

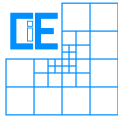
Faculty of Civil Engineering and Geodesy
Chair for Computation in Engineering
Prof. Dr. rer. nat. Ernst Rank

Acoustic source localization based on time reversed simulations with high order Finite-Element Methods

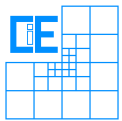
Matthias Benedikt Miksch

Bachelor's thesis
for the Bachelor of Science program Engineering Science

Author: Matthias Benedikt Miksch
Matriculation number: 03620765
Supervisor: Prof. Dr. rer. nat. Ernst Rank
Nils Zander, M.Sc.
Date of issue: 01. June 2014
Date of submission: 30. September 2014



Involved Organisations



Chair for Computation in Engineering
Faculty of Civil Engineering and Geodesy
Technische Universität München
Arcisstraße 21
D-80333 München

Declaration

With this statement I declare, that I have independently completed this Bachelor's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

München, September 29, 2014

Matthias Benedikt Miksch

Matthias Benedikt Miksch
Ungererstraße 58
D-80805 München
e-Mail: matze-miksch@gmx.de

Abstract

This bachelor's thesis depicts the source localization of acoustic source in solid bodies. For this purpose two fundamental equations are introduced, the equation of motion and the wave equation. Both equations were discretized in space with the Finite Element Method and in time with the Newmark's scheme. Beside the simulation of a wave that spreads out in space, it is possible to simulate the wave in time-reverse such that the wave propagates back to its origin. The equations were solved numerically within MATLAB. Therefore an existing finite element implementation, which was provided by the chair of Computation in Engineering of the Technische Universität München, was extended.

Kurzzusammenfassung

Diese Bachelorarbeit befasst sich mit der Lokalisierung von akustischen Quellen in Festkörpern. Hierfür werden zwei grundlegende Gleichungen vorgestellt, die Bewegungsgleichung und die Wellengleichung. Beide Gleichungen werden mithilfe der Finite Elemente Methode und dem Newmark Schema diskretisiert. Neben der Simulation einer Welle, die sich im Raum ausbreitet ist, es genau so möglich diese Welle zeit-invers zu simulieren, sodass sich die Welle zurück zu ihren Ausgangspunkt ausbreitet. Die Gleichungen wurden numerisch mithilfe des Programmes MATLAB gelöst. Hierfür wurde eine existierende Finite Elemente Implementierung erweitert, die vom Lehrstuhl für Computation in Engineering an der Technischen Universität München bereitgestellt wurde.

Contents

1	Introduction and Motivation	1
2	Fundamental concepts	3
2.1	The Equation of Motion for linear elasto-dynamics	3
2.1.1	Derivation of the strong form of the equation of motion	3
2.1.2	Derivation of the weak form of the equation of motion	7
2.2	The Wave Equation	9
2.2.1	Derivation of the strong form of the wave equation	9
2.2.2	Derivation of the weak form of the wave equation	10
2.3	Discretization	12
2.3.1	Finite elements ansatz	12
2.3.2	The Newmark's scheme	19
2.4	Time-reversed simulation	22
3	Simulations	23
3.1	Generation of different simulation setups	23
3.1.1	Simulation setups with the equation of motion	23
3.1.2	Simulation setups with the wave equation	24
3.2	Simulations with the different boundary condition setups	26
3.2.1	Convergence study	27
3.3	Reproduction of an experiment	30
3.3.1	Description of the experiment	30
3.3.2	Simulation of the experiment	32
3.3.3	Observations from the simulation of the experiment	34
3.4	Influences of the boundary on the reflected wave	36
3.5	Time-reversed simulations	38
3.5.1	Time-reversed simulation with the equation of motion	38
3.5.2	Time-reversed simulation with the wave equation	40
4	Conclusions	43
A	Sensor signals of the experiment E2 and synthetic sensor signals	45
B	List of implemented functionalities	55
C	Compact disc	57

Chapter 1

Introduction and Motivation

Seismologists often want to investigate wave propagation to estimate the location of the disturbance that caused the earthquake. For this application the wave is traced back to its source. In general a reversed wave propagation is also an equally wave propagation phenomenon with the exception that time is running backwards. If seismologists are able to reproduce the wave, they can also apply a time-reversed simulation to locate the epicenter of the wave.

The modeling of this problem involves the computation of solutions from partial differential equations which in most cases cannot be solved analytically. An ansatz that leads to a numerical description of the mechanical problem is the high order Finite Element Method (high order FEM). This approach is used especially for issues in solid mechanics where the engineers can investigate the behavior of the material under some load.

The Finite Element Method can be applied on wide range on application. In this thesis this method is used to investigate wave propagation in solid bodies. In general, a wave can propagate in every medium, but what defines a wave?

In [1], a wave is described as a “disturbance or oscillation that travels through matter or space, accompanied by a transfer of energy”. One of the best known wave propagation phenomenon is an earthquake. From geography basis knowledge, it is known that the Earth’s crust consists of several tectonic plates that are in constant motion. Sometimes those plates chafe and trigger a disturbance. The propagation of this disturbance through the Earth’s crust until its surface is what is called an earthquake.

Before the investigation of the wave propagation, a model that describes the wave is needed. This thesis depicts two different mathematical models that are able to describe wave propagation in solid bodies.

The first model to simulate is the equation of motion. It is possible to describe the wave knowing the displacementfield in the medium. In this case the fundamental equation is the equation of motion. In a solid body neighboring points can not change theirs positions unless the material fails. This fact implies that the movement of a material point is transmitted to the neighboring points. Finally, the wave travels through the medium.

The second model observes the volumetric change in the medium. Imagine that the medium consists of many small volumes that are linked together with no gaps. A disturbance in the size of one of these volumes affects also the neighboring volumes, as they are linked close together. In this case the wave propagation is described by the wave equation.

Returning to the example of the earthquake you can simulate the earthquake with both equations. In this thesis, the Finite Element Method is applied on both mathematical models, the equation of motion and the wave equation. Furthermore the forward wave propagation and a time reversed simulation are performed with both models.

Chapter 2

Fundamental concepts

In this chapter, the basic concepts for simulating wave propagation phenomena are introduced. There are two possible ways to describe wave propagation. One can solve the equation of motion for elasto-dynamics where the displacement field is calculated. Another possibility is to solve the wave equation. This approach results in a scalar field that describes the volumetric dilatation. Furthermore, the finite element ansatz for spatial discretization and a time integration scheme are explained. At the end of the chapter, the concept of time-reversed simulations is described.

2.1 The Equation of Motion for linear elasto-dynamics

In this section the equation of motion for linear elasto-dynamics in terms of the displacement is derived. The derivations are extracted from the lecture notes “Computational Solid Dynamics” written by Wall and Popp [8].

