0, T).$$

Our partial differential equation also requires initial conditions, e.g.

$$\begin{aligned} \underline{u}(\underline{x}, 0) &= 0, \\ \underline{\dot{u}}(\underline{x}, 0) &= 0 \end{aligned}$$

for all $\underline{x} \in \Omega$. In Section 2.3 we will add Rayleigh damping to the model.

2.1 Weak formulation

Only a few partial differential equations can be solved analytically. Many others can be solved with the help of various numerical methods. The solution of our partial differential equation (2.1) can also only be found numerically. For the numerical solving of the equation (2.1) we will use the finite element method. For that we need a weak formulation, which will be developed in the spirit of [16] and [30]. Next, we mention the spaces and functions that we will need for the weak formulation.

Let $\underline{u} \in L^2(0, T; H^3)$, where $H = H^1(\Omega)$. Moreover, the second time derivative $\ddot{\underline{u}}$ of the displacement \underline{u} has to be in $L^2(0, T; [H^*]^3)$, with the dual space H^* of H . We require for the Dirichlet data that $\underline{g}_D \in [H^{1/2}(\Gamma_D)]^3$ and for Neumann data $\underline{g}_N \in [H^{-1/2}(\Gamma_N)]^3$. Let further the space of test functions be defined as the space of all $[H^1(\Omega)]^3$ functions that vanish on the Dirichlet boundary Γ_D , formally written as

$$V = \{\underline{v} \in [H^1(\Omega)]^3 : \underline{v} = 0 \text{ on } \Gamma_D\}.$$

Lastly, we require the volume forces $\underline{f} \in [L^2(\Omega)]^3$. Now, we multiply the equation (2.1) by a test function $\underline{v} \in V$ and integrate over the domain Ω . Then we get

$$\int_{\Omega} \rho \ddot{\underline{u}}(\underline{x}, t) \cdot \underline{v}(\underline{x}) d\underline{x} - \int_{\Omega} \operatorname{div}(\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x})) \cdot \underline{v}(\underline{x}) d\underline{x} = \int_{\Omega} \rho \underline{f}(\underline{x}, t) \cdot \underline{v}(\underline{x}) d\underline{x}. \quad (2.2)$$

Further, we apply integration by parts to the second integral and get

$$\begin{aligned} - \int_{\Omega} \operatorname{div}(\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x})) \cdot \underline{v}(\underline{x}) d\underline{x} &= - \int_{\Gamma} (\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) \underline{n}(\underline{x})) \cdot \underline{v}(\underline{x}) dS_x \\ &\quad + \int_{\Omega} \boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) : \nabla \underline{v}(\underline{x}) d\underline{x}, \end{aligned}$$

where

$$\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) : \nabla \underline{v}(\underline{x}) = \sum_{i,j=1}^3 \sigma_{ij}(\underline{u}(\underline{x}, t), \underline{x}) \frac{\partial}{\partial x_j} v_i(\underline{x}).$$

The Neumann boundary condition provides

$$\int_{\Gamma} (\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) \underline{n}(\underline{x})) \cdot \underline{v}(\underline{x}) dS_x = \int_{\Gamma_N} \underline{g}_N(\underline{x}, t) \cdot \underline{v}(\underline{x}) dS_x$$

for all \underline{v} in V . We plug these transformations into (2.2) and get

$$\begin{aligned} &\int_{\Omega} \rho \ddot{\underline{u}}(\underline{x}, t) \cdot \underline{v}(\underline{x}) d\underline{x} + \int_{\Omega} \boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) : \nabla \underline{v}(\underline{x}) d\underline{x} \\ &= \int_{\Omega} \rho \underline{f}(\underline{x}, t) \cdot \underline{v}(\underline{x}) d\underline{x} + \int_{\Gamma_N} \underline{g}_N(\underline{x}, t) \cdot \underline{v}(\underline{x}) dS_x \end{aligned}$$

for all $v \in V$. The weak formulation read as follows:

Find $\underline{u} \in L^2(0, T; H^3)$ with $\ddot{\underline{u}} \in L^2(0, T; [H^*]^3)$ and $\underline{u}(\underline{x}, t) = \underline{g}_D(\underline{x}, t)$ for $\underline{x} \in \Gamma_D$ such that

$$b(\ddot{\underline{u}}, \underline{v}) + a(\underline{u}(t), \underline{v}) = F(\underline{v}, t) \quad \text{for all } \underline{v} \in V \quad (2.3)$$

holds, with

$$b(\underline{w}, \underline{v}) = \int_{\Omega} \rho \underline{w}(\underline{x}) \cdot \underline{v}(\underline{x}) d\mathbf{x},$$

$$a(\underline{u}, \underline{v}) = \int_{\Omega} \boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x}) : \nabla \underline{v}(\underline{x}) d\mathbf{x},$$

$$F(\underline{v}, t) = \int_{\Omega} \rho \underline{f}(\underline{x}, t) \cdot \underline{v}(\underline{x}) d\mathbf{x} + \int_{\Gamma_N} \underline{g}_N(\underline{x}, t) \cdot \underline{v}(\underline{x}) dS_x.$$

In case of Hook's law the bilinear form $a(\underline{u}, \underline{v})$ can be transformed into a symmetric representation [30, Ch.1, p.15-19].

2.2 Semidiscretization

In this section we discuss a discrete finite element method (FEM) approximation of the weak formulation (2.3). About finite element discretization one can find more information in ,e.g., [16, 22, 30].

For simplicity, we discuss the case that the Dirichlet boundary conditions are zero, i.e. $\underline{g}_D = 0$. LS-DYNA imposes such boundary conditions within its time discretization scheme [17, Ch. 23]. In our discussion we focus on the separator, which are discretized by solid elements. Note that shell elements are use for current collectors, pouch foil and tabs. The domain $\overline{\Omega} = \cup_{k=1}^N \overline{\tau}_k$ is splitted into elements τ_k . In our case the finite elements τ_k are 8-node hexahedron elements, which look like the element in Figure 2.1. The trilinear shape function ϕ_j is given for $j = 1, \dots, 8$ on the reference element as

$$\phi_j(\xi_1, \xi_2, \xi_3) = \frac{1}{8} (1 + \xi_1 \xi_1^j) (1 + \xi_2 \xi_2^j) (1 + \xi_3 \xi_3^j) \quad \text{for } (\xi_1, \xi_2, \xi_3) \in [-1, 1]^3$$

where $\xi_1^j, \xi_2^j, \xi_3^j$ represent the coordinates of j -th vertex of the element and take values the $(\pm 1, \pm 1, \pm 1)$, see [17, Ch.3, p.3.1-3.3]. Let V_h be the trial space given by $V_h = [Q_h^1(\Omega)]^3 \cap V$, where Q_h^1 is the space of the locally trilinear and globally continuous functions. In the following $3M$ is the number of nodes in V_h . Let $\{\varphi_\ell\}_{\ell=1}^{3M}$ be the basis of V_h with

$$\varphi_i(\underline{x}) = \begin{pmatrix} \tilde{\varphi}_i(x) \\ 0 \\ 0 \end{pmatrix}, \quad \varphi_{M+i}(\underline{x}) = \begin{pmatrix} 0 \\ \tilde{\varphi}_i(x) \\ 0 \end{pmatrix}, \quad \varphi_{2M+i}(\underline{x}) = \begin{pmatrix} 0 \\ 0 \\ \tilde{\varphi}_i(x) \end{pmatrix}$$

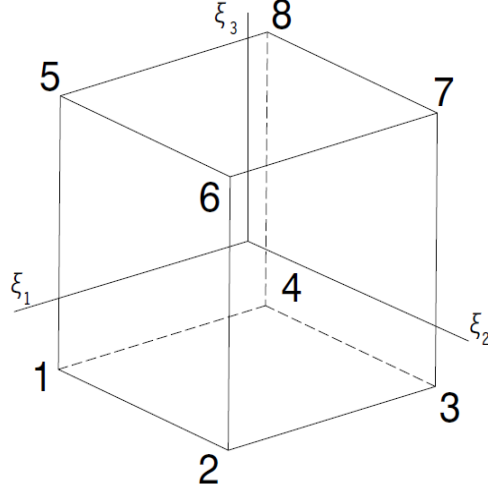


Figure 2.1: Eight node hexahedron element

for $i = 1, \dots, M$ and for $x \in \tau_k$

$$\tilde{\varphi}_i(\underline{x}) := \phi_j(\underline{\xi}),$$

for some j , where $\underline{x} = J(\underline{\xi})$ with the trilinear mapping J from the reference element $[-1, 1]^3$ to the element τ_k and the vectors $\underline{\xi} \in [-1, 1]^3$. More information can be found in [17, Ch.3, p. 3.1-3.3]. In the books [16, 22] you can read more about the hexahedral elements.

The related Galerkin formulation reads as:

Find $\underline{u}_h(t) \in V_h$ such that

$$b(\ddot{\underline{u}}_h(t), \underline{v}_h) + a(\underline{u}_h(t), \underline{v}_h) = F(\underline{v}_h, t) \quad \text{for all } \underline{v}_h \in V_h. \quad (2.4)$$

We represent \underline{u}_h as:

$$\underline{u}_h(\underline{x}, t) = \sum_{\ell=1}^{3M} u_\ell(t) \varphi_\ell(\underline{x}),$$

and observe

$$\ddot{\underline{u}}_h(\underline{x}, t) = \sum_{\ell=1}^{3M} \ddot{u}_\ell(t) \varphi_\ell(\underline{x}).$$

We plug $\ddot{\underline{u}}_h, \underline{u}_h$ and

$$\underline{v}_h(\underline{x}) = \varphi_k(\underline{x}) \quad \text{for } k = 1, \dots, 3M$$

into (2.4) and derive the equivalent system of linear equation:

The coefficients of the right-hand side are

$$f_k(t) = F(\underline{\varphi}_k, t) = \int_{\Omega} \rho \underline{f}(\underline{x}, t) \cdot \underline{\varphi}_k(\underline{x}) d\underline{x} + \int_{\Gamma_N} \underline{g}_N(\underline{x}, t) \cdot \underline{\varphi}_k(\underline{x}) dS_x, \quad k = 1, \dots, 3M.$$

Since $\boldsymbol{\sigma}(\underline{u}(\underline{x}, t), \underline{x})$ is linear, the bilinear form $a(\underline{u}_h, \underline{v}_h)$ looks like

$$a(\underline{u}_h, \underline{\varphi}_k) = \sum_{\ell=1}^{3M} u_{\ell}(t) \int_{\Omega} \boldsymbol{\sigma}(\underline{\varphi}_{\ell}(\underline{x}), \underline{x}) : \nabla \underline{\varphi}_k(\underline{x}) d\underline{x} = (K_h \underline{u})_k, \quad k = 1, \dots, 3M$$

with the entries of the stiffness matrix

$$K_h[k, \ell] = \int_{\Omega} \boldsymbol{\sigma}(\underline{\varphi}_{\ell}(\underline{x}), \underline{x}) : \nabla \underline{\varphi}_k(\underline{x}) d\underline{x}, \quad k, \ell = 1, \dots, 3M$$

and the vector

$$\underline{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_{3M} \end{pmatrix}.$$

Substituting $\underline{w}_h(\underline{x}, t) = \sum_{\ell=1}^{3M} w_{\ell}(t) \underline{\varphi}_{\ell}(\underline{x})$ and $\underline{v}_h(\underline{x}) = \underline{\varphi}_k(\underline{x})$ into $b(\underline{w}, \underline{v})$ yields

$$b(\underline{w}_h, \underline{\varphi}_k) = \sum_{\ell=1}^{3M} w_{\ell}(t) \int_{\Omega} \rho \underline{\varphi}_{\ell}(\underline{x}) \cdot \underline{\varphi}_k(\underline{x}) d\underline{x} = (M_h \underline{w})_k, \quad k = 1, \dots, 3M,$$

where the mass matrix is given as

$$M_h[k, \ell] = \int_{\Omega} \rho \underline{\varphi}_{\ell}(\underline{x}) \cdot \underline{\varphi}_k(\underline{x}) d\underline{x}, \quad k, \ell = 1, \dots, 3M.$$

Thus, the Galerkin weak formulation (2.4) is equivalent to the system of linear ordinary differential equations

$$M_h \ddot{\underline{u}}(t) + K_h \underline{u}(t) = \underline{f}(t)$$

with the following initial conditions

$$\begin{aligned} \underline{u}(0) &= 0, \\ \dot{\underline{u}}(0) &= 0. \end{aligned}$$

In simulations, some Hourglass deformations of hexahedral elements may occur. To prevent this, LS-DYNA uses the Hourglass resistance (See [17, Ch. 24]).

2.3 Time-stepping

In general, oscillating systems have damping. We will look at the Rayleigh damping, see [9]. In this case, the damping matrix is represented as a linear combination of the mass and stiffness matrix:

$$C_h = \alpha M_h + \beta K_h.$$

Here α and β are the Rayleigh coefficients proportional to the mass and stiffness, respectively. If we consider the damping matrix in our system, we have the following system of ordinary differential equations

$$M_h \ddot{\underline{u}}(t) + C_h \dot{\underline{u}}(t) + K_h \underline{u}(t) = \underline{f}(t), \quad (2.5)$$

with the initial conditions

$$\begin{aligned} \underline{u}(t_0) &= 0, \\ \dot{\underline{u}}(t_0) &= 0. \end{aligned}$$

With \underline{u} the displacement is described. The first time derivative of the displacement is the velocity $\dot{\underline{u}}$ and the derivative of the velocity is the acceleration $\ddot{\underline{u}}$. To solve the system, we will use a central differences method. More about the method can be read in [22, Ch. 7]. The version of the central difference method, that we describe, can be found in [17, Ch. 24], [25]. The central difference method is a numerical method for solving ordinary and partial differential equations. The idea of the method is to approximate the derivatives in the differential equation by central differential quotients. In LS-DYNA different problems are solved with different variants of the central difference method.

In the linear case and for fixed time step size Δt the velocity $\dot{\underline{u}}$ and the acceleration $\ddot{\underline{u}}$ at time t_n are approximated in the following way:

$$\dot{\underline{u}}(t_n) \approx \frac{1}{2\Delta t} (\underline{u}^{(n+1)} - \underline{u}^{(n-1)}), \quad (2.6)$$

$$\ddot{\underline{u}}(t_n) \approx \frac{1}{\Delta t^2} (\underline{u}^{(n+1)} - 2\underline{u}^{(n)} + \underline{u}^{(n-1)}), \quad (2.7)$$

where $\underline{u}^{(n)}$ is an approximation of \underline{u} at the time t_n . Furthermore, $\underline{u}^{(n+1)}$ and $\underline{u}^{(n-1)}$ denote \underline{u} at time t_{n+1} and t_{n-1} , where t_{n+1} and t_{n-1} are defined as $t_n + \Delta t$ and $t_n - \Delta t$. The time step will be computed as

$$\Delta t = \frac{\ell}{c},$$

where ℓ is a characteristic length of an element and c is a sound speed of the material, i.e.

$$c \sim \sqrt{\frac{E}{\rho}},$$

with the elastic modulus E and the density ρ , see [25]. Further, the approximations (2.6) and (2.7) are plugged into the equation (2.5) at time t_n and after some transformation, we get:

$$\left(\frac{M_h}{\Delta t^2} + \frac{C_h}{2\Delta t}\right) \underline{u}^{(n+1)} + \left(K_h - 2\frac{M_h}{\Delta t^2}\right) \underline{u}^{(n)} + \left(\frac{M_h}{\Delta t^2} - \frac{C_h}{2\Delta t}\right) \underline{u}^{(n-1)} = \underline{f}(t_n).$$

In the next step we obtain a system of linear equations to find $\underline{u}^{(n+1)}$:

$$\widehat{M} \underline{u}^{(n+1)} = \hat{f} \quad (2.8)$$

where

$$\begin{aligned} \widehat{M} &= \frac{M_h}{\Delta t^2} + \frac{C_h}{2\Delta t}, \\ \hat{f} &= \underline{f}(t_n) - \left(K_h - 2\frac{M_h}{\Delta t^2}\right) \underline{u}^{(n)} - \left(\frac{M_h}{\Delta t^2} - \frac{C_h}{2\Delta t}\right) \underline{u}^{(n-1)}. \end{aligned}$$

In order to solve the system of equations (2.8) more easily, the matrices M_h and C_h are diagonalized by LS-DYNA, see [25]. Furthermore, we need a representation for $\underline{u}^{(-1)}$. With the help of the Taylor expansion we get

$$\underline{u}(-\Delta t) \approx \underline{u}(0) - \Delta t \underline{\dot{u}}(0) + \frac{(\Delta t)^2}{2} \underline{\ddot{u}}(0) \approx \underline{u}(0) - \Delta t \underline{\dot{u}}(0) + \frac{(\Delta t)^2}{2} \underline{a}^{(0)} =: \underline{u}^{(-1)},$$

where $\underline{a}^{(0)}$ is computed from the equilibrium state (2.5) at $t = 0$, i.e.

$$\underline{a}^{(0)} = M_h^{-1}(\underline{f} - C_h \underline{\dot{u}}(0) - K_h \underline{u}(0)).$$

This method is stable for sufficiently small time step. A relation discussion can be found in [17, Ch.24, p. 24.4-24.5] and [25].