2.1.1 Derivation of the strong form of the equation of motion

As Newton’s Second Law of motion testifies, the sum of the forces acting on a body has to be equal to the product of the acceleration and the mass of the body:

$$\sum_i F_i = m \cdot a .$$

To get an expression for the forces acting on a body, an infinitesimal small cube is considered which represents a material point in a solid body, see figure 2.1. If we want to balance the forces acting on that cube, the stress resultants have to be balanced. Be aware that the stresses σ are forces per area. The first index indicates the orientation of the area and the second index indicates the direction of the stress.

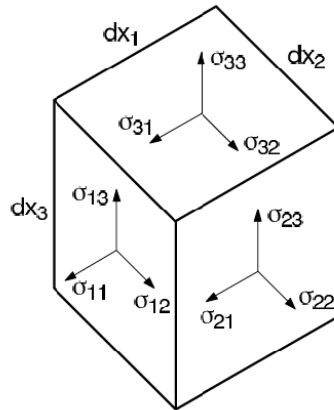


Figure 2.1: Stress resultants of an infinitesimal volume [8]

Given that the material point is not accelerated and considering only the components in x_1 -direction leads to the following expression

$$\begin{aligned}
 & \left(\sigma_{11} + \frac{\partial \sigma_{11}}{\partial x_1} dx_1 - \sigma_{11} \right) dx_2 dx_3 \\
 & + \left(\sigma_{21} + \frac{\partial \sigma_{21}}{\partial x_2} dx_2 - \sigma_{21} \right) dx_1 dx_3 \\
 & + \left(\sigma_{31} + \frac{\partial \sigma_{31}}{\partial x_3} dx_3 - \sigma_{31} \right) dx_1 dx_2 \\
 & \quad + \hat{b}_1 dx_1 dx_2 dx_3 \\
 & = 0 ,
 \end{aligned}$$

Figure 2.2: Stress balance in x-direction [8]

where \hat{b}_1 is the x_1 -component of the bodyload (forces per volume).

Now, the previous equation is divided by $dx_1 dx_2 dx_3$ which yields the linear momentum balance equation in x_1 -direction:

$$\frac{\partial \sigma_{11}}{\partial x_1} + \frac{\partial \sigma_{21}}{\partial x_2} + \frac{\partial \sigma_{31}}{\partial x_3} + \hat{b}_1 = 0 .$$

Writing the balance equation for all three directions in index notation and compact tensor notation will lead to the balance equation of elasto-statics:

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \hat{b}_i = 0 \quad \text{for } i, j = 1, 2, 3 \quad \text{or} \quad \nabla \underline{\underline{\sigma}} + \underline{\underline{\hat{b}}} = \underline{\underline{0}} . \quad (2.1)$$

If the internal stresses does not sum up to zero the infinitesimal volume must have an acceleration:

$$\nabla \underline{\underline{\sigma}} + \underline{\underline{\hat{b}}} = \rho \underline{\underline{\ddot{u}}} . \quad (2.2)$$

This is the equation of motion for elasto-dynamics. Note that this equilibrium holds for an infinitesimal volume which means that it balances the forces per volume. To receive an actual forces we have to integrate over the volume of the solid we are interested in.

In the next step, the stresses are defined precisely. Considering linear elastic material, the stresses are linearly related to the strains:

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} \quad \text{or} \quad \underline{\underline{\sigma}} = \mathbf{C} : \underline{\underline{\varepsilon}} . \quad (2.3)$$

This equation is the Constitutive equation (**CE**) where C_{ijkl} is a fourth-order tensor which is called the elastic moduli. It satisfies the following symmetries $C_{ijkl} = C_{jikl} = C_{klij} = C_{ijlk}$ due to the symmetries of stresses and strains.

If the material is completely isotropic the elastic moduli can be expressed with only two independent parameters, the Young's modulus E and the Poisson's ratio ν :

$$C = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ & 1-\nu & \nu & 0 & 0 & 0 \\ & & 1-\nu & 0 & 0 & 0 \\ & & & \frac{1-2\nu}{2} & 0 & 0 \\ & sym. & & & \frac{1-2\nu}{2} & 0 \\ & & & & & \frac{1-2\nu}{2} \end{bmatrix} .$$

An equivalent description of the elastic moduli with the Lamé parametes is the following expression:

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (2.4)$$

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)} , \quad (2.5)$$

As a last step, the kinematic equation (**KE**) is introduced which relates the strains to the displacement:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \quad \text{or} \quad \underline{\underline{\varepsilon}} = \frac{1}{2}(\nabla \underline{u} + (\nabla \underline{u})^T) . \quad (2.6)$$

Inserting the **CE** (2.3) and the **KE** (2.6) into the equation of motion (2.2) leads to the equation of motion in terms of the displacements:

$$\rho \ddot{u}_i = \left(C_{ijkl} \frac{1}{2}(u_{k,l} + u_{l,k}) \right)_{,j} + \hat{b}_i \quad \text{or} \quad \rho \ddot{\underline{u}} = \nabla \cdot \left(\mathbf{C} : \frac{1}{2}(\nabla \underline{u} + (\nabla \underline{u})^T) \right) + \hat{\underline{b}} .$$

This is the equation of motion in terms of the displacements.

In the following a whole solid body is considered. Inside the body the previously derived equation hold. To get an unique solution it is necessary to specify boundary condition on the outer surfaces Γ and initial conditions in the whole body Ω .

Initial condition

We can calculate the acceleration with the equation of motion (2.2). To deduce the velocity and the displacement we have to integrate twice which leads to two unknown integration constants. These unknowns are estimated with initial conditions. This implies that the displacement and the velocity has to be known in the whole domain at the starting time t_0 :

$$\underline{u}(\underline{x}, t_0) = \underline{u}_0 \quad \text{in } \Omega$$

$$\dot{\underline{u}}(\underline{x}, t_0) = \dot{\underline{u}}_0 \quad \text{in } \Omega .$$

With the above equations, it is possible to determine the unknown integration constants.

Boundary conditions

In general, the computation domain is limited and it is necessary to specify what happens on the boundary Γ of the domain Ω . There are two different types of boundary condition.

At first the Dirichlet boundary condition (**DBC**) that specifies the solution on the surface:

$$\underline{u} = \hat{\underline{u}} \quad \text{or} \quad u_i = \hat{u}_i \quad \text{on } \Gamma_u \times [t_0, t_E] . \quad (2.7)$$

In case of the equation of motion the displacement field is specified directly.

The second boundary condition specifies the derivative of the solution on the boundary. In this case a Neumann boundary condition (**NBC**) is applied on the surface:

$$\underline{\underline{\sigma}} \cdot \underline{\underline{n}} = \underline{\underline{\hat{t}}} \quad \text{or} \quad \sigma_{ij}n_j = \hat{t}_i \quad \text{on } \Gamma_\sigma \times [t_0, t_E] . \quad (2.8)$$

In case of the equation of motion a **NBC** relates to a force that is acting on the surface.

Be aware that boundary condition should be defined on the whole surface and that either DBC or NBC are specified:

$$\Gamma_u \cup \Gamma_\sigma = \Gamma \quad \text{and} \quad \Gamma_u \cap \Gamma_\sigma = \emptyset .$$

If there isn't explicitly specified any boundary condition the natural boundary condition is applied automatically, which is a **NBC** with $\hat{t} = 0$. Furthermore the applied boundary conditions should not be contradictory to the initial conditions.

2.1.2 Derivation of the weak form of the equation of motion

The goal is to solve the equation of motion. In general, it is not efficient and many times also impossible to calculate an analytical solution with the strong formulation of our problem. For numerical simulations, it is sufficient to calculate an approximation of the exact solution. Solving the weak formulation of the equation will not yield the exact solution, but will be a satisfactory approximation if the problem is well-posed and the discretization fulfills several requirements.

The equation of motion (2.2) can be formulated in terms of its residual. The residual becomes zero if the exact solution of the problem is inserted:

$$\underline{\underline{R}}_{BE} = \nabla \cdot \underline{\underline{\sigma}} + \underline{\underline{\hat{b}}} - \rho \underline{\underline{\ddot{u}}} \quad \text{or} \quad R_{BEi} = \sigma_{ij,j} + \hat{b}_i - \rho \ddot{u}_i \quad \text{in } \Omega \times [t_0, t_E] .$$

The residual is multiplied with a weighting function in space $\underline{\underline{\omega}}(\underline{\underline{x}})$ and then is integrated over the whole domain of the body:

$$\int_{\Omega} \left(\underbrace{\sigma_{ij,j} + \hat{b}_i - \rho \ddot{u}_i}_{R_{BEi}} \right) \cdot \omega_i d\Omega = 0$$

$$\int_{\Omega} \sigma_{ij,j} \omega_i d\Omega + \int_{\Omega} \hat{b}_i \omega_i d\Omega - \int_{\Omega} \rho \ddot{u}_i \omega_i d\Omega = 0 .$$

Subsequently, the first integral is integrated by part which leads to the following expression:

$$\int_{\Gamma} \sigma_{ij}n_j \omega_i d\Omega - \int_{\Omega} \sigma_{ij} \omega_{i,j} d\Omega + \int_{\Omega} \hat{b}_i \omega_i d\Omega = \int_{\Omega} \rho \ddot{u}_i \omega_i d\Omega .$$

The integral over the whole surface Γ can be split into the integral over the Dirichlet boundary Γ_u and the Neumann boundary Γ_σ :

$$\int_{\Gamma_u} \sigma_{ij} n_j \omega_i d\Gamma + \int_{\Gamma_\sigma} \sigma_{ij} n_j \omega_i d\Gamma - \int_{\Omega} \sigma_{ij} \omega_{i,j} d\Omega + \int_{\Omega} \hat{b}_i \omega_i d\Omega = \int_{\Omega} \rho \ddot{u}_i \omega_i d\Omega .$$

Imposing the condition that ω equal to zero on the Dirichlet boundary, vanishes the first integral. Furthermore, one inserts the **NBC** (2.8) in the integral over the Neumann boundary and one reorders all terms:

$$\int_{\Omega} \rho \ddot{u}_i \omega_i d\Omega + \int_{\Omega} \sigma_{ij} \omega_{i,j} d\Omega = \int_{\Gamma_\sigma} \hat{t}_i \omega_i d\Gamma + \int_{\Omega} \hat{b}_i \omega_i d\Omega .$$

Finally the **CE** (2.3) and the **KE** (2.6) are inserted to receive the weak formulation in terms of the displacement field:

$$\int_{\Omega} \underline{\omega}^T \rho \underline{\ddot{u}} d\Omega + \int_{\Omega} (\mathbf{L}\underline{\omega})^T \underbrace{\mathbf{C}}_{\underline{\varepsilon}} \underbrace{\mathbf{L}\underline{u}}_{\underline{\sigma}} d\Omega = \int_{\Gamma_\sigma} \underline{\omega}^T \underline{\hat{t}} d\Gamma + \int_{\Omega} \underline{\omega}^T \underline{\hat{b}} d\Omega . \quad (2.9)$$

\mathbf{L} is an appropriate differential operator in space. For solving for the displacements in the three spatial directions \mathbf{L} has the following structure:

$$\mathbf{L} := \begin{pmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{pmatrix} .$$

The weak formulation of the problem is finding an \underline{u} from suitable function space U such that (2.9) holds for any ω from another suitable function space W .

A function space is a set of functions with the properties that the sum of two elements from the set is an element of the set and that the product of a scalar and an element is an element of the set.

If the functionspaces U and W would include all possible functions, the exact solution would be found.

2.2 The Wave Equation

In this section the second mathematical model is introduced. The wave equation solves for the dilatation in a linear elastic continuum.

2.2.1 Derivation of the strong form of the wave equation

In this section the strong form of the wave equation is derived from the strong form of the equation of motion.

At first the stresses can be written in terms of the displacements, inserting equations (2.4) and **KE** (2.6) into the **CE** (2.3):

$$\underline{\underline{\sigma}} = \lambda(\nabla \cdot \underline{u})\underline{\underline{1}} + 2\mu \left[\nabla \underline{u} + (\nabla \underline{u})^T \right] . \quad (2.10)$$

The equation of motion in absence of body forces ($\tilde{\underline{b}} = 0$) simplifies to:

$$\nabla \cdot \underline{\underline{\sigma}} = \rho \ddot{\underline{u}} . \quad (2.11)$$

Inserting equation (2.10) in equation (2.11) yields:

$$\nabla \lambda(\nabla \cdot \underline{u}) + \nabla \mu \left[\nabla \underline{u} + (\nabla \underline{u})^T \right] + (\lambda + 2\mu)\nabla(\nabla \cdot \underline{u}) - \mu \nabla \times \nabla \times \underline{u} = \rho \ddot{\underline{u}} .$$

Under the assumption that the gradients of the material parameters λ and μ are negligible:

$$(\lambda + 2\mu)\nabla(\nabla \cdot \underline{u}) - \mu \nabla \times \nabla \times \underline{u} = \rho \ddot{\underline{u}} . \quad (2.12)$$

Now, the displacement \underline{u} is decomposed with the Helmholtz decomposition. According to Helmholtz, every vector field can be decomposed in a solenoidal and a rotation free part, where the rotation free part is the gradient of a scalar potential [5]:

$$\underline{u} = \underbrace{\nabla \psi}_{\nabla \times (\nabla \psi) = 0} + \underbrace{\nabla \times \underline{v}}_{\nabla \cdot (\nabla \times \underline{v}) = 0} . \quad (2.13)$$

In the next step the scalar Φ is defined as the divergence of the displacement \underline{u} . The scalar Φ describes the volumetric dilatation and is related to the scalar ψ :

$$\Phi = \nabla \cdot \underline{u} = \nabla \cdot (\nabla \psi) + \underbrace{\nabla \cdot (\nabla \times \underline{v})}_{=0} = \Delta \psi .$$

With this relation the simplified equation of motion (2.12) can be written as the wave equation for compression waves:

$$\left(\frac{\lambda + 2\mu}{\rho}\right) \Delta\Phi = \ddot{\Phi} . \quad (2.14)$$

This equation only describes compression waves, also called primary waves.