In the case of non-linearity (see [25]), the following system of equations is solved:

$$M_h \underline{\ddot{u}}(t) + C_h \underline{\dot{u}}(t) + \underline{K}(\underline{u}(t)) = \underline{f}(t). \quad (2.9)$$

Contrary to the linear case (2.5), \underline{K} is not a matrix but a function that depends non-linearly on displacement \underline{u} . A modified central difference method with varying time steps size on staggered grids is used for this purpose, [25]. Here a different notation is

used $t_{n+1/2} = t_{n-1/2} + \Delta t_n$ and $t_{n+1} = t_n + \Delta t_{n+1/2}$, where $\Delta t_{n+1/2} = \frac{\Delta t_{n+1} + \Delta t_n}{2}$. The velocity $\underline{\dot{u}}$ is approximated at time $t_{n+1/2}$:

$$\underline{\dot{u}}(t_{n+1/2}) \approx \frac{1}{\Delta t_{n+1/2}}(\underline{u}^{(n+1)} - \underline{u}^{(n)}) =: \underline{v}^{(n+1/2)}. \quad (2.10)$$

Furthermore, the acceleration $\underline{\ddot{u}}$ at time t_n is approximated by a central difference quotient

$$\underline{\ddot{u}}(t_n) \approx \frac{1}{\Delta t_n}(\underline{\dot{u}}(t_{n+1/2}) - \underline{\dot{u}}(t_{n-1/2})) \approx \frac{1}{\Delta t_n}(\underline{v}^{(n+1/2)} - \underline{v}^{(n-1/2)}) =: \underline{a}^{(n)}. \quad (2.11)$$

Since $\underline{\dot{u}}(t_n)$ is unknown, we replace it in the equation (2.9) with $\underline{\dot{u}}(t_{n-1/2})$ i.e.

$$\underline{\ddot{u}}(t_n) \approx M_h^{-1}(f(t_n) - C_h \underline{\dot{u}}(t_{n-1/2}) - \underline{K}(\underline{u}(t_n))). \quad (2.12)$$

Using the approximation $\underline{v}^{n-1/2}$ of $\underline{\dot{u}}(t_{n-1/2})$ we compute an approximation $\underline{a}^{(n)}$ of $\underline{\ddot{u}}(t_n)$ from

$$\underline{a}^{(n)} = M_h^{-1}(f(t_n) - C_h \underline{v}^{(n-1/2)} - \underline{K}(\underline{u}^{(n)})).$$

From (2.11) we get the update for the velocity

$$\underline{v}^{(n+1/2)} = \underline{v}^{(n-1/2)} + \Delta t_n \underline{a}^{(n)}$$

and from (2.10) for the displacement

$$\underline{u}^{(n+1)} = \underline{u}^{(n)} + \Delta t_{n+1/2} \underline{v}^{(n+1/2)}.$$

For $n = 0$ the first values are computed as

$$\begin{aligned} \underline{u}^{(1)} &= \underline{u}^{(0)} + \Delta t_{1/2} \underline{v}^{(1/2)} \\ \underline{v}^{(1/2)} &= \underline{v}^{(0)} + \frac{\Delta t_0}{2} \underline{a}^{(0)}. \end{aligned}$$

In [25], one finds more about the discretization and the critical time step sizes for varying time increment. Note that LS-DYNA uses derivatives of the actual position \underline{x} instead of the displacement \underline{u} .

2.4 Elastic-plastic behavior

In this section we describe the material behaviour during a deformation. The description is based on the information from the publications [15, 8, 28]. The deformation of the body consists of two parts, elastic and plastic. In materials science, the term elasticity (reversible deformation) refers to the property of a material to return to

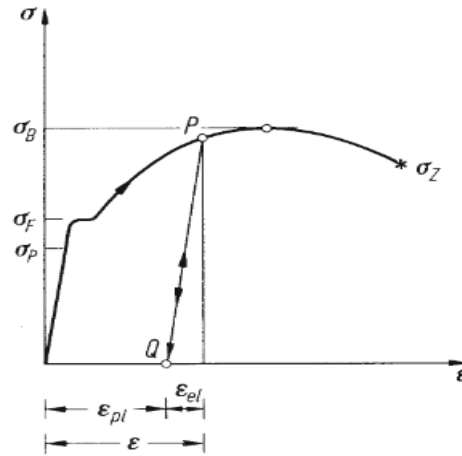


Figure 2.2: Load curve; [15, Ch. 6, p. 353]

its original shape after removing the forces. In contrast, plasticity (irreversible deformation) means that a residual deformation remains after the forces are removed. One speaks of elasto-plastic material behaviour when the material behaves elastically up to a certain point of the stress-strain curve (yield point) and deforms plastically above this point. Figure 2.2 shows the stress-strain curve for a material that behaves linearly elastically up to the yield point. The slope of this straight line is called the modulus of elasticity E . The curve is linear up to the proportionality limit σ_P . If the load increases, the relationship between σ and ε becomes non-linear. The material behaviour is still elastic (for $\sigma < \sigma_F$), i.e. at complete unloading, the strain along the load curve decreases to zero. It should be noted that the non-linear elastic range for many metals is really small and often negligible. After the yield stress σ_F is reached, plastic deformation sets in. Further, the stress will increase until the maximum stress σ_B is reached. The strain can increase while the stress remains practically constant. Subsequently, additional elongation requires an increase in load. When unloading after reaching any point P (Fig. 2.2) in the plastic range, the unloading curve is parallel to the straight line in the linear-elastic region, and a permanent (plastic) strain ε_{pl} remains. Reloading from point Q takes place along the straight line QP (A thin loop of hysteresis occurs, but this is often neglected.). Plastic flow then only begins again at point P . The stress at point P can therefore be regarded as a new yield stress for the already plastically deformed material. According to Fig. 2.2, the elongation at the point P is the sum of the elastic strain ε_{el} and the plastic strain ε_{pl} :

$$\varepsilon = \varepsilon_{el} + \varepsilon_{pl}.$$

2.5 Modified Honeycomb Material

In this section we deal with the modified Honeycomb material, which is described in [18, Ch.2, p.2-796 - 2-807] and [17, Ch.19, p.19.149-19.150]. The material is orthotropic, which means that in the reference frame parallel to the orthotropy axes there is no coupling between normal strains and shear strain. Therefore, we can say that a body of such material deforms differently in all directions if it is under some load. Many structural materials are orthotropic, e.g. engineered wood, woven fabrics, many fiber-plastic composites and textured rolled sheets. More information about orthotropic materials can be found in [1] and [21].

The modified Honeycomb material model in LS-DYNA is usually referred to as MAT_126. The further text is based on the descriptions from [18, Ch.2, p.2-796 - 2-807] and [17, Ch.19, p.19.149 - 19.150].

First we will deal with the coordinate system in LS-DYNA. In LS-DYNA, the Cartesian coordinate system (x, y, z) is set as the general coordinate system. MAT_126 has the property that it can deform differently in different directions. In order to describe the load cases in different directions, we define a local coordinate system with the directions $\vec{a}, \vec{b}, \vec{c}$. In Figure 2.3 you can see how the coordinate systems look like in our case.

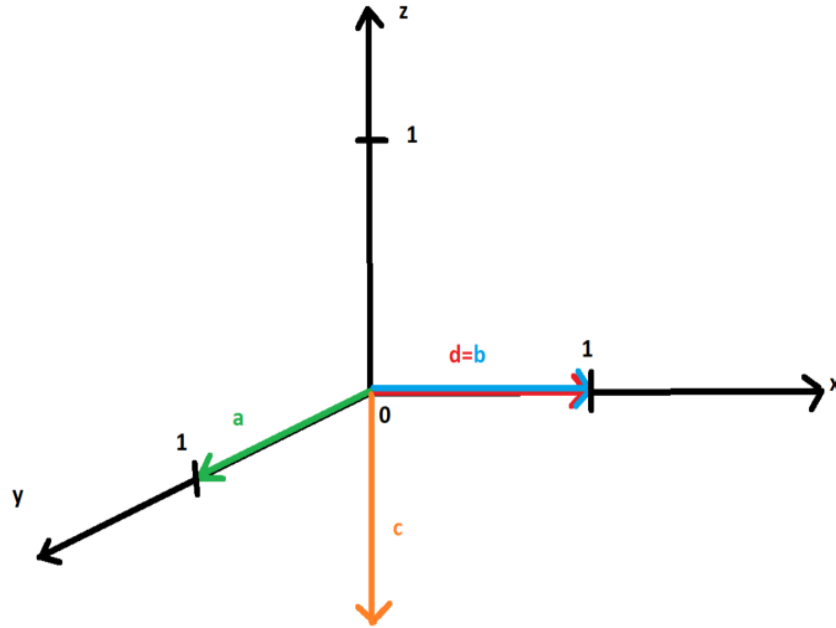


Figure 2.3: Local and global coordinate system

We compute two vectors \vec{c} and \vec{b} by choosing two vectors \vec{a} and \vec{d}

$$\vec{c} = \vec{a} \times \vec{d}, \quad \vec{b} = \vec{c} \times \vec{a}.$$

In our model

$$\vec{a} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

are chosen. Then

$$\vec{c} = \vec{a} \times \vec{d} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \quad \vec{b} = \vec{c} \times \vec{a} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

In addition we will use the material card for the model in order to enhance the description from the Chapters 1 and 2. In the material card, Figure 2.4, we specify the properties of the material through the parameters according to the standard in [18, Ch. 2, p. 2.796 - 2.807]. Only parts of the material card relevant for our model will be described. The description of the terms from the material card are given in Table 2.1.

```
*MAT_MODIFIED_HONEYCOMB_TITLE
Separator
$# mid ro e pr sigy vf mu bulk
    5 5.02E-6 2.62312 0.45 6.05E-5 0.0 0.02 0.0
$# lca lcb lcc lcs lcab lcba lcca lcsr
   5020 5040 5010 0 0 0 0 0
$# eaau ebbu eccu gabu gbcu gcau aopt macf
  2.62312 2.62312 2.25237 0.7659 0.7659 0.7659 2.0 1
$# xp yp zp a1 a2 a3
    0.0 0.0 0.0 0.0 1.0 0.0
$# d1 d2 d3 tsef ssef vref tref shdflg
    1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

Figure 2.4: Material card; Copyright: Virtual Vehicle

Card Variable	Variable	Definition of Variables
ro	ρ	Material density
e	E	Young's modulus for compacted honeycomb material
pr	ν	Poisson's ratio of compacted honeycomb material
sigy		Yield stress of fully compacted honeycomb material
mu	μ	Material viscosity coefficient
lca, lcb, lcc		Load curve of stress $\sigma_{aa}, \sigma_{bb}, \sigma_{cc}$ as function of normal strain $\varepsilon_{aa}, \varepsilon_{bb}, \varepsilon_{cc}$
eaau, ebbu, eccu	E_{iiu}	Modulus of elasticity in $a-, b-, c-$ direction, respectively, in uncompressed configuration
gab, gbcu, gcua	G_{iju}	Shear modulus in $ab-, bc-, ca-$ direction, respectively, in uncompressed configuration
aopt=2		Globally orthotropic with material axes determined by vectors \vec{a}, \vec{d}
a_1, a_2, a_3		The components of vector \vec{a}
d_1, d_2, d_3		The components of vector \vec{d}

Table 2.1: Description of relevant parameters on the material card in Figure 2.4, based on [18, p.796-809]

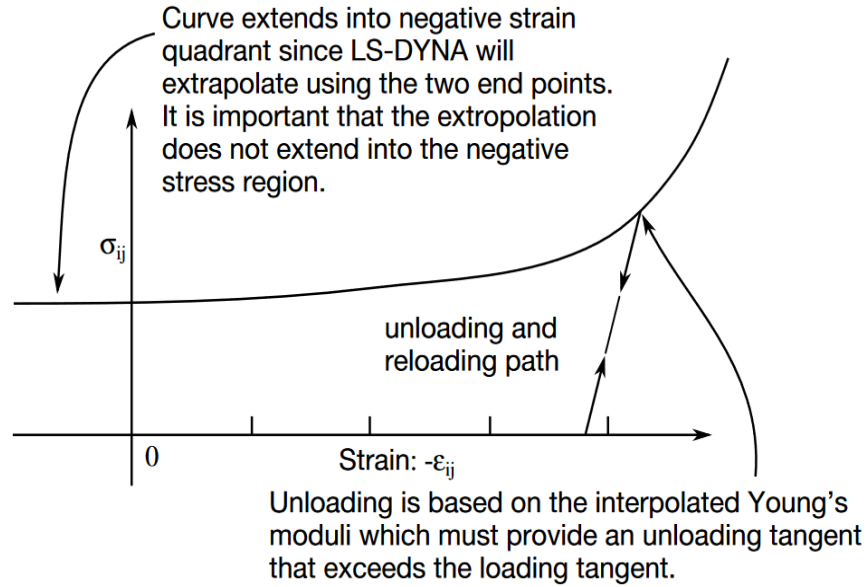


Figure 2.5: Load curve; [18, p.796-809]

The stress-strain relationship for the plastic range is given in LS-DYNA by the yield curve, see Figure 2.5 and [18, Ch.2, p.806], [17, Ch.19]. The general description of the stress-strain curve of the elasto-plastic behaviour one can find in the Section 2.4. In our model it is important that the stress is never negative. In Figure 2.5 the function in the range to the left of 0 represents the tension and in the range to the right of 0 the pressure. When the pressure decreases, the stress goes to zero, along the unloading path in Figure 2.5. Since we are in plastic range, the strain will not be zero. Another important property is that the stress is not zero when the strain is zero. This is important because $\sigma_{ij}(0)$ is the elastic limit, which is denoted as *sigy* in Table 2.1.

The load curves, which are used in our model, are given in Figure 2.6. In the material card in Figure 2.4 they are denoted by *lca*, *lcb*, *lcc* and described in Table 2.1. The curves describe the stress-strain relationship in the main directions \vec{a} , \vec{b} , \vec{c} , see Figure 2.3. Since our layers are very thin, the shear stresses do not play an important role in our case, so we have omitted them. For this reason, we have set *lcab*, *lcbc*, *lcca* to zero in the material card, Figure 2.4.

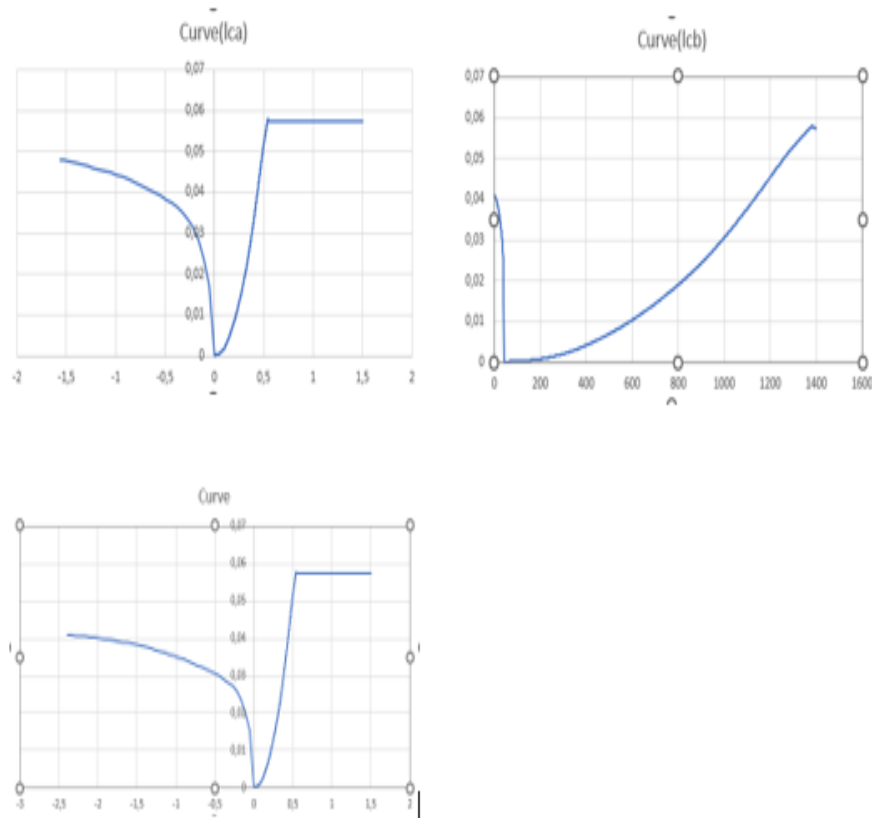


Figure 2.6: Load curves; Copyright: Virtual Vehicle

2.5.1 Stress Update

The nonlinear stress-strain relation of the modified Honeycomb material is described on the discrete level as in [18, Ch. 2, p. 796-809] and in Section 2.3. The stress changes in every time step for the material modified Honeycomb. The trial stress components $\hat{\sigma}_{ij}$ are according to a possible linear behaviour

$$\begin{aligned}\hat{\sigma}_{ii}^{n+1} &= \sigma_{ii}^n + E_{ii}\Delta\varepsilon_{ii}, \\ \hat{\sigma}_{ij}^{n+1} &= \sigma_{ij}^n + G_{ij}\Delta\varepsilon_{ij},\end{aligned}$$

where quantities with superscript n are related to the time t_n . The strain increment is given as

$$\Delta\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial \Delta u_i}{\partial y_j} + \frac{\partial \Delta u_j}{\partial y_i} \right),$$

where $\Delta u_i = u_i^{n+1} - u_i^n$ are the incremental displacements and y_j are the deformed coordinates. [17] More about the strain increment can be found in [17, Ch. 5.3, p. 5.8] and [24, eq.(40)]. The shear modulus G_{ij} describes the response of the material to the stress tensor. Moreover, the elastic modulus E_{ii} indicates the stiffness of the material in the elastic range. The elastic and shear module are computed as

$$\begin{aligned}E_{ii} &= E_{iiu} + \beta(E - E_{iiu}), \\ G_{ij} &= G_{iju} + \beta(G - G_{iju}).\end{aligned}$$

The quantities E, E_{ii}, G_{ij} are described in Table 2.1. The constant β is given as $\beta = \max \left\{ \min \left(\frac{1-V}{1-V_f}, 1 \right), 0 \right\}$, where V is the relative volume and V_f is the relative volume of the fully compacted element. The relative volume V is defined as the ratio between the current volume and the initial volume. Typically, $V = 1$ at the beginning of a calculation. [17] The elastic shear modulus for fully compacted element is computed as $G = \frac{E}{2(1+\nu)}$. The elastic modulus is defined as the slope of the load curve, which in general looks like as the curve in Figure 2.5. When the stress reaches a limit, the actual yield stress, the deformation goes from elastic to the plastic range. After that the stress tensor will take the values from load curve $\sigma_{ij}(\varepsilon_{ij})$. In our model the load curves $\sigma_{ij}(\varepsilon_{ij})$ look like in Figure 2.6. The stress update is described in [18, Ch. 2, p.806-809]. Each component of the trial stress tensor is checked, if it exceeds the allowable value determined from the load curves, which you can see in Figure 2.6, i.e. if

$$|\hat{\sigma}_{ij}^{n+1}| > \lambda \sigma_{ij}(\varepsilon_{ij}),$$

where $\lambda = 1$, in our model, then the current stress component is set to the value related to $\sigma_{ij}(\varepsilon_{ij})$ on the load curve

$$\sigma_{ij}^{n+1} = \sigma_{ij}(\varepsilon_{ij}) \frac{\lambda \hat{\sigma}_{ii}^{n+1}}{|\hat{\sigma}_{ii}^{n+1}|},$$

else it is set to the trial stress component

$$\sigma_{ij}^{n+1} = \hat{\sigma}_{ii}^{n+1}.$$

2.6 Volumetric strain as failure criterion

The failure criterion considered in the thesis is based on the idea in [33]. Failure criteria are typically based on the stress or strain state calculated as a result of the simulation and compared with a critical value of the material data. Let ε_c be the critical value which typically depends on the battery. For the meso model (Fig. 1.4), we get the value from measurements, and for the macro model (Fig. 1.6), it is an empirical value. The volumetric strain is given as

$$\varepsilon_{vol} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3,$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are the principal strains.

In our examples, the y -direction is the direction of the largest values because the impactor moves in the y -direction. For this reason, the values of volumetric strain and strain in the y -direction are very similar. For simplicity, we use the strains in the y -direction as a good approximation to the volumetric strain.

Our short circuit criterion is:

For

$$\frac{|\varepsilon_2|}{\varepsilon_c} \geq 1$$

short circuit appears and for

$$0 \leq \frac{|\varepsilon_2|}{\varepsilon_c} < 1$$

short circuit doesn't appear.

3 Empirical Interpolation Method

The Empirical Interpolation Method is a numerical technique that has many applications in numerical analysis and model order reduction. The method is used to solve high-dimensional problems through the lower-dimensional approximation of the function. The basic approach is to use a collection of evaluation points for the construction of an interpolation, which is then used to calculate the approximation. This chapter is based on the ideas of [29], [14] and [26].

We consider a parameter set $\mathcal{P} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N\}$ with $|\mathcal{P}| = N$ and determine a set of the interpolation points $T_M = \{\mathbf{t}^1, \dots, \mathbf{t}^M\}$ that plays an important role in our analysis. The basis functions $X_M = \text{span}\{\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_M\}$ are carefully chosen to ensure an appropriate representation of the underlying data. We search an efficient approximation of functions $g(\cdot, \boldsymbol{\mu})$, where a function $g(\cdot, \boldsymbol{\mu})$ belongs to a family of parameter-dependent functions $\mathcal{G} = \{g(\cdot, \boldsymbol{\mu}), \boldsymbol{\mu} \in \mathcal{P}\} \subset C^0(\bar{\Omega})$, with $\Omega \subset \mathbb{R}^d$. Moreover, for $g(\cdot, \boldsymbol{\mu})$ in \mathcal{G} we define the maximum norm as

$$\|g(\cdot, \boldsymbol{\mu})\|_{L_\infty(\Omega)} = \sup_{\mathbf{x} \in \Omega} |g(\mathbf{x}, \boldsymbol{\mu})|.$$

3.1 Basic Method

In our study we follow [29]. First we consider the method for $m = 1$, and then generalize the approach to larger $m \in \mathbb{N}$ where $m \leq M$ and M is smaller than N . To initiate the interpolation process, we select the first sampling point in the following way

$$\boldsymbol{\mu}_1^{EIM} = \arg \max_{\boldsymbol{\mu} \in \mathcal{P}} \|g(\cdot, \boldsymbol{\mu})\|_{L_\infty(\Omega)}.$$

Then the first generating function looks like

$$r_1(\mathbf{x}) = g(\mathbf{x}, \boldsymbol{\mu}_1^{EIM}), \quad \mathbf{x} \in \Omega.$$

Next we determine the first interpolation point

$$\mathbf{t}^1 = \arg \max_{\mathbf{x} \in \bar{\Omega}} |r_1(\mathbf{x})| \quad \text{and} \quad T_1 = \{\mathbf{t}^1\}.$$

After choosing the first interpolation point, we set the initial basis function based on this point as

$$\rho_1(\mathbf{x}) = \frac{r_1(\mathbf{x})}{r_1(\mathbf{t}^1)} \quad \text{and} \quad X_1 = \{\rho_1\}.$$

If we want to find the interpolation

$$(\mathcal{I}_1)g(\mathbf{x}, \boldsymbol{\mu}) = \alpha_{1,1}(\boldsymbol{\mu})\rho_1(\mathbf{x})$$

for some $\boldsymbol{\mu} \in \mathcal{P}$, we have to satisfy the interpolation equation

$$\mathcal{I}_1 g(\mathbf{t}^1, \boldsymbol{\mu}) = \alpha_{1,1}(\boldsymbol{\mu})\rho_1(\mathbf{t}^1) = g(\mathbf{t}^1, \boldsymbol{\mu})$$

for that $\boldsymbol{\mu} \in \mathcal{P}$. Thus we have to find $\alpha_{1,1}(\boldsymbol{\mu})$ from the linear equation

$$\mathbb{Q}_1 \alpha_{1,1} = g(\mathbf{t}^1, \boldsymbol{\mu}),$$

with the interpolation matrix $\mathbb{Q}_1 \in \mathbb{R}^{1 \times 1}$ and

$$\mathbb{Q}_1[1, 1] = \rho_1(\mathbf{t}^1) = 1.$$

The already established process for $m = 1$ serves as the basis for the extension to larger m . By expanding the set of basis functions ρ_i we can define an enriched interpolation

$$\mathcal{I}_m g = \sum_{j=1}^m \alpha_{m,j}(g)\rho_j.$$

Having established the set of interpolation points $T_{m-1} = \{\mathbf{t}^1, \dots, \mathbf{t}^{m-1}\}$ and the basis functions $X_{m-1} = \text{span}\{\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_{m-1}\}$ in the $(m-1)$ -th step, we now proceed in the m -th step with $m \leq M$. Here we select an additional sampling point that maximizes the error between the function g and the previous approximation \mathcal{I}_{m-1} over $\boldsymbol{\mu}$ i.e.

$$\boldsymbol{\mu}_m^{EIM} = \arg \max_{\boldsymbol{\mu} \in \mathcal{P}} \|g(\cdot, \boldsymbol{\mu}) - \mathcal{I}_{m-1}g(\cdot, \boldsymbol{\mu})\|_{L_\infty(\Omega)}.$$

By maximizing the error between g and $\mathcal{I}_{m-1}g$, we identify a parameter where the previous interpolation still has potential for improvement. Now, we compute the related residual,

$$r_m(\mathbf{x}) = g(\mathbf{x}, \boldsymbol{\mu}_m^{EIM}) - \mathcal{I}_{m-1}g(\mathbf{x}, \boldsymbol{\mu}_m^{EIM}),$$

which we use to establish the m -th basis function. Next, we select the related interpolation points by using an appropriate strategy. The selection of the interpolation points is important to achieve an accurate and robust approximation and to ensure a good performance of the interpolation procedure. The next interpolation point is computed as

$$\mathbf{t}^m = \arg \max_{\mathbf{x} \in \Omega} |r_m(\mathbf{x})|.$$

Finally, we define the basis functions as

$$\rho_m(\mathbf{x}) = \frac{r_m(\mathbf{x})}{r_m(\mathbf{t}^m)}.$$

The related interpolation

$$(\mathcal{I}_m g)(\mathbf{x}, \boldsymbol{\mu}) = \sum_{\ell=1}^m \alpha_{m,\ell}(\boldsymbol{\mu}) \rho_\ell(\mathbf{x}), \quad (3.1)$$

is determined from the interpolation equations

$$(\mathcal{I}_m g)(\mathbf{t}^j, \boldsymbol{\mu}) = g(\mathbf{t}^j, \boldsymbol{\mu}) \quad \text{for all } j \in \{1, \dots, M\}$$

for a considered $\boldsymbol{\mu} \in \mathcal{P}$. Thus the coefficients $\alpha_{m,\ell}(\boldsymbol{\mu})$ are the solutions of the linear system

$$\mathbb{Q}_m \boldsymbol{\alpha}_m(\boldsymbol{\mu}) = \mathbf{b}(\boldsymbol{\mu}),$$

where $b_j(\boldsymbol{\mu}) = g(\mathbf{t}^j, \boldsymbol{\mu})$ and $(\mathbb{Q}_m)[j, \ell] = \rho_\ell(\mathbf{t}^j)$. Therefore, the coefficients are computed as

$$\alpha_{m,\ell}(g) = \sum_{i=1}^m (\mathbb{Q}_m^{-1})_{\ell,i} g(\mathbf{t}^i, \boldsymbol{\mu}). \quad (3.2)$$

With each iteration, a space X_m is created, so that in the end we have a sequence of spaces $X_1 \subset X_2 \subset \dots \subset X_m$, so that the interpolation is exact for all $v \in X_m$, i.e.

$$\mathcal{I}_m v = v \quad \forall v \in X_m$$

for $m \in \{1, \dots, M\}$.

Theorem 3.1. [29, Theorem 10.1.] *Let M be less or equal as the dimension of $\text{span}\{\mathcal{G}\}$, where \mathcal{G} is the set of the parameter dependent functions $g(\cdot, \boldsymbol{\mu})$. Then for all $m \leq M$ hold:*

- (i) *The matrix \mathbb{Q}_m is a lower triangular matrix with unity diagonal and thus invertible.*
- (ii) *The space $X_m = \text{span}\{\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_m\}$ has dimension m .*

In the following, we deal with the error analysis of the Empirical interpolation method. Error analysis is crucial to evaluate how well the approximation produced by the numerical method corresponds to the original function. We rewrite the interpolation (3.1) and (3.2)

$$(\mathcal{I}_M g)(\mathbf{x}, \boldsymbol{\mu}) = \sum_{i=1}^M g(\mathbf{t}^i, \boldsymbol{\mu}) l_i^M(\mathbf{x}) \quad (3.3)$$

by introducing the characteristic functions

$$l_i^M(\mathbf{x}) = \sum_{j=1}^M (\mathbb{Q}_M^{-1})_{j,i} \rho_j(\mathbf{x}). \quad (3.4)$$

In the next theorem we present the interpolation error. The interpolation error quantifies the difference between the original function and its approximation created by the Empirical interpolation method.

Theorem 3.2. *[29, Proposition 10.1] Let $g \in \mathcal{G}$ and $\Lambda_M = \sup_{\mathbf{x} \in \Omega} \sum_{i=1}^M |l_i^M(\mathbf{x})|$ the Lebesgue constant. Then the interpolation error satisfies*

$$e_M(\boldsymbol{\mu}) = \|g(\cdot, \boldsymbol{\mu}) - \mathcal{I}_M g(\cdot, \boldsymbol{\mu})\|_{L_\infty(\Omega)} \leq (1 + \Lambda_M) \inf_{g_M \in X_M} \|g(\cdot, \boldsymbol{\mu}) - g_M\|_{L_\infty(\Omega)}.$$

The statement as well as additional knowledge on error analysis can be found in [29]. Based on the information from this source we can also prove the statement.

Proposition 3.3. *[29, p. 199] The Lebesgue constant Λ_M related to (3.4) satisfies*

$$\Lambda_M \leq 2^M - 1.$$

3.2 Implementation of the Empirical Interpolation Method

In the previous Section 3.1, we gained some insight into the theoretical background of the Empirical Interpolation Method. This theoretical background allows us to understand the fundamental underlying concept of the Empirical Interpolation Method. In this section, we focus on the practical implementation and execution of the Empirical Interpolation Method, still relying on the book [29]. We will discuss the steps required to construct the necessary empirical basis functions and how they are applied in practice to efficiently reduce high-dimensional problems.