Defining $\underline{\gamma}$ as the curl of the displacement \underline{u} leads to another wave equation:

$$\underline{\gamma} = \nabla \times \underline{u} = \underbrace{\nabla \times (\nabla\psi)}_{=0} + \nabla \times (\nabla \times \underline{v}) = -\Delta \underline{v} ,$$

where $\underline{\gamma}$ is related to the solenoidal part of the displacement field. Inserting this definition in equation (2.12) yields the wave equation for shear waves (secondary waves):

$$\left(\frac{\mu}{\rho}\right) \Delta\underline{\gamma} = \ddot{\underline{\gamma}} . \quad (2.15)$$

The wave equations (2.14) and (2.15) look similar except for the prefactor and dimension of the solution field. The prefactor is the wave propagation speed. Since λ is greater than zero, shear waves are slower than compression waves. In the further course of this thesis, the wave equation for shear waves is no longer considered.

Φ in equation (2.14) is a scalar that describes the volumetric change at position x and time t . The volumetric change depends on the normal strains, which are the diagonal elements of the strain tensor (2.6). These strains depend only on the rotation free part of the displacement. The other strains do influence the shape of the volume but have no effect on the magnitude of the volumetric dilatation.

The wave propagation speed in equation (2.14) is named α :

$$\alpha = \sqrt{\frac{\lambda + 2\mu}{\rho}} .$$

This leads to the common representation of the wave equation for compression waves:

$$\ddot{\Phi} = \alpha^2 \Delta\Phi . \quad (2.16)$$

2.2.2 Derivation of the weak form of the wave equation

In section 2.1.2 we have derived the weak solution of the equation of motion. The same procedure is applied on the wave equation. Be aware that compared to the equation of motion, the wave equation solves for a scalar not for a vector field. The derivation is extracted from [3].

The previously derived strong form of the wave equation is multiplied with a testfunction ω and integrate over the whole domain:

$$\int_{\Omega} \ddot{\Phi} \cdot \omega \, d\Omega = \int_{\Omega} \alpha^2 \Delta \Phi \cdot \omega \, d\Omega .$$

The second integral is integrated by part which yields:

$$\int_{\Omega} \ddot{\Phi} \cdot \omega \, d\Omega = \int_{\Gamma} \alpha^2 (\nabla \Phi \cdot \underline{n}) \omega \, d\Gamma - \int_{\Omega} \alpha^2 \nabla \Phi \nabla \omega \, d\Omega .$$

The boundary of the domain Γ is split into two parts where boundary conditions as DBC (2.7) and NBC (2.8) are applied:

$$\int_{\Gamma} \alpha^2 (\nabla \Phi \cdot \underline{n}) \omega \, d\Gamma = \int_{\Gamma_u} \alpha^2 (\nabla \Phi \cdot \underline{n}) \omega \, d\Gamma + \int_{\Gamma_\sigma} \underbrace{\alpha^2 (\nabla \Phi \cdot \underline{n})}_{\tilde{t}} \omega \, d\Gamma .$$

In the next step it is assumed that the weighting functions are zero on the Dirichlet boundary Γ_u . Additionally, the Neumann condition on Γ_σ is inserted which leads to the weak formulation of the wave equation:

$$\int_{\Omega} \ddot{\Phi} \cdot \omega \, d\Omega + \int_{\Omega} \alpha^2 \nabla \Phi \nabla \omega \, d\Omega = \int_{\Gamma_\sigma} \alpha^2 \tilde{t} \omega \, d\Gamma .$$

Notice that equation (2.17) looks similar to the weak formulation of the equation of motion (2.9) if the gradient is replaced by an differential operator $\bar{\mathbf{L}}$:

$$\int_{\Omega} \ddot{\Phi} \cdot \omega \, d\Omega + \int_{\Omega} \bar{\mathbf{L}} \Phi \alpha^2 \bar{\mathbf{L}} \omega \, d\Omega = \int_{\Gamma_\sigma} \alpha^2 \tilde{t} \omega \, d\Gamma . \quad (2.17)$$

The gradient can be expressed with the differential operator $\bar{\mathbf{L}}$ that differentiates the shape-functions to the global coordinates:

$$\bar{\mathbf{L}} := \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} .$$

Notice that the L operator looks different than in the weak formulation of the equation of motion (2.1.2).

The weak formulation of the wave equation is finding Φ from a suitable function space such that equation (2.17) holds for any ω from another suitable function space.

2.3 Discretization

2.3.1 Finite elements ansatz

This section explains the general finite element approach for the spatial discretization of the problem. These explanations are based on the lecture notes “Computational Solid Dynamics” [8] and “Algorithms of partial differential equations” [6] that were visited during the winter semester 2013/2014. Additionally, the paper “FCMLab: A finite cell research toolbox for MATLAB” [10] were aduced.

The aim is to discretize the displacement u and the weighting functions ω from the weak formulations of the equation of motion. Similarly the volumetric dilatation Φ and is discretized in the wave equation. The desired function spaces are therefore spanned by a selection of basis functions \mathbf{N} . Following a Bubnov-Galerkin approach the same basis functions are used for the displacement and the weighing function:

$$u = \mathbf{N}(x) \tilde{u}(t) \tag{2.18}$$

$$\Phi = \mathbf{N}(x) \tilde{\Phi}(t) \tag{2.19}$$

$$\omega = \mathbf{N}(x) \tilde{\omega}(t) . \tag{2.20}$$

The same basis functions are used in every timestep, therefore notice that these are independent of time and only the coefficients \tilde{u} , $\tilde{\omega}$ and $\tilde{\Phi}$ depend on time.

Local support

For a finite element approach, the domain Ω is subdivided into multiple equally spaced domains $\Omega^{(e)}$. The basis functions are chosen such that they only have local support. This means each basis function is non-zero on a defined interval in Ω and zero everywhere else.