We have orientated ourselves on the Algorithm 10.2 from [29, Ch. 10.2, p. 202]. Firstly, let us mention some important quantities:

- **Input**

- snapshot matrix $S \in \mathbb{R}^{N_q \times N}$
- maximal number of iterations M

- **Output**

For some $m \leq M$:

- matrix of basis vectors $\mathbb{Q}_m \in \mathbb{R}^{N_q \times m}$
- approximations $S_m \in \mathbb{R}^{N_q \times N}$.

Our own algorithm is also divided into an offline and an online phase. Complex calculations are carried out during the offline phase. During the offline phase a basis is constructed, which can be used during the online phase. In the online phase low-dimensional interpolation problems are solved to provide an approximation quickly.

At the beginning of the offline phase, a parameter set $\mathcal{P} = \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N$ is considered, where $|\mathcal{P}| = N$ represents the number of parameters. The columns of the snapshot matrix S are defined as the results of the evaluation of the function $g(\cdot, \boldsymbol{\mu}_i)$ for the parameters $\boldsymbol{\mu}_i$ from the parameter set \mathcal{P} in discrete points \mathbf{x}^i , *i.e.*

$$S = [\mathbf{g}(\boldsymbol{\mu}_1) | \dots | \mathbf{g}(\boldsymbol{\mu}_N)] \in \mathbb{R}^{N_q \times N},$$

where $\mathbf{g}(\boldsymbol{\mu})[i] = g(\mathbf{x}^i, \boldsymbol{\mu})$ for $i = 1, \dots, N_q$. In our case the matrix is created by reading data from our finite elements calculations and using this data as input. Within the framework of this method, the computation of the empirical basis vectors $\boldsymbol{\rho}_i$ is performed.

After the data is generated for the input, we choose the first sample point $\boldsymbol{\mu}_1^{EIM}$ related to the column of S with the maximal norm, accordingly we choose the index k_1 as

$$k_1 = \arg \max_{j \in \{1, \dots, N\}} \|S[:, j]\|_{max},$$

where

$$\|\mathbf{b}\|_{max} = \max_{i=1, \dots, N_q} |\mathbf{b}[i]|.$$

Correspondingly, the chosen parameter is

$$\boldsymbol{\mu}_1^{EIM} = \boldsymbol{\mu}_{k_1}.$$

The first generating vector is

$$\mathbf{r} = S[:, k_1] \in \mathbb{R}^{N_q}.$$

The first interpolation index is computed as

$$i_1 = \arg \max_{i=1, \dots, N_q} |\mathbf{r}[i]|.$$

The related interpolation point is $\mathbf{t}^1 = \mathbf{x}^{i_1}$, and the first basis vector

$$\boldsymbol{\rho}_1 = \frac{\mathbf{r}}{\mathbf{r}[i_1]}$$

is stored in the matrix $\mathbb{Q} \in \mathbb{R}^{N_q \times 1}$. The offline phase is completed by calculating the basis vector, which serves as the basis for the reduction in the parametric dimension.

In the subsequent online phase, interpolation takes place based on this previously calculated basis. The related interpolation for a given vector $\mathbf{b} \in \mathbb{R}^{N_q}$ is given as

$$\mathcal{I}_1 \mathbf{b} = \alpha_{1,1} \boldsymbol{\rho}_1,$$

where the coefficient $\alpha_{1,1}$ is the solution of the interpolation equation

$$\alpha_{1,1} \boldsymbol{\rho}_1[i_1] = \mathbf{b}[i_1],$$

and

$$\mathbb{Q}_1 \alpha_{1,1} = \mathbf{b}[i_1],$$

respectively, with the interpolation matrix $\mathbb{Q}_1 \in \mathbb{R}^{1 \times 1}$ and

$$\mathbb{Q}_1[1, 1] = \boldsymbol{\rho}_1[i_1].$$

In particular, we consider $\mathbf{b} = S[:, j]$ for $j \in \{1, \dots, N\}$ and collect the related approximations in a matrix S_1 .

For the following calculation we consider $m \in \{2, \dots, M\}$. As a new interpolation is calculated for each m , there is both an offline and an online phase in each step. As in the case of $m = 1$, the offline phase begins with the generation of the required data. In the m -th iteration step, we determine the basis vector $\boldsymbol{\rho}_m$ and the iteration index i_m . To find the basis vector, we need the generating vector evaluated in a sampling point $\boldsymbol{\mu}_m^{EIM}$. For this, we first choose the index k_m as the

$$k_m = \arg \max_{j \in \{1, \dots, N\}} \|S[:, j] - S_{m-1}[:, j]\|_{max},$$

where S_{m-1} is the interpolation matrix from the $(m-1)$ -th step, i.e. $S_{m-1} = \mathcal{I}_{m-1} S$. The sampling point $\boldsymbol{\mu}_m^{EIM}$ is given as

$$\boldsymbol{\mu}_m^{EIM} = \boldsymbol{\mu}_{k_m}.$$

The new generating vector is computed as

$$\mathbf{r} = S[:, k_m] - S_{m-1}[:, k_m].$$

The next interpolation index is determined as

$$i_m = \arg \max_{i=1, \dots, N_q} |\mathbf{r}[i]|,$$

for $m \in \{1, \dots, M\}$. The related interpolation point is $\mathbf{t}^m = \mathbf{x}^{i_m}$. The new basis vector is set to

$$\boldsymbol{\rho}_m = \frac{\mathbf{r}(\boldsymbol{\mu}_m^{EIM})}{\mathbf{r}[i_m]}.$$

The basis vectors are stored as the columns of the matrix \mathbb{Q} , i.e. $\mathbb{Q}[i, m] = \boldsymbol{\rho}_m[i]$ for $i = 1, \dots, N_q$. After we have calculated the matrix \mathbb{Q} , we start again with the online phase to calculate the corresponding interpolation. Then the related interpolation of a given vector $\mathbf{b} \in \mathbb{R}^{N_q}$ is given as

$$\mathcal{I}_m \mathbf{b} = \sum_{\ell=1}^m \alpha_{m,\ell} \boldsymbol{\rho}_\ell = \mathbb{Q} \boldsymbol{\alpha}_m,$$

where the coefficients $\boldsymbol{\alpha}_m$ are the solutions of the linear system

$$\mathbb{Q}_m \boldsymbol{\alpha}_m = \mathbf{h},$$

with $\mathbf{h}[j] = \mathbf{b}[i_j]$ for $j = 1, \dots, m$ and a square matrix $\mathbb{Q}_m \in \mathbb{R}^{m \times m}$ with the entries $\mathbb{Q}_m[j, \ell] = \mathbb{Q}[i_j, \ell]$ for $j, \ell = 1, \dots, m$. We consider $\mathbf{b} = S[:, j]$ for $j \in \{1, \dots, N\}$ and collect the related approximations in a matrix S_m . Instead of specifying the maximal number of iterations, we can stop the algorithm if $\|\mathbf{r}(\mu_m^{EIM})\|$ is below a prescribe value, i.e. the approximations are sufficiently accurate.

4 Model Order Reduction by the Empirical Interpolation Method

In this chapter, we discuss the application of the empirical interpolation method to answer our research question. Our task was to find a method that would allow us to reduce the model, see Section 1.2. Importantly, based on the reduced model, we can use the failure criterion, see Section 2.6, to make a faster short circuit prediction. In the first section, we describe the numerical model, which corresponds to the indentation tests from Section 1.2. After that we explain how we derived the method. Finally, we present the numerical results. These results demonstrate the effectiveness of the Empirical Interpolation Method in the context of our research goal.

4.1 Numerical Model of the Load Cases

In Section 1.2, we provided information about the geometry of the indentation cases. In this chapter, we use other data as in Section 1.2, see Table 4.1a and 4.1b, for the finite element models. Fig. 4.1 shows the geometrical setting of the macro models used for the computation.

This model illustrates the indentation tests in case where the battery is pressed with an impactor, which can be a sphere or a cylinder. The acceleration of the impactor is constant 0.14 mm/s^2 in negative y -direction in the time interval $[0, 3.4]$. Furthermore, the jellyroll, the red part in the Fig. 4.1, is modeled as a solid body, see Chapter 2. The remaining parts of the model (pouch, impactor, plate) are modeled as a shell.

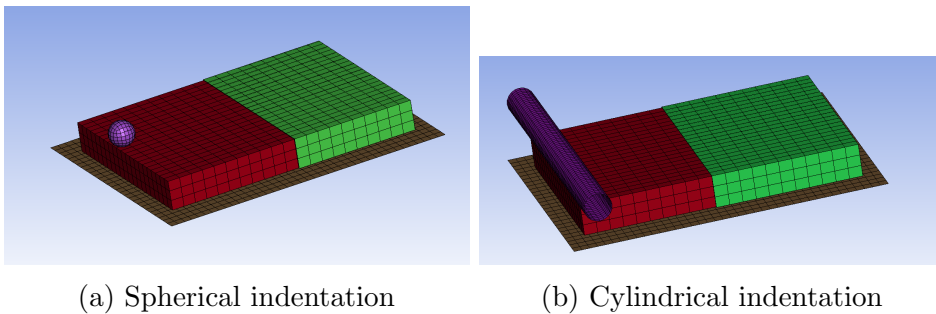


Figure 4.1: Isometric projection of the macro models. The pouch (green) is half blanked out such that the jellyroll (red) is visible.

Part	number of finite elements
Sphere	151
Cylinder	1339
Pouch	1133
Jellyroll	1323
Plate	1137

(a) Number of the elements

Part	Length	Width
Sphere	0.308999	0.308999
Cylinder	0.312869	0.343283
Pouch	0.952381	0.714286
Jellyroll	0.942841	0.760067
Plate	0.5625	0.70833

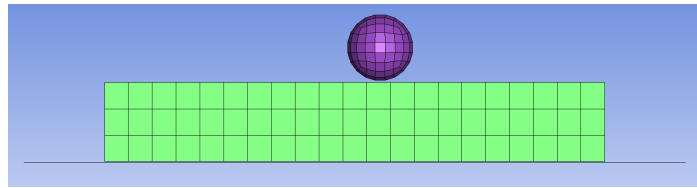
(b) Dimension of elements in mm

Table 4.1: Information about computational model of indentation cases

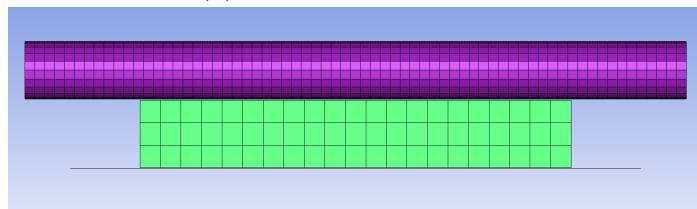
Table 4.1a shows how many finite elements are in each part of the model. As we mentioned before, the jellyroll is a solid body and for this purpose, we employ the eight-node hexahedral elements. This type of finite elements is described in Section 2.2. To get a better idea of the dimensions of the undeformed elements, their length is 0.942841, width is 0.760067 and height is 0.70714 mm. Table 4.1b shows the dimension of the shell elements. The elements are of rectangular shape, as indicated in the Table 4.1b and visible in the Fig. 4.1.

The pouch around the battery cells is rectangular in shape and its length is 20, the width is 15 and the height is 2.35 mm. The jellyroll which contains the active area of the battery has a length of 19.8, a width of 14.85 and a height 2.2802 mm.

The spherical impactor is modeled as a rigid body with radius 1 mm and is centered in (8.263568, 2.92, 0.56667614) at initial time. The center point of the sphere is over the point (8.263568, 1.4582766, 0.56667614), see Fig. 4.1a and Fig. 4.2a, which corresponds to the center of gravity of an element of the jellyroll. In a similar way, the cylinder is modeled as a rigid body in the simulation.



(a) Spherical indentation



(b) Cylindrical indentation

Figure 4.2: Front view of the macro models

Its symmetry axis passes through the same point as the center of the sphere in the case of spherical indentation model and is parallel to the pouch, see Fig 4.1b and Fig. 4.2b. Moreover, the length of the cylinder is 23.0868 mm.

The main focus of our work is to design a reduced model to predict short circuits with the short circuit criterion described in Section 2.6. For this reason, we concentrate on the strains in y -direction for each element of the jellyroll at the last time step, corresponding to the appropriate parameters $\boldsymbol{\mu}$. For the sphere, the parameterization is done by its initial position in the x - and z -directions as well as the radius. For the cylinder, the initial position in the z -direction and the radius are used as parameters.

4.2 Application of the Empirical Interpolation Method

In this section, we describe how the Empirical Interpolation Method, which we discussed in Chapter 3, in combination with other methods can be used for model order reduction. Assuming that the strains in y -direction are quite invariant to translations of the impactors, we put the strains y -direction for different parameters on top of each other by appropriate shifts. This approach aims to achieve an improved low-rank approximation by the Empirical Interpolation Method. Subsequently, the reduced approximation is transformed back to the corresponding original position. The idea of data shifting has been used in model order reduction for other applications, see [5], [31], [32].

Firstly, let $s(\mathbf{x}, \boldsymbol{\mu}) \in \mathbb{R}$ be a function evaluated in position \mathbf{x} and for parameter $\boldsymbol{\mu}$. In our application s is the strain y -direction. Moreover, we define the transformation $f_{\boldsymbol{\mu}}$ of the coordinate \mathbf{x} to a coordinate $\hat{\mathbf{x}}$ in a reference system by

$$\hat{\mathbf{x}} = f_{\boldsymbol{\mu}}(\mathbf{x}) = \mathbf{x} - (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}), \quad (4.1)$$

where $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ belong to the parameter set $\mathcal{P} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N\}$ and $\hat{\boldsymbol{\mu}}$ is a fixed chosen parameter. Here, we assume that \mathcal{P} is only related to the initial center of the sphere. This transformation applies a shift to the initial variable \mathbf{x} based on the difference between a parameter $\boldsymbol{\mu}$ and the chosen parameter $\hat{\boldsymbol{\mu}}$. After applying this transformation to all data sets we get a new parameterized function

$$\hat{s}(\hat{\mathbf{x}}, \boldsymbol{\mu}) = s(\hat{\mathbf{x}} + (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})).$$

On a fixed finite element grid and for the considered initial position this corresponds to some index shift of the matrix entries of $S \in \mathbb{R}^{N_q \times N}$ with N_q the number of elements, N the number of parameters and

$$S[i, j] = s(\mathbf{x}_i, \boldsymbol{\mu}_j)$$

for $i = 1, \dots, N_q$ and $j = 1, \dots, N$.

Since shifting the data leads to a loss of data and other data is missing, respectively, we have considered two approaches. The first approach is to set the missing data to zero. The second approach is to complete the data by a reflection with respect to the center of contact in each plane. We denote the resulting matrix by $\hat{S}(\hat{\mathbf{x}}, \boldsymbol{\mu}) \in \mathbb{R}^{N_q \times N}$.

The Empirical Interpolation Method, see Chapter 3, is applied on the matrix \hat{S} . Since the method computes a low-rank approximation, an interpolation error in the reference configuration arises

$$e = \|\hat{S} - \mathcal{I}_M \hat{S}\|_\infty \quad (4.2)$$

with the interpolation $\mathcal{I}_M \hat{S}$ of the matrix \hat{S} and the norm given as

$$\|A\|_\infty = \max_{i,j} |A[i, j]|.$$

An error estimate is given in Theorem 3.2. After the low-rank approximation is obtained, we reverse the transformation. From (4.1) follows

$$\mathbf{x} = \hat{\mathbf{x}} + (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}).$$

This transformation allows us to push the low-rank approximation back to the original position and get an approximation \tilde{S} . Subsequently, we evaluate the error between the original matrix S and the approximation \tilde{S}

$$E = \|S - \tilde{S}\|_\infty. \quad (4.3)$$

This error allows us to understand how well the low-rank approximation represents the behavior of original matrix and solution vectors, respectively.

4.3 Basic Test Case

In this subsection, we consider the spherical indentation test for several positions of the sphere in z -direction. The choice of different positions of the sphere in the z -direction was motivated by an interest in studying the effects of battery loading on the occurrence of short circuits. In this case, the parameters are calculated as multiples of the element length $h = 0.942841$, i.e.

$$\mu_j = j \cdot h \quad \text{for } j \in \{0, \dots, 20\}.$$

Thus the sphere is always placed exactly above the center of gravity of an element. The parameter set is $\mathcal{P} = \{0, 0.94841, \dots, 18.85682\} = \{\mu_0, \mu_1, \dots, \mu_{20}\}$. The situation seen in Fig. 4.3 is for μ_0 . For the parameter μ_0 the sphere pressure is applied at the left margin of the battery, see Fig. 4.3, while at parameter μ_{20} it is applied at the right margin. Meanwhile, the pressure is applied to any element of the jellyroll that

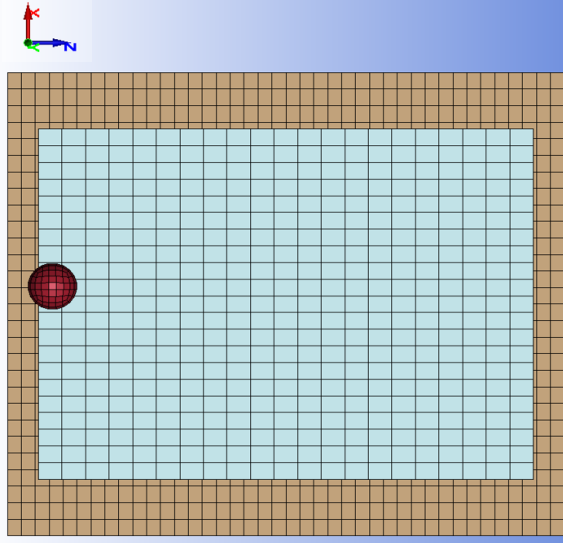


Figure 4.3: Top view of the macro model for spherical indentation test

matches the same x and y coordinates as the center of gravity of the sphere for this parameter, as the sphere only moves in the z -direction. For the parameter μ_0 the sphere is centered in $(8.263568, 2.29, 0.56667614)$.

In this context, the entries of the snapshot matrix $S \in \mathbb{R}^{N_q \times N}$ are the strain y -direction values for each element of the jellyroll at final time, corresponding to the different parameters μ . Each column of the matrix S represents the strains y -direction for a parameter μ_j for $j \in \{0, \dots, 20\}$. The number of columns in the matrix is $N = 21$, which corresponds to the number of parameters considered. The rows of the snapshot matrix S correspond to the individual elements of the jellyroll. The total number of these elements is denoted as N_q .

4.4 Approximation by the Empirical Interpolation Method

In the previous sections, we presented the numerical model of the load cases and the basis methodology employed for our reduced order modeling. In this section, we demonstrate the results of our analysis for different approaches. Our investigation first include the comparison of the results on a coarse and finer mesh. Subsequently, we examine the interpolation errors according to equation (4.2) as well as the error between the original matrix S and its approximation \tilde{S} according to equation (4.3) in the original system for the entire parameter set P . In the process, we consider two approaches, with and without reflection, and take into account two different radii and we consider the whole parameter set.

4.4.1 Comparing the Errors of Coarse and Fine Meshes

The model from Section 4.1 was computed with the specific grid resolution. To understand the impact of the mesh on the results we refined the mesh for the jellyroll and the pouch, see Table 4.2. For both coarse and fine mesh datasets, we used the same set of parameters $\mathcal{P} = \{0, 0.94841, \dots, 18.85682\}$. By using the identical parameter set, we are able to make sure that the comparison is valid. This allows us to evaluate the grid dependence of our model. As $\hat{\mu}$ we chose in this case $\mu_{10} = 9.4841$.

The related approximation errors (4.2) for different ranks and both meshes are compared in Fig. 4.4. Unexpectedly, the error decreases more quickly for the model with the coarser mesh. The finer resolution provides more information and thus may require a larger rank. But we are also somehow surprised that we need a larger rank for a comparable reasonable error for finer finite element model as we had expected an effect of the finite element discretization error on the smoothness of the data. There can be several reasons for the observation, such as noise, perturbations, parameter selection and etc. However, it is not our task here to find the reason.

Part	number of the finite elements
Pouch	4535
Jellyroll	10584

Table 4.2: Number of the finite elements of the model with refined mesh

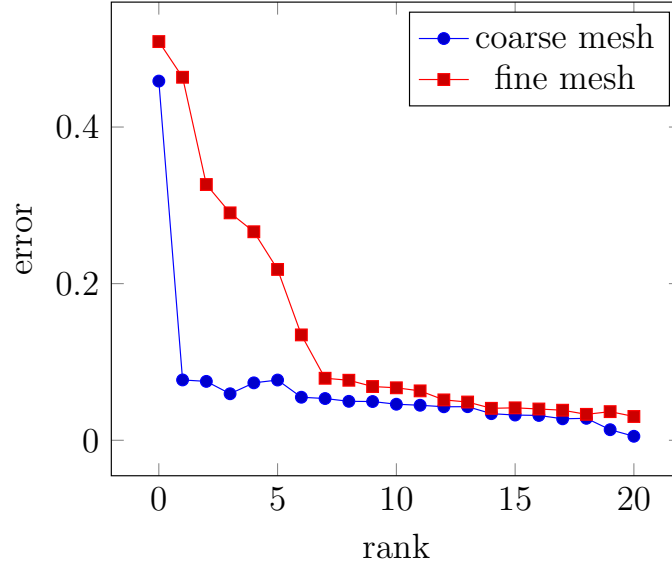


Figure 4.4: Interpolation error (4.2) for the spherical indentation with reflection in the reference configuration for two meshes

Due to the computing times and memory requirements, all further calculations were made on the coarse mesh. Using a coarse mesh allows us to reduce the computing times, making the simulation more practicable and simpler to perform the tests.

4.4.2 Low-Rank Approximation in the Entire Parameter Set

In this case, we continue to use the set of parameters $\mathcal{P} = \{0, 0.94841, \dots, 18.85682\}$ for our computation. For the reference parameter $\hat{\mu}$ we have again chosen $\mu_{10} = 9.4841$. As mentioned in Section 4.2, we used two different approaches, with and without reflection, to complete the data during the first shift.

Figure 4.5 compares the approximation errors (4.2) for both approaches and the full rank approximation. In this case, the curves are on top of each other. We only see the error introduced by a loss of information due to the shift of data. Compared to a maximal value of about 0.45 this error is acceptable.

Table 4.3 shows the error (4.3) in original system for several ranks in the Empirical Interpolation Method and the maximal number of iterations used in the procedure described in Section 3 respectively, for both approaches. In addition, Table 4.3 contains the results for a sphere with radius 2 mm and the initial center in $(8.263568, 3.29, 0.56667614)$. The influence of the reflection is noticeable. In the case $r = 1$ mm the error decreases faster already for $max_it = 2$, if the calculation is done with the reflection. For $r = 2$ mm the reflection has a stronger influence. It is further to be noted that the errors in the case of the computation without reflection are bigger for $r = 2$ mm but in the case of the computation with reflection they decrease faster as for $r = 1$ mm. Therefore, we can conclude that reflection is a good choice for the completion of the data.

In the remainder of this discussion, we will examine the results for the cylinder. We parameterize the movement of cylinder in z -direction and the parameter set $\mathcal{P} = \{0, 0.94841, \dots, 18.85682\}$ and the reference parameter $\hat{\mu} = \mu_{10} = 9.4841$ are the same as for the sphere.

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.077020579	0.077020579	0.385737270	0.117024681
2	0.091762221	0.075188369	0.131577536	0.079884598
3	0.092470461	0.059387362	0.128245893	0.062307615
4	0.092470461	0.073325006	0.129527356	0.060888271
5	0.088503311	0.076913930	0.138844301	0.055726282
6	0.086356830	0.054944315	0.126527438	0.055495608
21	0.031043823	0.031043823	0.037056356	0.037056356

Table 4.3: Approximation error (4.3) in original system for the entire parameter set \mathcal{P} in the case that the impactor is the sphere

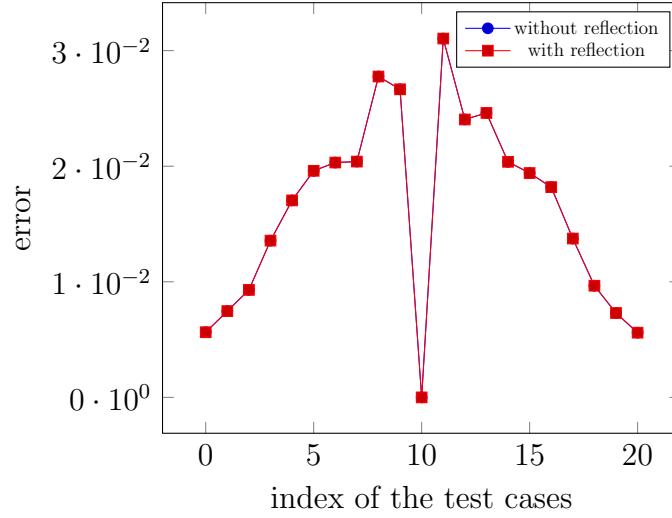


Figure 4.5: Shift error (4.2) for spherical indentation in the entire parameter set \mathcal{P} and full rank

Table 4.4 presents the errors (4.3) in the original configuration for the cylinder with base radii of $r = 1$ mm and $r = 2$ mm. An observation of interest made from the data is that the application of reflection has a positive effect on the errors. Further, we can notice that the error decays faster for the calculation with the reflection for the radius $r = 1$.

For the cylinder we get smaller errors than for the sphere, if we do the computations with the full rank. A general conclusion is that the method works also for the cylinder. In the further we will present the results only for the sphere.

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.240735337	0.152446319	0.327822595	0.197929003
2	0.240735337	0.098303366	0.225610569	0.197929003
3	0.265515631	0.082000903	0.225610569	0.197817345
4	0.146611117	0.079025340	0.108931664	0.210761317
5	0.169624789	0.071261202	0.124325802	0.182224269
6	0.128420354	0.067581520	0.088552675	0.095356712
21	0.018922030	0.018922030	0.023391012	0.023391012

Table 4.4: Approximation error (4.3) in original system for the entire parameter set \mathcal{P} in the case that the impactor is the cylinder

4.5 Local Approximation

After studying the influence of the reflection and testing the method for the entire set of parameter, we were interested to see if it makes a difference if we split the parameter set into three smaller subsets to compute local approximation, see Fig. 4.6. The concept of local approximation is already established in following source [2, Sec. 4.6.] and [20, Sec. 4.3]. This allows a more detailed evaluation of the parameter set that may have a relevance for the application. Splitting the parameter set into three subsets also allows us to better identify potential problems of the system and also to solve those locally. The local approximations may help to reduce effects of the somehow “artificial” completion of the data in the reference configuration by reflection or by zero. The choice of subsets should be well considered to ensure that relevant relationships are captured.

We represent the parameter set \mathcal{P} as a rectangle, corresponding to shape of the battery, in the Fig. 4.6. The left parameter set \mathcal{P}_ℓ is represented in green, the middle parameter set \mathcal{P}_m in blue and the right parameter set \mathcal{P}_r in orange. Furthermore, there is no overlap between the three parameter sets, i.e. $\mathcal{P}_\ell \cap \mathcal{P}_m = \emptyset, \mathcal{P}_\ell \cap \mathcal{P}_r = \emptyset, \mathcal{P}_m \cap \mathcal{P}_r = \emptyset$. At the same time, the subsets together cover the entire parameter set \mathcal{P} , i.e. $\mathcal{P}_\ell \cup \mathcal{P}_m \cup \mathcal{P}_r = \mathcal{P}$. Each parameter set can represent different characteristic and behaviors of the system.

Again, we consider the parameter set $\mathcal{P} = \{0, 0.94841, \dots, 18.85682\} = \{\mu_0, \dots, \mu_{20}\}$. To evaluate the effect of the two boundary stripes we consider two different splitting. In one case, we decided to include three parameters in each of the parameter sets \mathcal{P}_ℓ and \mathcal{P}_r . In another case, we slightly modified this splitting and considered four parameters in each of the parameter sets \mathcal{P}_ℓ and \mathcal{P}_r .

In the following, we present the results obtained by applying the local approximation in the case of a translation of the sphere in the z -direction. The parameters to that we apply the method are described in Subsection 4.3.



Figure 4.6: Dividing of the set of the parameters \mathcal{P}

4.5.1 Pure Splitting of the Parameter Set

In this case, we consider the left parameter set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$, the right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$ and the middle parameter set $\mathcal{P}_m = \mathcal{P} \setminus (\mathcal{P}_\ell \cup \mathcal{P}_r)$. In the following, we check and compare the results for all three sets. Additionally, we evaluate the errors in original system (4.3) for the radius $r = 1$ mm as well as for the radius $r = 2$ mm. By comparing the results for the left, middle and right areas, we get a comprehensive overview of how the parameters influence the system behavior.

The Left Parameter Set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$

Since we now apply the method from Section 4.2 in the left parameter set only, we choose the parameter μ_2 as the reference parameter $\hat{\mu}$. In Fig. 4.7, the interpolation error (4.2) for the case $r = 1$ mm and rank three is visualized. We notice that the displacement error does not change.

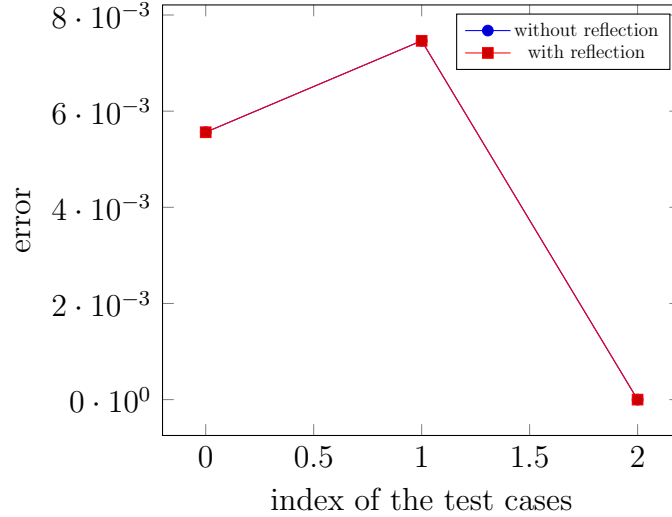


Figure 4.7: Shift error (4.2) for the spherical indentation in the left parameter set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$, full rank and radius $r = 1$

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.162693738	0.109922345	0.385737270	0.117024681
2	0.162693738	0.051721815	0.117024681	0.034419390
3	0.007463249	0.007463249	0.012866856	0.012866856

Table 4.5: Approximation error (4.3) in the original system in the left parameter set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$

Table 4.5 is useful for understanding the impact of the rank and of the reflection on the interpolation error (4.3). Lower errors in the reflection case indicate that reflection helps to get better approximations. We note that using one or two basis vectors makes no difference for the computation without reflection in the case the radius is 1 mm. Otherwise, for $r = 2$ it makes a difference whether the calculation is done with one or two basis vectors. Furthermore, it can be seen that for the case $r = 2$ mm there is a significant difference whether the calculation is carried out with one or two basis vectors. Overall, these results illustrate that the reflection and the number of basis vectors used have a significant impact on the exactness of the interpolation.

The Middle Area $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$

After considering the left parameter set, we now turn to the middle part. In this case we again choose μ_{10} as reference parameter $\hat{\mu}$. The approximation errors (4.2) can be seen in Fig. 4.8.

Table 4.6 allows us to identify specific differences between the results with and without reflection. It is interesting to note that the reflection has an influence on the error, if we use three or more basis vectors for the computation. Another observation is that the error decreases faster for \mathcal{P}_m compared to the error on the whole domain, compare Tables 4.3 and 4.6. This indicates that the two boundary strips have an influence on the error and we can benefit from local approximation.

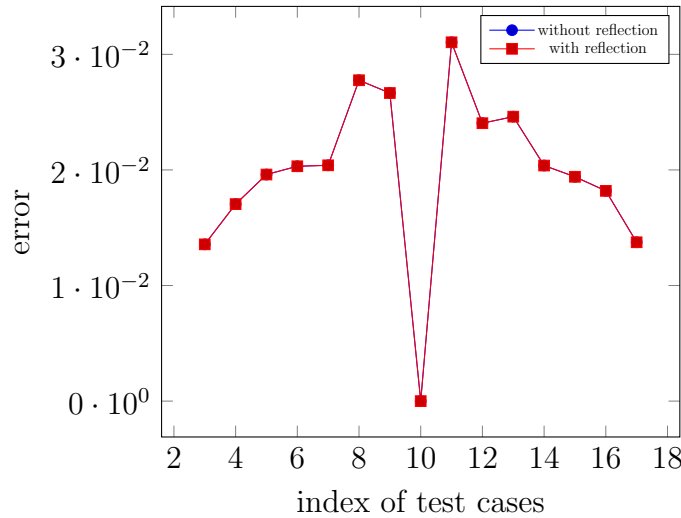


Figure 4.8: Shift error (4.2) for the spherical indentation in the middle parameter set $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$ and full rank

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.077020579	0.077020579	0.089304782	0.056962404
2	0.057447819	0.057447819	0.089304782	0.055827657
3	0.057323654	0.055160643	0.062886111	0.055776697
4	0.057327537	0.048564385	0.077481862	0.054746339
5	0.067430063	0.047445733	0.068930109	0.052779294
6	0.061036877	0.045291604	0.067909414	0.051984155
15	0.031043823	0.031043823	0.037056356	0.037056356

Table 4.6: Approximation error (4.3) in original system in the middle parameter set $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$

The Right Parameter Set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$

At the end, we now consider the remaining right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$. In this case, we choose the parameter μ_{18} as the reference parameter $\hat{\mu}$. Fig. 4.9 shows the approximation error (4.2) for the right parameter set and full rank. By comparing the results in Table 4.5 and Table 4.7, it can be seen that the errors for the left parameter set \mathcal{P}_ℓ and the right parameter set \mathcal{P}_r show similar patterns. Here the effect of the reflection is also notable. The analysis of the right parameter set completes our understanding of the influence of the parameters on the error.

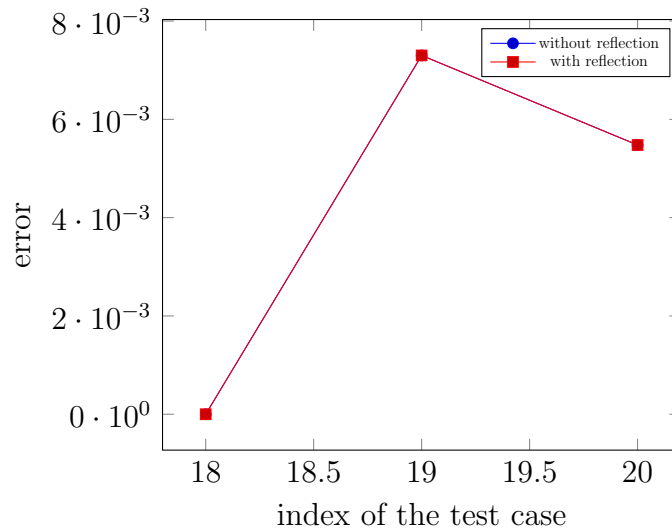


Figure 4.9: Shift error (4.2) for spherical indentation in the right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$ and full rank

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.163364440	0.114805471	0.389189988	0.112561714
2	0.163364440	0.049776363	0.112561740	0.031363091
3	0.007300539	0.007300539	0.011683957	0.011683957

Table 4.7: Approximation error (4.3) in the original system for the right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$

4.5.2 Pure Splitting with Larger Margins

In the second splitting, we increased the left and right parameter sets, thereby decreasing the middle parameter set. Since the approximation errors (4.2) for full rank look similar to the errors from the first splitting, we will not show further plots of the approximation errors (4.2). Instead, we will focus on the errors (4.3) in the original system. Focusing on the errors in the original system allows us to better assess the practical relevance of the different splitting strategies. However, we will continue to consider the influence of reflection and the two different radii, $r = 1$ and $r = 2$.

The Left Parameter Set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2, \mu_3\}$

After increasing the left parameter set, we choose μ_3 as new reference parameter $\hat{\mu}$. Similar to the first splitting, we find that when computing without reflection for the sphere with the radius $r = 1$, it makes no difference whether we make the computation with one or two basis vectors, except in the case of the computation without reflection for the sphere with the radius $r = 1$. However, when computation with reflection, it should be noted that the errors at $max_it = 1$ are the same as for the first splitting, see Table 4.5. But the errors are larger than in the first splitting, if the computation is made with more basis vectors. This observation underlines that the first splitting may be a better choice to obtain more precise approximation results.

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.185202807	0.109922345	0.385737270	0.117024681
2	0.185202807	0.065525934	0.131577536	0.055623847
3	0.082160771	0.043965500	0.131577536	0.033683607
4	0.009299094	0.009299094	0.015250443	0.015250443