This ansatz is very useful because now it is possible to choose the basis functions such that locally on every element the same functions are used. As shown in figure 2.3, the domain Ω is divided into equally spaced finite elements. The global basis are identical if these are transformed to local coordinates on each element. Proceeding, it is sufficient to define local basis functions. For this purpose local coordinates $r, s, t \in [-1, 1]$ are defined.

The index in brackets identifies the element, see figure 2.3. The selection of these basis functions characterizes the finite element approach.

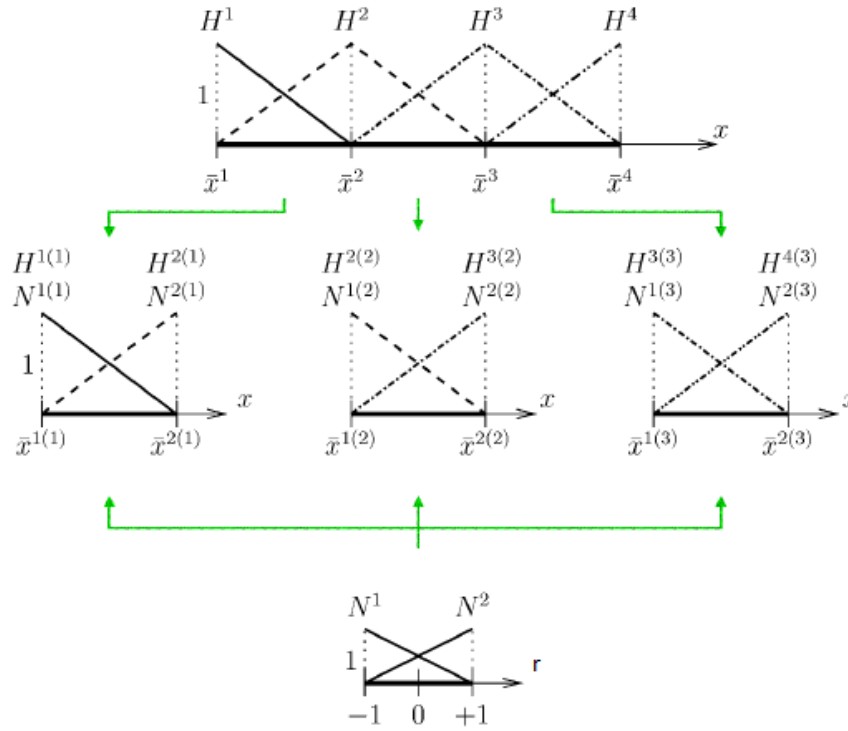


Figure 2.3: Decomposition of the domain into equally finite element, each element is discretized with two linear basis functions [8]

High order FEM

The basic component of the high order Finite Element Method are the 1D basis functions, also called 1D-modes (see figure 2.4). These are defined as integrated Legendre polynomials [7].

Depending on how accurate the solution should be approximated, one choose as many 1D basis functions until a desired polynomial degree is reached. Finally, these 1D-modes are combined to receive 2D- and 3D-modes:

$$N_{i,j}^{2D}(r, s) = N_i^{1D}(r) N_j^{1D}(s) .$$

The combination of two linear modes results in a bi-linear nodal-mode (2.5a). This mode is specified by the node where it is non-zero. Combining a linear mode with a higher order mode results in an edge-mode (2.5b). The only combination left is two higher order 1D-modes which leads to a 2D internal mode (2.5c).

Exactly the same way you combine the 2D-modes to obtain the 3D-modes:

$$N_{i,j,k}^{3D}(r, s, t) = N_{i,j}^{2D}(r, s) N_k^{1D}(t) .$$

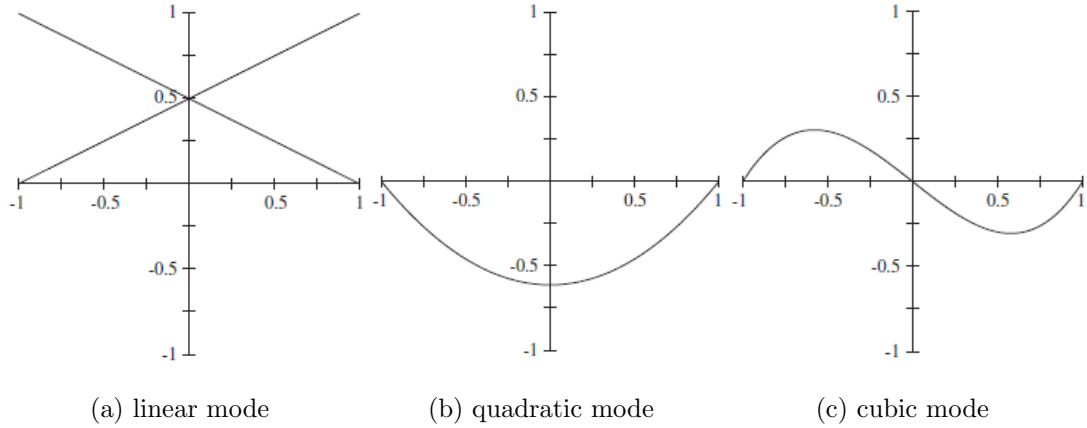


Figure 2.4: 1D basis functions [10]

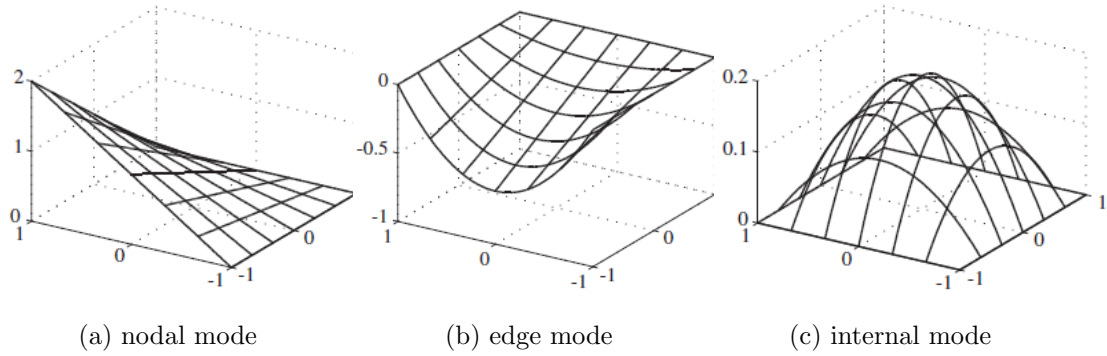


Figure 2.5: 2D basis functions [10]