Table 4.8: Approximation error (4.3) in original system in the left parameter set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2, \mu_3\}$

The Middle Parameter Set $\mathcal{P}_m = \{\mu_4, \dots, \mu_{16}\}$

Even though we reduced the size of the middle range, we kept our reference parameter $\hat{\mu}$ unchanged as μ_{10} . Keeping the reference parameter μ_{10} allows us to perform a comparative analysis of the different splitting strategies and better understand their influence on the accuracy of the approximation. We observe that the approximation errors in Table 4.9 decrease a little faster in this case than in the first splitting, see Table 4.6. Thus the second splitting is minimally better.

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.077020579	0.077020579	0.089304782	0.056962404
2	0.047045963	0.047045963	0.062886111	0.055827657
3	0.057078528	0.048779427	0.069234205	0.055776697
4	0.047002741	0.047075925	0.067569939	0.053062446
5	0.059450949	0.045632697	0.051969522	0.052175417
6	0.036358300	0.041442787	0.052126759	0.044633224
13	0.031043823	0.031043823	0.037056356	0.037056356

Table 4.9: Approximation error (4.3) in original system in the middle parameter set $\mathcal{P}_m = \{\mu_4, \dots, \mu_{16}\}$

The Right Parameter Set $\mathcal{P}_r = \{\mu_{17}, \mu_{18}, \mu_{19}, \mu_{20}\}$

Here we choose μ_{17} as a new reference parameter $\hat{\mu}$. Also in this splitting we see that the errors for the left and right parameter set behave similarly and that the influence of reflection is the same. If we compare it with the errors from Table 4.7, we see that the errors are smaller in the case of the first splitting.

rank	error, $r = 1$		error, $r = 2$	
	without reflection	with reflection	without reflection	with reflection
1	0.184548899	0.114805471	0.389189988	0.112561740
2	0.184548899	0.067149929	0.132722526	0.058046260
3	0.082949770	0.041263268	0.135677607	0.032341161
4	0.009659592	0.009659592	0.015022624	0.015022624

Table 4.10: Approximation error (4.3) in original system in the right parameter set $\mathcal{P}_r = \{\mu_{17}, \mu_{18}, \mu_{19}, \mu_{20}\}$

In our investigation, we found that the reflection generally has a positive influence on the error, especially for the left and right parameter set. Therefore, we will do all further tests for the sphere with reflection.

We have also observed that the first splitting of the parameter set is significantly better for the left and the right parameter set while the second splitting is only minimally better in the middle range. Thus, in further applications of the local approximation for the sphere, we will use the first splitting.

We tested the effects of different radii for several radii on the accuracy of the approximation but only entered the results for $r = 1$ mm and $r = 2$ mm. This observation provides that the method also works for different radii.

For a comparison of the global and the local approximation, we consider the calculation with reflection for the case when the sphere has a radius of $r = 1$. If we use six basis vectors in the computation for the entire parameter set, we get an error (4.3) in the original configuration of 0.054944315, see Table 4.3. However, if we use the local method of the first splitting and take two vectors from the left, two from the middle and two from the right parameter set in the case of local approximation, we get an error (4.3) in original configuration of 0.057447819. Thus the two approaches are comparable in this case.

4.5.3 Two-Step Approximation at the Margins

The results of the local approximation inspired us to develop another method for the local approximation. We use shifted results from the middle parameter set \mathcal{P}_m as initial guess for the approximation for the left parameter set \mathcal{P}_ℓ and the right parameter set \mathcal{P}_r and improve it by a second interpolation. So we hope to find approximations with overall lower rank.

After finding in Sections 4.4 and 4.5 that the use of reflection has a significant positive impact on the qualities of the approximation, we will test this method specifically for this case. Furthermore, we use the first splitting of the parameter set. Specifically, this splitting looks like $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$, $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$, $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$.

Let $S_m \in \mathbb{R}^{N_q \times N_m}$ be the snapshot matrix for the middle parameter set \mathcal{P}_m , where $N_m = 15$ is the number of parameters in \mathcal{P}_m . Here we assume that the basis vectors for the middle parameter set are available in Q_m , see Chapter 3. Further, the index set J_m contains information about which columns of the snapshot matrix are used for the following computation.

In the following paragraph, we describe the method using the left parameter set. The same method applies to the right parameter set. This approach allows us to study the different sets of parameters and apply the method to both sides, keeping the description and approach the same. For the left parameter set, we define the snapshot matrix $S_\ell \in \mathbb{R}^{N_q \times N_\ell}$, where $N_\ell = 3$ is the number of parameters in the left parameter set \mathcal{P}_ℓ .

As the name of the method already suggests, it consists of the two following steps. In the first step, we shift the data of \mathcal{P}_ℓ with respect to the reference parameter $\hat{\mu}_m = \mu_{10}$ and then apply the online phase of the Empirical Interpolation Method to compute an initial approximation \hat{S}_ℓ^1 with respect to μ_{10} .

In the second step of the method, we perform the approximation for the left parameter set as a correction to the approximation computed in the first step, i.e. we apply our approximation by Empirical Interpolation Method in \mathcal{P}_ℓ to $S_\ell - \hat{S}_\ell^1$. At first, we shift both the initial approximation \hat{S}_ℓ^1 and the snapshot matrix S_ℓ to the second reference parameter $\hat{\mu}_\ell = \mu_2$. This step allows us to set the approximation and the snapshot matrix to a common point. In the next step, we apply the offline phase of the Empirical Interpolation Method again to $\tilde{S}_\ell - \tilde{S}_\ell^1$. By the offline phase of the Empirical Interpolation Method we obtain an approximation \tilde{S}_ℓ^2 and after reverting the shift \tilde{S}_ℓ^2 in the original configuration. The final approximation \tilde{S}_ℓ is the sum of the two approximations \tilde{S}_ℓ^1 and \tilde{S}_ℓ^2

$$\tilde{S}_\ell = \tilde{S}_\ell^1 + \tilde{S}_\ell^2. \quad (4.4)$$

The method includes two successive approximation steps. In the first step, an approximation is performed for the middle parameter set \mathcal{P}_m . Due to the completeness of the data in the middle parameter set, we performed the tests here with reflection. As already noted in Section 4.5.2, the application of the reflection for the left and right parameter set has a positive influence on the approximation results. For this reason, we decide to perform the approximation for the left and right parameter set exclusively with reflection.

Left Parameter Set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$

Now we discuss the approximations obtained by applying the two-step method to the left parameter set \mathcal{P}_ℓ . In the approximation for the parameter set $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$ we keep μ_{10} as our reference parameter $\hat{\mu}_m$. On the other hand, for the approximation in the left parameter set \mathcal{P}_ℓ , we choose μ_2 as our reference parameter $\hat{\mu}_\ell$. Of special interest are the errors between the original matrix S_ℓ and the final approximation \tilde{S}_ℓ in the original configuration. This error is calculated as in equation (4.3).

The values listed in Table 4.11 illustrate the number of basis vectors from the matrix Q_m in the first column and the number of basis vectors we calculated for the left parameter set \mathcal{P}_ℓ in the second column. In the last column is present the error (4.3) in original configuration.

The number of basis vectors in the middle and left parameter sets influences the accuracy of the approximation. It is noticeable that whenever we use all basis vectors from the left parameter set, only shift errors occur. Furthermore, we find that it makes no significant difference whether we use two or three basis vectors from the middle parameter set.

rank		error
rank for \mathcal{P}_m	rank for \mathcal{P}_ℓ	with reflection for \mathcal{P}_m
1	0	0.072196521
1	1	0.068915184
1	2	0.050869776
1	3	0.007463249
2	0	0.069150511
2	1	0.061421169
2	2	0.042366192
2	3	0.007463249
3	0	0.068119931
3	1	0.060820114
3	2	0.042683869
3	3	0.007463249

Table 4.11: Approximation error (4.3) in the original system in the left parameter set $\mathcal{P}_\ell = \{\mu_0, \mu_1, \mu_2\}$ for the two-step approximation

If we use two basis vectors each from the middle parameter set and from the left parameter set, we find that the error is lower than for the one-step local approximation in Table 4.5 in case that the rank is two. This indicates that the combination of basis vectors from both sets leads to an improved approximation.

Right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$

We now discuss the approximation obtained by applying the two-step method for the right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$ in Table 4.12. For the approximation for the middle parameter set $\mathcal{P}_m = \{\mu_3, \dots, \mu_{17}\}$, we keep μ_{10} as the reference parameter $\hat{\mu}_m$. For the approximation in the right parameter set \mathcal{P}_r , we choose μ_{18} as the reference parameter $\hat{\mu}_r$.

We pay special attention to the errors between the original matrix S_r and the final approximation \tilde{S}_r in the original configuration as in the equation (4.3). We note that shift errors occur when using all basis vectors from the right parameter set. On the other hand, we obtain similar results as with the one-step local approximation when we use two basis vectors each from the middle and two from the right parameter set.

If we use two basis vectors for \mathcal{P}_m and two additional basis vectors in \mathcal{P}_ℓ and \mathcal{P}_r each, we have better approximations than for the global approximation by rank six, see Table 4.3, at the same costs. If we use rank two for \mathcal{P}_m and rank one for the second step approximations each, we have a comparable error but need only four instead of six vectors. Thus we notice that the local approximation together with the two-step approximation can improve the efficiency of our approximation.

rank		error
rank for \mathcal{P}_m	rank for \mathcal{P}_ℓ	with reflection for \mathcal{P}_m
1	0	0.075819130
1	1	0.057080294
1	2	0.066927796
1	3	0.007300539
2	0	0.075188369
2	1	0.056796644
2	2	0.051123730
2	3	0.007300539
3	0	0.071518685
3	1	0.056123152
3	2	0.041650692
3	3	0.007300539

Table 4.12: Approximation error (4.3) in the original system in the right parameter set $\mathcal{P}_r = \{\mu_{18}, \mu_{19}, \mu_{20}\}$ for the two-step approximation

4.6 Training on the Half Parameter Set

In this section, we dedicate ourselves to model training performed on half of the middle parameter set, $\mathcal{P}_o = \{\mu_3, \mu_5, \mu_7, \mu_9, \mu_{11}, \mu_{13}, \mu_{15}, \mu_{17}\}$, and we will approximate the data for the parameter set $\mathcal{P}_e = \{\mu_4, \mu_6, \mu_8, \mu_{10}, \mu_{12}, \mu_{14}, \mu_{16}\}$ using the approximation of the data for the parameter set \mathcal{P}_o . The middle parameter set \mathcal{P}_m is given as union of \mathcal{P}_o and \mathcal{P}_e , i.e. $\mathcal{P}_m = \mathcal{P}_o \cup \mathcal{P}_e$.

Firstly, we apply the Empirical Interpolation Method to the training set \mathcal{P}_o and after that we have two approaches how to approximate the data for the parameters from \mathcal{P}_e . Moreover, for μ_i in the parameter set \mathcal{P}_e are μ_{i-1} and μ_{i+1} in the training set \mathcal{P}_o .

Our first approach is to shift the approximation of the data for the parameter set \mathcal{P}_o to the data for the parameter set \mathcal{P}_e by shifting the approximation from μ_{i+1} or μ_{i-1} to $\mu_i \in \mathcal{P}_e$.

Our second approach follows a slightly different strategy. Here we represent the data for the parameter μ_i as the shifted mean value of the approximations for the parameters μ_{i-1} and μ_{i+1} , i.e.

$$\hat{S}[i] = \frac{\hat{S}[i-1] + \hat{S}[i+1]}{2}.$$

This approach aims to provide some consistency in the approximations between neighboring parameters. After we approximate the data for the parameter set \mathcal{P}_e , we move it back to the original configuration and there we calculate the errors (4.3).

rank	error		
	First approach		Second approach
	μ_9	μ_{11}	
1	0.0142	0.0142	0.0142
2	0.0142	0.0696	0.0339
3	0.0142	0.0696	0.0339

Table 4.13: Approximation error (4.3) in the original system in case of approximating data for the parameter μ_{10} with the approximations of μ_9 or/and μ_{11}

Table 4.13 shows the errors in the original configuration in case we approximate the data for μ_{10} with the help of data for the parameters μ_9 and μ_{11} . For the first approach, we tested two possibilities. First, we shifted the approximation of data for μ_9 to μ_{10} and then also shifted the approximation of data for μ_{11} to μ_{10} . For the second approach, we approximate data for μ_{10} as the mean value of the shifted approximations of data for μ_9 and μ_{11} . We observe a good approximation for the parameter μ_{10} . Further, we can notice that for μ_9 in the first approach we get only the shift error, since the approximation for μ_9 is already exact. By calculating the approximation for μ_{11} , μ_9 is also selected at first in the Empirical Interpolation Method for interpolation, which leads to the same errors as for μ_9 itself. In the second iteration, μ_{11} is chosen as a parameter in the Empirical Interpolation Method, which leads to an exact approximation and thus it remains without changes in rank three. The errors are caused due to the already large differences in the data sets for μ_{10} and μ_{11} .

In Table 4.14 we present results in the case where we approximate the data for the parameter μ_6 with approximations of the data for the parameters μ_5 and μ_7 . In this case, we get an error of approximately 0.033 almost everywhere, which is close to the shift error. Only for rank three in the case when we approximate the data for μ_6 by shifting the approximation of the date for μ_5 , we get a slightly larger error. Also in this case, μ_9 and μ_{11} are used in the Empirical Interpolation Method. Overall we notice that we do not need the data of all parameters to train our model.

rank	error		
	First approach		Second approach
	μ_5	μ_7	
1	0.03356	0.03316	0.03336
2	0.03530	0.03319	0.03381
3	0.05529	0.03559	0.03598

Table 4.14: Approximation error (4.3) in the original system in the case that we approximate the data for the parameter μ_6 with the approximations of μ_5 or/and μ_7

4.7 Translation in x - and z -direction

Up to now, we have performed tests only for the translation of the impactor in the z -direction, as described in Section 4.3. In this section, we will look at translation in the x - and z -directions. The parameter set for the z -direction remains as previously defined, namely as $\mathcal{P}_z = \{\mu_0, \dots, \mu_{20}\} = \{0, 0.94841, \dots, 18.85682\}$, while the parameter set for the x -direction is given as $\mathcal{P}_x = \{\mu_0, \dots, \mu_{20}\} = \{-7.778557, \dots, 6.364242999999\}$. The parameters in the parameter set \mathcal{P}_z are calculated as multiples of the element length $h_z = 0.942841$, i.e.