The next step is to insert the discretization approach (2.18), (2.19) and (2.20) into the weak formulations of the problem. The weak form of the equation of motion leads to:

$$\sum_{e=1}^{\#elements} \left[\tilde{\omega}^T \left(\underbrace{\int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \rho \mathbf{N}^{(e)} d\Omega}_{\text{element mass matrix } K_m^{(e)}} \ddot{\mathbf{u}}^{(e)} + \underbrace{\int_{\Omega^{(e)}} (\mathbf{L}\mathbf{N}^{(e)})^T \mathbf{C} (\mathbf{L}\mathbf{N}^{(e)}) d\Omega}_{\text{element stiffness matrix } K_s^{(e)}} \tilde{\mathbf{u}}^{(e)} \right. \right. \\
 \left. \left. - \underbrace{\int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \hat{\mathbf{b}}^{(e)} d\Omega - \int_{\Gamma_\sigma^{(e)}} \mathbf{N}^{(e)T} \hat{\mathbf{t}}^{(e)} d\Gamma}_{\text{element loadvector } F^{(e)}} \right) \right]^{(e)} = 0 .$$

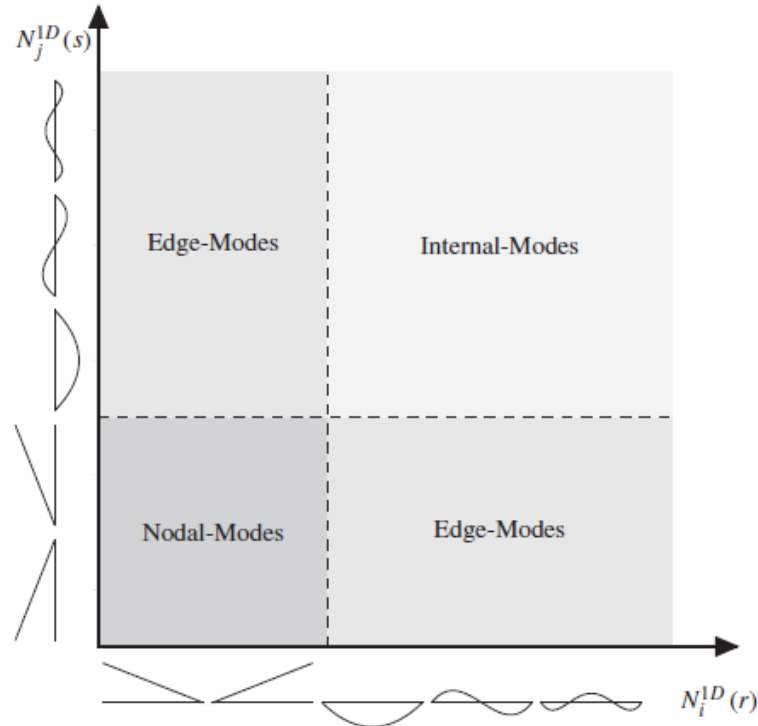


Figure 2.6: Structure of combined 1D-modes [10]

Inserting the discretization into the weak form of the weak equation yields to a very similar equation:

$$\sum_{e=1}^{\#elements} \left[\tilde{\omega}^T \left(\overbrace{\int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \mathbf{N}^{(e)} d\Omega}^{\text{element mass matrix } \bar{K}_m^{(e)}} \ddot{\tilde{\Phi}}^{(e)} + \overbrace{\int_{\Omega^{(e)}} (\bar{\mathbf{L}}\mathbf{N}^{(e)})^T \alpha^2 (\bar{\mathbf{L}}\mathbf{N}^{(e)}) d\Omega}^{\text{element stiffness matrix } \bar{K}_s^{(e)}} \tilde{\Phi}^{(e)} - \underbrace{\int_{\Gamma_\sigma^{(e)}} \mathbf{N}^{(e)T} \hat{t}^{(e)} d\Gamma}_{\text{element load vector } \bar{F}^{(e)}} \right) \right]^{(e)} = 0 .$$

In the case that three-dimensional vector field \underline{u} is solved, the same basis functions are used for every spatial direction which implies that for every element three components have to be calculated.

As previously mentioned, locally supported shapefunctions are used. To evaluate the integrals in local coordinates the basis functions has to be transformed to the local coordinates $r, s, t \in [-1, 1]$. For this purpose a local-to-global mapping is necessary.

Isoparametric concept

To reproduce our solution in terms of the global coordinates $x, y, z \in \mathbb{R}^3$, a mapping from the local coordinates to the global ones is needed. One possible local to global mapping is the isoparametric concept. The idea of this approach is to use the basis functions for discretizing the solution field likewise for the local-to-global mapping

$$x = \mathbf{Q}_e(r, s, t) = \mathbf{N}(r, s, t) x^{(e)} , \quad (2.21)$$

where x are the global coordinates and $x^{(e)}$ are the local coordinates of the corresponding nodes. Using high order finite elements this concept only provides a linear mapping, because the edge and faces modes have no node associated.

Now, it is possible to set up the coordinate transformation to compute the integral in the weak formulations of the equations. The aim is to integrate over the whole domain Ω

$$d\Omega = dx dy dz = |\det(\mathbf{J})| dr ds dt ,$$

where \mathbf{J} is the Jacobian being defined as:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{pmatrix} .$$

With equation (2.21), the Jacobian can be written in terms of the basis functions:

$$\mathbf{J} = \frac{\partial(\mathbf{x})}{\partial(\xi)} = \frac{\partial \mathbf{N}}{\partial \xi} \mathbf{x}^e .$$

Knowing the local to global mapping \mathbf{Q}_e from equation (2.21), the standard shapefunctions can be determined:

$$N^{(e)}(x, y, z) = N_{st}(\mathbf{Q}_e^{-1}(x, y, z)) .$$

Now the element mass matrices can be expressed as an integral over the standard basis functions in local coordinates:

$$\mathbf{K}_m^{(e)} = \int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \rho \mathbf{N}^{(e)} dx dy dz = \int_{\Omega_{st}^{(e)}} \mathbf{N}_{st}^T \rho \mathbf{N}_{st} |det(\mathbf{J})| dr ds dt \quad (2.22)$$

$$\hat{\mathbf{K}}_m^{(e)} = \int_{\Omega^{(e)}} \mathbf{N}^{(e)T} \mathbf{N}^{(e)} dx dy dz = \int_{\Omega_{st}^{(e)}} \mathbf{N}_{st}^T \mathbf{N}_{st} |det(\mathbf{J})| dr ds dt . \quad (2.23)$$

Exactly the same way, the element load vectors are transformed.

Before it is possible to express the stiffness matrices in terms of the local coordinates, it is necessary to transform the derivatives of the basis functions with respect to the global coordinates

$$\begin{pmatrix} \frac{\partial N^i}{\partial x_1} \\ \frac{\partial N^i}{\partial x_2} \\ \frac{\partial N^i}{\partial x_3} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_3}{\partial x_1} \\ \frac{\partial \xi_1}{\partial x_2} & \frac{\partial \xi_2}{\partial x_2} & \frac{\partial \xi_3}{\partial x_2} \\ \frac{\partial \xi_1}{\partial x_3} & \frac{\partial \xi_2}{\partial x_3} & \frac{\partial \xi_3}{\partial x_3} \end{pmatrix}}_{\frac{\partial(\xi)}{\partial(\mathbf{x})}} \begin{pmatrix} \frac{\partial N^i}{\partial \xi_1} \\ \frac{\partial N^i}{\partial \xi_2} \\ \frac{\partial N^i}{\partial \xi_3} \end{pmatrix} ,$$

where $\frac{\partial(\xi)}{\partial(\mathbf{x})}$ is the Inverse of the Jacobian \mathbf{J} . Now it is possible to estimate the derivatives of the shapefunctions with respect to the local coordinates.

Finally we can determine the element stiffness matrices in terms of the local coordinates for both equations:

$$\mathbf{K}_s^{(e)} = \int_{\Omega^{(e)}} (\mathbf{LN}^{(e)})^T \mathbf{C} (\mathbf{LN}^{(e)}) dx dy dz = \int_{\Omega_{st}^{(e)}} \mathbf{B}_{st}^T \mathbf{C} \mathbf{B}_{st} |det(\mathbf{J})| dr ds dt \quad (2.24)$$

$$\hat{\mathbf{K}}_s^{(e)} = \int_{\Omega^{(e)}} (\hat{\mathbf{L}}\mathbf{N}^{(e)})^T \alpha^2 \hat{\mathbf{L}}\mathbf{N}^{(e)} dx dy dz = \int_{\Omega_{st}^{(e)}} \hat{\mathbf{B}}_{st}^T \alpha^2 \hat{\mathbf{B}}_{st} |det(\mathbf{J})| dr ds dt . \quad (2.25)$$

To evaluate these integrals a numerical integration method is illustrated in the next section.

Numerical integration

For integrating over the element domain $\Omega_{st}^{(e)}$, the Gaussian quadrature is used. As the local coordinates distributed from -1 to +1 over the element, it is sufficient to consider the numerical integration on this domain.

The integration over a function f is approximated by the sum over function values at specific coordinates, weighted with a specific constant:

$$\int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} f(r, s, t) dr ds dt \approx \sum_{i=1}^{\#GP} \sum_{j=1}^{\#GP} \sum_{k=1}^{\#GP} \alpha^i \alpha^j \alpha^k f(r^i, s^j, t^k) .$$

The specific coordinates r^i, s^j, t^k are called Gauß points (GP). Choosing the quantity and the locations of the Gauss points and the corresponding weights properly, polynomials of order $p = 2 \cdot \#GP - 1$ can be integrated exactly.

# GP	i	r^i	α^i
1	1	0	2
2	1	$-\sqrt{3}/3$	1
	2	$+\sqrt{3}/3$	1
3	1	$-\sqrt{3/5}$	5/9
	2	0	8/9
	3	$+\sqrt{3/5}$	5/9
4	1	$-\sqrt{(15 + \sqrt{120})/35}$	$(18 - \sqrt{30})/36$
	2	$-\sqrt{(15 - \sqrt{120})/35}$	$(18 + \sqrt{30})/36$
	3	$+\sqrt{(15 - \sqrt{120})/35}$	$(18 + \sqrt{30})/36$
	4	$+\sqrt{(15 + \sqrt{120})/35}$	$(18 - \sqrt{30})/36$
5	

Table 2.1: Gauss points and weights for numerical integration

For multidimensional integration the same set of Gauss points and weights are used for every spatial direction.

After the evaluation of each element mass matrix and element stiffness matrix, the element matrices have to be assembled into the global matrices. Finally, the spatially discretized equations with global mass matrix \mathbf{M} , global stiffness matrix \mathbf{K} and global loadvector \mathbf{F} is obtained:

$$\mathbf{M}\ddot{\hat{x}}(t) + \mathbf{K}\hat{x}(t) = \mathbf{F}(t) .$$

The spatial discretization is finished. In the subsequent section the time discretization is introduced.

2.3.2 The Newmark's scheme

The Newmark's scheme is a second-order integration scheme, which is used for solving differential equations. The basis of following derivation is taken from [8].

Two parameters control the behavior of the approximation, $\gamma \in [0, 1]$ and $\beta \in [0, 0.5]$. The starting point is the approximation of the acceleration with a finite differences scheme depending on the two parameters:

$$A^\gamma(\tau) = A_n + 2\gamma \frac{A_{n+1} - A_n}{\Delta t} \tau \quad (2.26)$$

$$A^\beta(\tau) = A_n + 6\beta \frac{A_{n+1} - A_n}{\Delta t} \tau . \quad (2.27)$$

The integration (2.26) over time τ yields:

$$V(\tau) = A_n \tau + \frac{2\gamma}{2} \frac{A_{n+1} - A_n}{\Delta t} \tau^2 + c .$$

The integration constant is set to the value $V(\tau = 0) = V_n$:

$$V(\tau) = V_n + A_n \tau + \gamma \frac{A_{n+1} - A_n}{\Delta t} \tau^2 . \quad (2.28)$$

Exactly the same equation (2.27) is integrated and another equation for the velocity is obtained:

$$V(\tau) = V_n + A_n \tau + \frac{6\beta}{2} \frac{A_{n+1} - A_n}{\Delta t} \tau^2 .$$

This equation is integrated once again over τ , which results in an expression for the displacement.

$$D(\tau) = V_n \tau + \frac{1}{2} A_n \tau^2 + \frac{6\beta}{6} \frac{A_{n+1} - A_n}{\Delta t} \tau^3 + c .$$

This time, the integration constant is set to $D(\tau = 0) = D_n$:

$$D(\tau) = D_n + V_n \tau + \frac{1}{2} A_n \tau^2 + \frac{6\beta}{6} \frac{A_{n+1} - A_n}{\Delta t} \tau^3 . \quad (2.29)$$

In a next step, τ is set to Δt in (2.28) and (2.29) to get an approximation of the velocity and the displacement of the next timestep:

$$V_{n+1} = V_n + (1 - \gamma)\Delta t A_n + \gamma A_{n+1} \quad (2.30)$$

$$D_{n+1} = D_n + \Delta t V_n + \left(\frac{1}{2} - \beta\right)\Delta t^2 A_n + \beta\Delta t^2 A_{n+1} . \quad (2.31)$$

Until now, there are two equation for the three unknowns displacement, velocity and acceleration, or in case of the wave equation the volumetric dilatation and its first and second derivative. The third equation is obtained from the differential equation, which is derived from the problem we want to solve.