$$\mu_j = j \cdot h_z \quad \text{for } j \in \{0, \dots, 20\}$$

and the parameters in the parameter set \mathcal{P}_x are calculated as multiples of the element width $h_x = 0.70714$, i.e.

$$\mu_i = i \cdot h_x \quad \text{for } (i - 11) \in \{0, \dots, 20\}.$$

For the present case, we define the parameter set as the Cartesian product of these two parameter sets, making our parameters two-dimensional. Fig. 4.10 shows the sphere in the initial position for the parameter $\boldsymbol{\mu} = (\mu_0, \mu_0)$. We should also note that the distance between the sphere and the cell remains the same as in Section 4.3. The radius r of the sphere also stays the same as 1 mm.

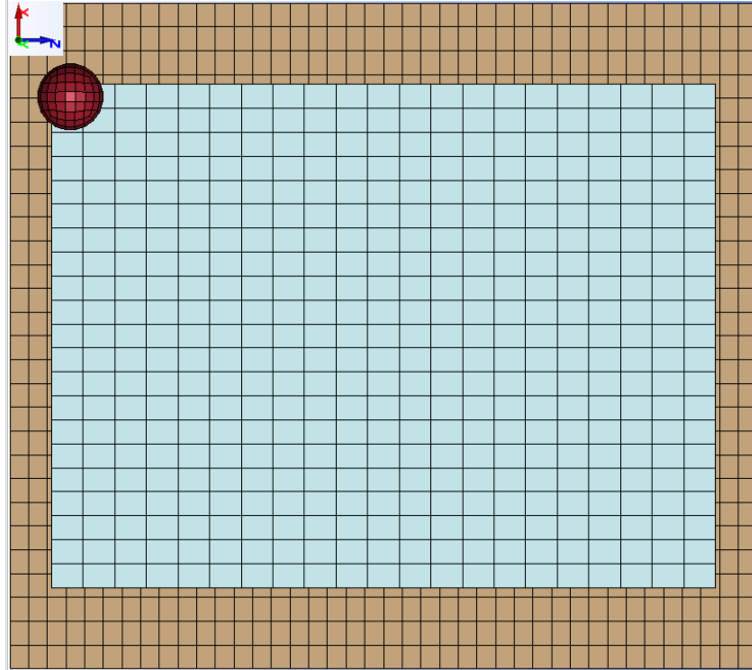


Figure 4.10: Top view of the macro model for spherical indentation test

In mathematical notation, we can express this as follows:

$$\mathcal{P}_{xz} = \{\boldsymbol{\mu} = (\mu_x, \mu_z) | \mu_x \in \mathcal{P}_x, \mu_z \in \mathcal{P}_z\},$$

where \mathcal{P}_{xz} represents the set of parameters in both directions (x and z), and each element (μ_x, μ_z) in \mathcal{P}_{xz} corresponds to a combination of parameters for the x - and z -directions. Now we can extend our tests to the two-dimensional parameter set \mathcal{P}_{xz} and examine the motion of the impactor in both directions. All computations will be done using the reflection, since it has been shown that it has a good influence on the error.

4.7.1 Global Approximation

Consistent with Section 4.4, we first apply our method to the full parameter set, analyzing both the interpolation error according to equation (4.2) and the error in the original configuration as defined by equation (4.3). Fig. 4.11 shows the error according to the equation (4.2) for the entire parameter set for different ranks. In this case we choose $\hat{\boldsymbol{\mu}} = (\mu_{10}, \mu_{10})$ as reference parameter. We observe a fast initial decay of the error, but then there seems to be a certain variance in the data sets. In addition, Table 4.15 provides an overview of the error (4.3) in the original configuration. It is noticeable that the error at a rank of 86 takes the shift error value, which actually corresponds to the error at full rank. The error at a rank of 45 is only 0.05930. All in all the error decreases quite slowly.

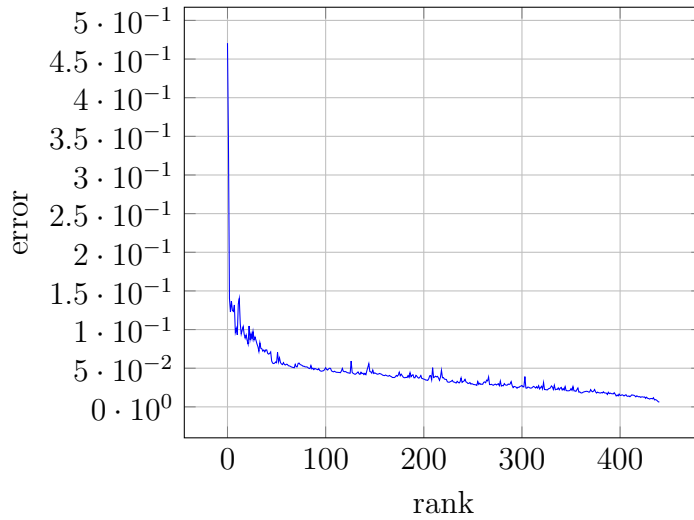


Figure 4.11: Interpolation error (4.2) in the reference configuration for spherical indentation in the parameter set \mathcal{P}_{xz}

rank	error
1	0.33794
2	0.14023
3	0.12273
4	0.13689
5	0.12480
10	0.09193
20	0.08616
30	0.08148
32	0.07182
42	0.06819
45	0.05930
86	0.05048
441	0.05048

Table 4.15: Approximation error (4.3) in the original system for the parameter set \mathcal{P}_{xz} and the reference parameter $\hat{\boldsymbol{\mu}} = (\mu_{10}, \mu_{10})$

4.7.2 Local Approximation

Since we have found that a local approximation has a good influence on the results, we will also consider a subdivision for the parameter set \mathcal{P}_{xz} . In Fig. 4.12 you can see that we use nine subsets.



Figure 4.12: Splitting the parameter set \mathcal{P}_{xz}

The red part on the Fig. 4.12 represents the parameter set

$$\mathcal{P}_{tl} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i, j \in \{0, \dots, 3\}\},$$

the yellow one is

$$\mathcal{P}_{ml} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{4, \dots, 16\}, j \in \{0, \dots, 3\}\},$$

and the blue one is

$$\mathcal{P}_{bl} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{17, \dots, 20\}, j \in \{0, \dots, 3\}\}.$$

The white part represents

$$\mathcal{P}_{mm} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i, j \in \{4, \dots, 16\}\},$$

the black one

$$\mathcal{P}_{tm} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{0, \dots, 3\}, j \in \{4, \dots, 16\}\}$$

and the brown one

$$\mathcal{P}_{bm} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{17, \dots, 20\}, j \in \{4, \dots, 16\}\}.$$

Finally, on the right side we have the following three parameter sets, in light purple

$$\mathcal{P}_{tr} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{0, \dots, 3\}, j \in \{17, \dots, 20\}\},$$

in green

$$\mathcal{P}_{mr} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{4, \dots, 16\}, j \in \{17, \dots, 20\}\}$$

and in dark purple

$$\mathcal{P}_{br} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i, j \in \{17, \dots, 20\}\}.$$

After splitting the parameter set, we first consider the results for the parameter set \mathcal{P}_{mm} . Here, $\hat{\boldsymbol{\mu}} = (\mu_{10}, \mu_{10})$ is chosen as the reference parameter. Table 4.16 presents the error values in the original configuration. It is noticeable that the error decreases fast and we can already achieve an acceptable approximation with only 5 basis vectors instead of 169.

Table 4.17 contains the error values (4.3) for the parameter sets \mathcal{P}_{ml} and \mathcal{P}_{tm} . The set \mathcal{P}_{ml} corresponds to the part marked with the color yellow in Fig. 4.12. For this set of parameters we selected the reference parameter $\boldsymbol{\mu} = (\mu_{10}, \mu_3)$ and found that already 4 basis vectors are sufficient to obtain an acceptable error. Furthermore, the parameter set \mathcal{P}_{tm} refers to the part marked with the color black in Fig. 4.12. For this

rank	error
1	0.33794
2	0.11994
3	0.09113
4	0.07459
5	0.05776
21	0.05048
169	0.05048

Table 4.16: Approximation error (4.3) in the original system for the parameter set \mathcal{P}_{mm} and the reference parameter $\hat{\boldsymbol{\mu}} = (\mu_{10}, \mu_{10})$

rank	error	
	\mathcal{P}_{ml}	\mathcal{P}_{tm}
1	0.10774	0.08989
2	0.09191	0.09995
3	0.07043	0.09470
4	0.05818	0.06389
5	0.06030	0.09629
6	0.05749	0.06149
7	0.05895	0.06104
8	0.06056	0.05669
9	0.04930	0.06911
10	0.0577	0.04841
52	0.03470	0.02549

Table 4.17: Approximation error (4.3) in the original system the parameter sets \mathcal{P}_{ml} and \mathcal{P}_{tm} with the reference parameters $\hat{\boldsymbol{\mu}} = (\mu_{10}, \mu_3)$ and $\hat{\boldsymbol{\mu}} = (\mu_3, \mu_{10})$, respectively.

set, the reference parameter $\boldsymbol{\mu} = (\mu_3, \mu_{10})$ was used, and we note that an acceptable error of 0.06389 is already achieved with a rank of 4. We have to note, that we obtained similar results for the parameter sets \mathcal{P}_{bm} and \mathcal{P}_{mr} .

Finally, we turn our focus to the parameter sets \mathcal{P}_{tl} and \mathcal{P}_{bl} . The parameter set \mathcal{P}_{tl} is marked in red and \mathcal{P}_{bl} in blue in Fig. 4.12. As reference parameters we have chosen the parameter $\boldsymbol{\mu} = (\mu_3, \mu_3)$ in \mathcal{P}_{tl} and the parameter $\boldsymbol{\mu} = (\mu_{17}, \mu_3)$ in \mathcal{P}_{bl} . In Table 4.18 is noticeable that the error values decrease slowly in both quantities. Therefore, we decided to split these parameter sets.

rank	error	
	\mathcal{P}_{tl}	\mathcal{P}_{bl}
1	0.14028	0.15792
2	0.10226	0.09353
3	0.11195	0.09908
4	0.09985	0.08068
5	0.09891	0.10568
6	0.07940	0.09854
7	0.07268	0.07756
8	0.06469	0.06362
16	0.01251	0.01114

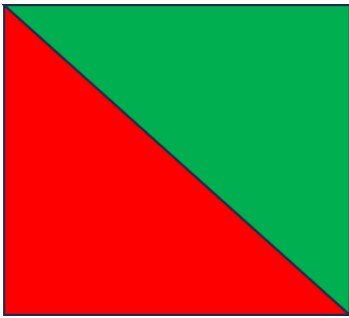
Table 4.18: Approximation error (4.3) in the original system the parameter sets \mathcal{P}_{tl} and \mathcal{P}_{bl} with the reference parameters $\hat{\boldsymbol{\mu}} = (\mu_3, \mu_3)$ and $\hat{\boldsymbol{\mu}} = (\mu_{17}, \mu_3)$, respectively

4.7.3 Refined Local Approximation at a Corner

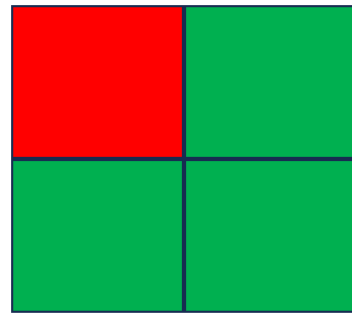
In this section we consider the refined local approximation at the corner according to the approaches in Fig. 4.13. We will present the results for the parameter set \mathcal{P}_{tl} , and for the other parameter sets $(\mathcal{P}_{bl}, \mathcal{P}_{tr}, \mathcal{P}_{br})$ it goes analogously. In the first partition, as shown in Fig. 4.13a, the set \mathcal{P}_{tl} is split into two subsets along the diagonal. The parameter set

$$\mathcal{P}_{tlr}^1 = \{(\mu_1, \mu_0), (\mu_2, \mu_0), (\mu_2, \mu_1), (\mu_3, \mu_0), (\mu_3, \mu_1), (\mu_3, \mu_2)\}$$

is marked in red in Fig. 4.13a. The parameter set \mathcal{P}_{tlg}^1 is the set of remaining parameters in \mathcal{P}_{tl} , i.e. $\mathcal{P}_{tlg}^1 = \mathcal{P}_{tl} \setminus \mathcal{P}_{tlr}^1$ and is shown in green in Fig. 4.13a.



(a) Diagonal splitting of the corner diagonal



(b) Subdivision of the corner

Figure 4.13: Splitting of the corner set \mathcal{P}_{tl}

rank	error	
	\mathcal{P}_{tlg}^1	\mathcal{P}_{tlr}^1
1	0.12770	0.11143
2	0.08475	0.09881
3	0.07928	0.10252
4	0.08004	0.06098
5	0.05415	0.03700
6	0.06029	0.01251
10	0.00945	

Table 4.19: Approximation error (4.3) in the original system the parameter sets \mathcal{P}_{tlg}^1 and \mathcal{P}_{tlr}^1 with the reference parameters $\hat{\boldsymbol{\mu}} = (\mu_3, \mu_3)$

Table 4.19 shows the error (4.3) in the original configuration for the parameter sets \mathcal{P}_{tlr}^1 and \mathcal{P}_{tlg}^1 . We notice that the errors decay quickly compared to the ones in Table 4.18 and 5 basis vectors are enough for us to achieve a good approximation. We should note that the parameter set \mathcal{P}_{tlg}^1 contains 10 and \mathcal{P}_{tlr}^1 contains 6 parameters.

In the second partition, as shown in Fig. 4.13b, \mathcal{P}_{tl} is split into two new subsets: \mathcal{P}_{tlg}^2 and \mathcal{P}_{tlr}^2 . The parameter set

$$\mathcal{P}_{tlr}^2 = \{(\mu_0, \mu_0), (\mu_0, \mu_1), (\mu_1, \mu_0), (\mu_1, \mu_1)\},$$

which are marked in red in Fig. 4.13b. The parameter set \mathcal{P}_{tlg}^2 is the set of remaining points in \mathcal{P}_{tl} , i.e. $\mathcal{P}_{tlg}^2 = \mathcal{P}_{tl} \setminus \mathcal{P}_{tlr}^2$ and they are shown in green in Fig. 4.13b.

Tables 4.20 and 4.21 show us the error (4.3) in original configuration for the second partition of the set \mathcal{P}_{tl} . Compared to Table 4.18, the error decays faster but compared to the first partition, see Table 4.19, the error decays slower. In Table 4.20 we get the error 0.05622 first for rank 7 and in Table 4.21 the error would be acceptable at rank 2.

rank	error
1	0.13405
2	0.11939
3	0.08869
4	0.06259
5	0.06262
6	0.06738
7	0.05622
12	0.01251

Table 4.20: Approximation error (4.3) in the original system for the parameter set \mathcal{P}_{tlg}^2 with the reference parameter $\hat{\boldsymbol{\mu}} = (\mu_3, \mu_3)$

rank	error
1	0.10964
2	0.06082
3	0.07322
4	0.01004

Table 4.21: Approximation error (4.3) in the original system for the parameter set \mathcal{P}_{tlr}^2 with the reference parameter $\hat{\boldsymbol{\mu}} = (\mu_1, \mu_1)$

At the end we can say that the second division, see Fig 4.13b, gives the best results. All in all, we observe that it is hard to approximate the data for the corners of the parameter set by low rank. This does not come as a surprise, as our initial assumption, that the data do not differ except for the translation, does not hold true in these extreme positions of the impactor.

4.7.4 Local Two-step Approximation at the Margins

In this section, we implement the two-step approximation for the parameter sets \mathcal{P}_{ml} , \mathcal{P}_{tm} , \mathcal{P}_{lm} and \mathcal{P}_{rm} in the same way as described in Section 4.5. We hope to reduce the overall rank of our approximations in this way. First, we apply the Empirical Interpolation Method to the parameter set \mathcal{P}_{mm} , and we use the resulting approximation from the middle set as the starting point for the approximation in the parameter sets \mathcal{P}_{ml} , \mathcal{P}_{tm} , \mathcal{P}_{bm} and \mathcal{P}_{rm} .

We denote the snapshot matrix by $S_{mm} \in \mathbb{R}^{N_q \times N_m}$ for the parameter set

$$\mathcal{P}_{mm} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i, j \in \{4, \dots, 16\}\}.$$

The number of the finite elements is N_q and the number of the parameter in the parameter set \mathcal{P}_{mm} is $N_m = 169$. The basis vectors computed for the middle parameter set \mathcal{P}_{mm} are stored in Q_{mm} .

Calculations were performed for all four parameter sets, but we present the results only for the top middle parameter set

$$\mathcal{P}_{tm} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{0, \dots, 3\}, j \in \{4, \dots, 16\}\}$$

in Table 4.22 and for the left middle parameter set

$$\mathcal{P}_{ml} = \{\boldsymbol{\mu} = (\mu_i, \mu_j) | \mu_i \in \mathcal{P}_x, \mu_j \in \mathcal{P}_z \text{ for } i \in \{4, \dots, 16\}, j \in \{0, \dots, 3\}\}$$

in Table 4.23. We use the term $S_{ml} \in \mathbb{R}^{N_q \times N_{ml}}$ for the snapshot matrix that represents the data of the parameter sets \mathcal{P}_{ml} , while $S_{tl} \in \mathbb{R}^{N_q \times N_{tm}}$ is the snapshot matrix that represents the data of the parameter sets \mathcal{P}_{tl} . The parameter sets \mathcal{P}_{tm} and \mathcal{P}_{ml} have the same number of the parameters $N_{ml} = N_{tm} = 52$. The basis vectors are stored in Q_{ml} and Q_{tm} , respectively. In the same way as in Section 4.5, we calculate the final approximations \tilde{S}_{ml} and \tilde{S}_{tm} , respectively, see the equation (4.4).

Top Middle Parameter Set \mathcal{P}_{tm}

We discuss the approximations obtained by applying the two-step method to the top middle parameter set \mathcal{P}_{tm} . In the approximation for the parameter set \mathcal{P}_{mm} , we use $\boldsymbol{\mu}_{10,10} = (\mu_{10}, \mu_{10})$ as our reference parameter $\hat{\boldsymbol{\mu}}_{mm}$. For the approximation in the top middle parameter set \mathcal{P}_{tm} , we select $\boldsymbol{\mu}_{2,10} = (\mu_2, \mu_{10})$ as our reference parameter $\hat{\boldsymbol{\mu}}_{tm}$.

Particularly interesting are the errors (4.3) that occur between the original data, i.e. matrix S_{tl} and the final approximation \tilde{S}_{ml} in the original configuration. The values listed in Table 4.22 illustrate the number of basis vectors from the matrix Q_{mm} in the first column and the number of basis vectors we use for the top middle parameter set \mathcal{P}_{tm} in the second column. In the last column you will find the error (4.3) in the original configuration. In Table 4.17, we can see that we needed four basis vectors for

rank		error
rank for Q_{mm}	rank for Q_{tm}	
1	0	0.10240
1	3	0.08229
1	4	0.06405
1	5	0.06786
2	0	0.11994
2	3	0.08636
2	4	0.06542
2	5	0.06312
3	0	0.12361
3	3	0.07112
3	4	0.07069
3	5	0.06365
4	0	0.12458
4	4	0.07036
4	5	0.06421
5	0	0.11561
5	1	0.09977
5	2	0.10321
5	3	0.07216
5	4	0.09033
5	5	0.06489
5	6	0.06158
5	52	0.02549

Table 4.22: Approximation error (4.3) in the original system in \mathcal{P}_{tm} for the local two-step approximation

the top middle parameter set \mathcal{P}_{tm} in order to obtain a suitable approximation with an error of 0.06389. In Table 4.22 we notice that for a suitable approximation for different numbers of basis vectors for the parameter set \mathcal{P}_{mm} we need at least four basis vectors from the parameter set \mathcal{P}_{tm} . In different combinations of basis vectors from Q_{mm} and Q_{tm} : once one from Q_{mm} and four from Q_{tm} , then four from Q_{mm} and five from Q_{tm} , and finally five from both. In each case, the error is about 0.064. In case we use two basis vectors from Q_{mm} and four basis vectors from Q_{tm} , we get the error of 0.06542. If we take three basis vectors from Q_{mm} and 5 basis vectors from Q_{tm} , the error is 0.06365. From this we can conclude that the local approximation has not brought any particular improvements in this case.

Left Middle Parameter Set \mathcal{P}_{ml}

rank		error
rank for \mathcal{P}_{mm}	rank for \mathcal{P}_{ml}	
1	0	0.09967
1	3	0.06447
1	4	0.06098
1	5	0.06244
2	0	0.09984
2	3	0.06743
2	4	0.06411
2	5	0.05554
3	0	0.10134
3	2	0.07627
3	3	0.06094
3	4	0.05937
4	0	0.10121
4	1	0.09909
4	2	0.06204
4	3	0.06782
5	0	0.10151
5	1	0.09749
5	2	0.06427
5	3	0.07009
5	4	0.06527
5	52	0.03470

Table 4.23: Approximation error (4.3) in the original system in \mathcal{P}_{ml} for the local two-step approximation

Now we discuss the results of the approximations obtained by applying the two-step method to the upper middle parameter set \mathcal{P}_{ml} . In the approximation for the parameter set \mathcal{P}_{mm} , we use the parameter pair $\boldsymbol{\mu}_{10,10} = (\mu_{10}, \mu_{10})$ as our reference values $\hat{\boldsymbol{\mu}}_{mm}$. For the approximation in the left middle parameter set \mathcal{P}_{ml} we set the parameter pair $\boldsymbol{\mu}_{10,2} = (\mu_{10}, \mu_2)$ as our reference points $\hat{\boldsymbol{\mu}}_{ml}$.

Table 4.23 lists the number of basis vectors from the matrix Q_{mm} in the first column and the number of basis vectors that we use for the upper middle parameter set \mathcal{P}_{ml} in the second column. The last column shows the error according to equation (4.3) in the original configuration.

From Table 4.17 we can conclude that we need four basis vectors from the parameter set \mathcal{P}_{ml} for a good approximation with an error of 0.05818. The data from Table 4.23 show that if we take one, two or three basis vectors from Q_{mm} , we need three basis or more vectors in the second set to get a comparable error. If we take four or five basis vectors from Q_{mm} , we need two additional basis vectors to get a similar accuracy. We notice that the two-step approximation has a positive influence on the results in this case. Overall, it remains difficult to predict the benefits of the two-step approximation and the required ranks.

4.8 Evaluation of the failure criteria

In this section, we apply the failure criterion introduced in Section 2.6 to our examples. As the critical value for the failure criterion we set $\varepsilon_c = 0.3$. We consider the entire two dimensional parameter set \mathcal{P}_{xz} and the global approximation, as described in Section 4.7.1.

Number of variances	Number of parameters
0	133
1	146
2	94
3	44
4	17
5	5
6	2

(a) Number of variances for rank 45
for the entire parameter set

Number of variances	Number of parameters
0	74
1	99
2	118
3	81
4	36
5	25
6	4
7	4

(b) Number of variances for rank 30
for the entire parameter set

Table 4.24: Information about computational model of indentation cases

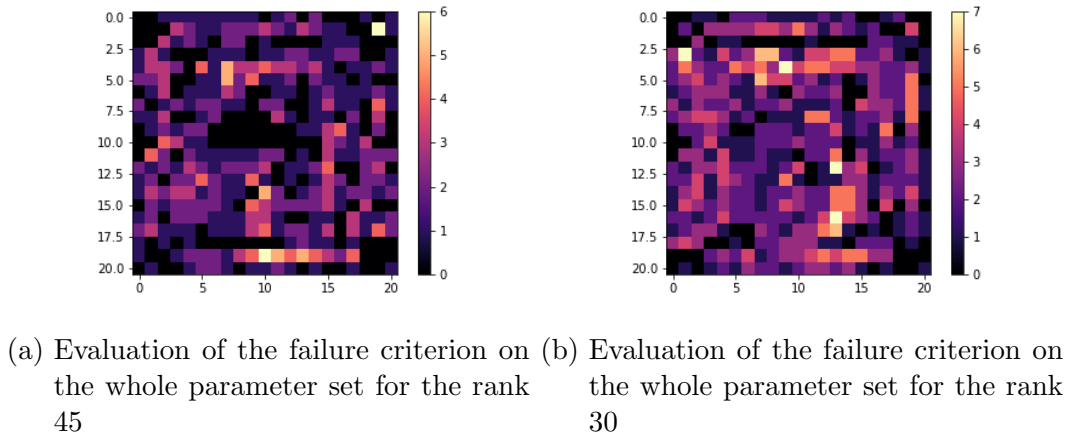


Figure 4.14: Evaluation of the failure criterion

We apply the failure criterion to both the original and the approximated data and compare the number of elements that fulfill the criterion. Table 4.15 provides information that reasonable approximations are achieved by the Empirical Interpolation Method for the ranks 30 and 45. Applying the failure criterion to the data sets for the ranks 30 and 45 allows a direct comparison of the performance of the approximation compared to the original data.

In Table 4.24, the first column shows the number of variances, i.e. number of cells, which are falsely determined as failure or not. The second column shows the number of parameters for which these variances occur. For example, for rank 45, there is a total of 133 parameters for which no deviations were found.

Figure 4.14 shows the variations on a 21×21 plot. Each cell represents a related parameter pair. From these pictures we can see for each parameter pair how many variations occur. We do not recognize a specific pattern.

We continued to focus on the ranks 45, 30 and 5 to determine the number of false negative and false positive cases. The number of false positive cases is the number of parameters for which failure is detected in the approximation but none is detected from the original data. The number of false negative cases is the number of parameters for which failure is detected from the original data but not from the approximation data. As you can see in the first column of Table 4.25, we have tested for different critical values ε_c . The second column of Table 4.25 shows us the number of parameters in the original data for which a failure occurs.

We notice that for different critical values the number of parameters where failure occurs is also different. We notice that the number of false negative and false positive cases depends on the critical value. The value 0.42 is very close to the extremum and in this case we get both false positive and false negative cases for the rank 30. We also note that neither false negative nor false positive cases exist for the approximation with

ε_c	Failure cases	rank 45		rank 30		rank 5	
		false negative	false positive	false negative	false positive	false negative	false positive
0.30	441	0	0	0	0	0	0
0.40	439	0	0	2	0	5	0
0.42	366	0	0	3	1	3	0

Table 4.25: Number of false negative and false positive cases for the ranks 45, 30 and 5

rank 45. For rank 5, we have five false negative cases for critical value 0.40 and three false negative cases for the critical value 0.42, although the approximation error (4.3) is 0.12480, see Table 4.15. In a review of the data related to parameters for which the prediction was wrong, we noticed that the original data sets of two symmetric load cases are not symmetric. We regard this as an indication that there is some noise in the data caused by rough approximation from the simulations. We conclude that we predict most failure cases correctly for quite rough approximations. For a more accurate approximation (rank 45) we predict all failures correctly.

5 Conclusion

The primary objective of the master’s thesis was to develop a method for providing a reduced model for predicting failures in a lithium-ion battery cell model. The empirical interpolation method served as the foundation, combined with other methods throughout the study. All data were initially superimposed, and the empirical interpolation method was applied to the aggregated data. The approximation data were then shifted back to the original configuration. Due to this shift, data were lost, prompting the investigation of two additional approaches: first, replacing the missing data with zeros, and second, reflection of the data. Initially, we analyzed the data concerning the impactor movement along the z -axis. We observed that the reflections provided smaller errors.

The approximation worked for the entire parameter set, but local approximation led to improved results. Furthermore, the two-step approximation enhanced the accuracy of our approximation. At the end we performed the training with half of the parameters and approximated the remaining half using the available data.

In a large test, we considered the case in which we parameterized the translation of the sphere in the x - and z -directions. We also split the parameter set and applied local approximations, with particular attention to the corners. We found that the initial local approximation was insufficient to ensure precise data approximations for load cases at the corners. Therefore, we created new splittings for the related parameter sets and performed refined local approximations specifically at the corners. For the large data set, we also conducted the local two-step approximation. Here the two-step approximation brought improvements for some subsets only. Finally, we evaluated the failure criterion on the original and approximated data and found that we correctly predicted the short circuit also for rather coarse approximation.

We can conclude that the combination of the Empirical Interpolation Method with different approximation methods, the consideration of reflections and an efficient parameterization allows us to predict the short circuit in lithium-ion battery cell models with our reduced model. The finding that local approximations and two-step approximations can lead to better results indicates that the local approach to the problem makes sense.

In view of the obtained results, the thesis offers many possibilities for future studies. In particular, one could consider the reduced models for large parameter sets with different positions and different radii of the impactor. Another possibility would be the construction of reduced models on small sets and in combination of with smarter training techniques for the prediction of approximations for new parameters. Further-

more, we could test the reduction method for a larger finite element model, as we guess that there is some noise in the data of our small model. In our approximation with the Empirical Interpolation Method we try to reduce the error globally. For the evaluation of the failure criterion good approximations of the values close to and above the critical value are relevant. Thus it could be beneficial to focus on such values and the respective area in the approximation with the Empirical Interpolation Method.

We may apply alternative methods for approximating and even completing the data in the reference configuration. For example, we might use only real values for the approximation in a gappy POD and even construct the extensions in an additional loop, see, e.g., [4, 6, 13].

Bibliography

- [1] B. Wieland and U. Simon. Handout: Anisotropie. [Online; accessed October 3, 2022]. URL: https://www.uni-ulm.de/fileadmin/website_uni_ulm/uzwr/mmsm/mmsm1-ws1213/mmsm1-handout-anisotropie.pdf.
- [2] M. Bebendorf, Y. Maday, and B. Stamm. Comparison of some reduced representation approximations. In *Reduced order methods for modeling and computational reduction*, volume 9 of *MS&A. Model. Simul. Appl.*, pages 67–100. Springer, Cham, 2014. doi:10.1007/978-3-319-02090-7_3.
- [3] C. Breitfuss, P. Kolm, K. Ismail, W. Leitgeb, and S. Kirschbichler. Meso-mechanical battery crash model. Project report, unpublished, 2019.
- [4] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. In *21st AIAA Applied Aerodynamics Conference*, number AIAA 2003-4213, pages 1–11, 2012. doi:10.2514/6.2003-4213.
- [5] N. Cagniard, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects. In *Contributions to partial differential equations and applications*, volume 47 of *Comput. Methods Appl. Sci.*, pages 131–150. Springer, Cham, 2019.
- [6] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 242:623–647, 2013. doi:10.1016/j.jcp.2013.02.028.
- [7] T. Chen, Y. Jin, H. Lv, A. Yang, M. Liu, B. Chen, Y. Xie, and Q. Chen. Applications of lithium-ion batteries in grid-scale energy storage systems. *Transactions of Tianjin University*, 26:208–217, 2020. doi:10.1007/s12209-020-00236-w.
- [8] Y.-S. Chen. Testing and modeling tensile stress-strain curve for prestressing wires in railroad ties. Master’s thesis, Department of Civil Engineering, College of Engineering, Kansas State University, 2016. URL: <http://hdl.handle.net/2097/32582>.
- [9] C. Cruz and E. Miranda. A critical review of the rayleigh damping model. In *16th World Conference on Earthquake*, number 2674, pages 1–12, 2017. URL: <https://www.wcee.nicee.org/wcee/article/16WCEE/WCEE2017-2674.pdf>.

- [10] Digi-Key. A Designer's Guide to Lithium (Li-ion) Battery Charging , 2016. [Online; accessed September 27, 2022]. URL: <https://www.digikey.be/nl/articles/a-designer-guide-fast-lithium-ion-battery-charging>.
- [11] L. Dörmann, K. Sann-Ferro, P. Heiniger, and J. Mähliß. *Kompendium: Li-Ionen-Batterien, Grundlagen, Merkmale, Gesetze und Normen*. VDE Verband der Elektrotechnik Elektronik Informationstechnik e. V., Frankfurt, 2021. URL: <https://www.dke.de/resource/blob/933404/fa7a24099c84ef613d8e7afd2c860a39/kompendium-li-ionen-batterien-data.pdf>.
- [12] A. Dumon, M. Andres, S. Menegazzi, C. Breitfuss, C. Jilmenez, F. Chinesta, F. Daim, and A. Tramecon. AI enhanced methods for virtual prediction of short circuit in full vehicle crash scenarios. In *Proceedings of WCX SAE World Congress Experience*, number 2020-01-0950, 2020. doi:10.4271/2020-01-0950.
- [13] R. Everson and L. Sirovich. Karhunen–Loève procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, 1995. doi:10.1364/JOSAA.12.001657.
- [14] M. Grepl, Y. Maday, N. Nguyen, and A. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: M2AN*, 41(3):575–605, 2006. doi:doi:10.1051/m2an:2007031.
- [15] D. Gross, W. Hauger, and P. Wriggers. *Technische Mechanik 4*. Springer, Heidelberg Dordrecht London New York, 2011. doi:10.1007/978-3-642-16828-4.
- [16] C. Großmann, H.-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equation*. Springer, Berlin Heidelberg, 2007.
- [17] J. O. Hallquist. *LS-DYNA Theory Manual*, 2006. URL: https://ftp.lstc.com/anonymous/outgoing/jday/manuals/ls-dyna_theory_manual_2006.pdf.
- [18] J. O. Hallquist. *LS-DYNA, Keyword user's manual, Volume II*, 2020. [Online; accessed February 22, 2022]. URL: https://www.dynasupport.com/manuals/ls-dyna-manuals/ls-dyna_manual_volume_ii_r12.pdf.
- [19] H. H. Heimes, A. Kampker, C. Lienemann, M. Locke, C. Offermanns, S. Michaelis, and E. Rahimzei. *Lithium-Ion battery cell production process*. PEM of RWTH Aachen and VDMA, 2019. URL: https://www.pem.rwth-aachen.de/global/show_document.asp?id=aaaaaaaaabdqbtqk.
- [20] J. S. Hesthaven, C. Pagliantini, and G. Rozza. Reduced basis methods for time-dependent problems. *Acta Numer.*, 31:265–345, 2022. doi:10.1017/S0962492922000058.
- [21] B. Hoegel. Orthotropie, 2020. [Online; accessed July 13, 2022]. URL: <http://www.biancahoegel.de/material/const/orthotropie.html>.
- [22] M. Jung and U. Langer. *Methode der finiten Elementen für Ingenieure*. Teubner, Stuttgart-Leipzig-Wiesbaden, 2001.

- [23] R. Korthauer. *Handbuch Lithium-Ionen-Batterien*. Springer Vieweg, Berlin Heidelberg, 2013. doi:10.1007/978-3-642-30653-2.
- [24] LS-DYNA Support. Solving problems with path dependent materials. [Online; accessed January 30, 2023]. URL: <https://www.dynasupport.com/tutorial/computational-plasticity/solving-problems-with-path-dependent-materials>.
- [25] LS-DYNA Support. Time integration. [Online; accessed February 7, 2023]. URL: <https://www.dynasupport.com/tutorial/ls-dyna-users-guide/time-integration>.
- [26] Y. Maday, N. Nguyen, A. Patera, and G. Pau. A general, multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8(1):383–404, 2008. doi:10.3934/cpaa.2009.8.383.
- [27] Museum für Energiegeschichte(n). Die Entdeckung der Elektrizität Bernstein, Blitz und Batterie. URL: https://www.energiegeschichte.de/content/dam/revu-global/energiegeschichte/images/BilderNeu/DieAusstellung/Sonderausstellungen/BernsteinBlitzundBatterie/BBB_Begleitheft.pdf.
- [28] F. Osmak. Usporedba razlicitih teorija plasticnosti u analizi postupaka elasto plasticnog oblikovanja valjcima. Undergraduate thesis / Završni rad, Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučiliste u Zagrebu, Fakultet strojarstva i brodogradnje, 2017. URL: <http://repozitorij.fsb.hr/8061/>.
- [29] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Method for Partial Differential Equations*. Springer International Publishing, Switzerland, 2016. doi:10.1007/978-3-319-15431-2.
- [30] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente*. Teubner, Stuttgart-Leipzig-Wiesbaden, 2003. doi:10.1007/978-3-322-80054-1.
- [31] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM J. Sci. Comput.*, 39(4):A1225–A1250, 2017. doi:10.1137/16M1059904.
- [32] G. Welper. Transformed snapshot interpolation with high resolution transforms. *SIAM J. Sci. Comput.*, 42(4):A2037–A2061, 2020. doi:10.1137/19M126356X.
- [33] C. Yuan, L. Wang, S. Yin, and J. Xu. Generalized separator failure criteria for internal short circuit of lithium-ion battery. *Journal of Power Sources*, 467:228–360, 2020. doi:10.1016/j.jpowsour.2020.228360.