Before we insert the approximation in the differential equation, equation (2.31) is reordered to get an expression for the new acceleration:

$$A_{n+1} = \frac{1}{\beta\Delta t^2}(D_{n+1} - D_n) - \frac{1}{\beta\Delta t}V_n + \frac{1 - 2\beta}{2\beta}A_n . \quad (2.32)$$

As one can see, it is possible to calculate the acceleration and velocity of the new timestep if the new displacement is known.

Finally, equations (2.30) and (2.32) are inserted in the differential equation. As an example we choose the discretized force balance with mass matrix M , damping matrix C and stiffness matrix K :

$$MA_{n+1} + CV_{n+1} + KD_{n+1} = F(t_{n+1}) .$$

Reordering all terms with the unknown displacement of the new timestep leads to:

$$\begin{aligned} \overbrace{\left(\frac{1}{\beta\Delta t^2}M + \frac{\gamma}{\beta\Delta t}C + K\right)}^{K_{effdyn}} D_{n+1} &= \left(\frac{1}{\beta\Delta t^2}M + \frac{\gamma}{\beta\Delta t}C\right)D_n \\ &+ \left(\frac{1}{\beta\Delta t}M + \left(\frac{\gamma}{\beta} - 1\right)C\right)V_n \\ &+ \left(\frac{1 - 2\beta}{2\beta}M + \Delta t\left(\frac{\gamma - 2\beta}{2\beta}C\right)\right)A_n \\ &+ F(t_{n+1}) . \end{aligned}$$

The matrix on the left does not change over time and is called the effective dynamic stiffness matrix. The right hand side is called the effective dynamic force vector, which is not constant over time because it depends on the displacement, velocity and acceleration of the previous timestep. Thus, in every timestep, the following equation system has to be solved:

$$\mathbf{K}_{\text{effdyn}} D_{n+1} = \mathbf{F}_{\text{effdyn}} .$$

This equation can be solved in the usual manner. Knowing the new displacement, one can calculate the new acceleration with (2.32) and knowing the new acceleration, we can calculate the new velocity with (2.30).

Properties of Newmark's scheme

As mentioned previously, the choice of the parameters γ and β characterizes the approximation. In general the Newmark's scheme is an implicit integration scheme unless β is zero.

There are several possible combinations of the two parameters with $\gamma \in [0, 1]$ and $\beta \in [0, 0.5]$. The common central difference approximation can be obtained choosing $\gamma = \frac{1}{2}$ and $\beta = 0$. In case the acceleration should be approximated linearly, the proper choice is $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{6}$. Changing only β to $\frac{1}{4}$ one get a constant averaged acceleration scheme.

Besides the specification of the approximation, also the stability of the Newmark's method is controlled by the parameters. Choosing $\beta \geq \frac{\gamma}{2} \geq \frac{1}{4}$ will lead to an unconditionally stable scheme. For example, the previously mentioned constant acceleration is unconditionally stable.

If $\gamma \geq \frac{1}{2}$ and $\beta \leq \frac{\gamma}{2}$, the method is conditionally stable since this choice implies conditions on the timestep size Δt .

It is also possible to build in numerical damping. In this case γ should be greater than one half.

2.4 Time-reversed simulation

Time-reversed simulations are used to locate the source of a wave and is commonly applied in seismology. Scientists record the signals of seismic waves with the goal to detect the origin of the wave. The further explanations follow the article [2].

The general set up for time-reversed simulation is an acoustic time-reversal mirror that surrounds the acoustic source. The mirror consists of many sensors. The time reversed simulation operates in two steps. At first, the sensors react like listeners and record the arriving propagation waves over time. In a second step, the recorded signals are reproduced in reversed time. This time, every sensor behaves like an acoustic source and starts to propagate a wave into the medium. The waves will interfere, and the highest positive interference will occur at the point where the original wave was initially triggered.

The time-reversed simulation works because wave propagation is linear. This implies that a small change in the initial data results only in a small change in the final wave [2]. This fact is essential for computerbased simulation because small round-off errors and truncation are inevitable.

In [2] an interesting example is mentioned, where an under water ultrasonic pulse is sent through 2,000 randomly placed steel rods. The time-reversing mirror is able to reproduce the original pulse.

In general, not every simulation can be reproduced in time-reverse. Transferring the previous example to a pinball that is fired through hundreds of steel rods, the trajectory of the pinball could not be recreated in a time-reversed simulation. After many collisions a small deviation of the initial shooting direction determines whether an object is hit or not. This small deviation has a huge effect on the trajectory of the pinball.

Principally, it is not necessary to completely enclose the acoustic source with the recording sensors. The quality of the reproduction is not deteriorated if the gaps between the sensor are not greater than half the wavelength of the propagating wave. The reason is the diffraction of waves [2]. Gaps of half the wavelength will get filled in as the wave propagates. This size is also the resolution limit of the simulation. Any information smaller than this can not be reproduced. In most cases, an accuracy in the scale of half the wavelength is not required, such that the number of sensors can be highly reduced.

Chapter 3

Simulations

In the following chapter several simulations with the wave equation and the equation of motion are performed and compared with an experiment. In the end, a time-reversed simulation is minuted to locate the initial starting point of the wave.

3.1 Generation of different simulation setups

In section 2.1.1, different boundary conditions are declared. Based on the circumstance that there are two different equations and two different boundary conditions, several different simulation setups were conducted, see figure 3.1. The basic computation domain is a simple cube. On every single surface of the cube exactly one type of boundary condition is defined.

The goal is to generate synthetic sensor signals on the surface of the cube. For that reason, it is convenient to measure a component of the solution that is not constrained by the boundary conditions. Therefore, the derivative of the solution is recorded if the surface is restricted with a Dirichlet boundary condition (DBC) and if the surface is constrained by a Neumann boundary condition (NBC), the solution is observed.

3.1.1 Simulation setups with the equation of motion

Setup 1

Assuming that the wave is initiated by a force requires a NBC. The cube is bore freely and the six transational and rotational degrees of freedoms of a solid body are restricted. This implies natural NBC (no forces) on the other surfaces and six nodal DBC. For the generation of the synthetic sensor signals the displacements in the direction of the surface normals is measured at the sensor locations.

It is possible to construct a simulation setup with non-natural NBC. In case of the equation of motion, this is unphysical. In the equation of motion a NBC corresponds to a force that is applied on the boundary of our domain. If this force is non-zero, this would imply that the body is connected to something that would change the wave propagation inside of the cube. For that reason, there is only one meaningful setup for the equation of motion.

3.1.2 Simulation setups with the wave equation

As the wave equation does not solve for displacements but rather for the volumetric change simulations with DBC and NBC on the outer surfaces are reasonable.

Setup 2

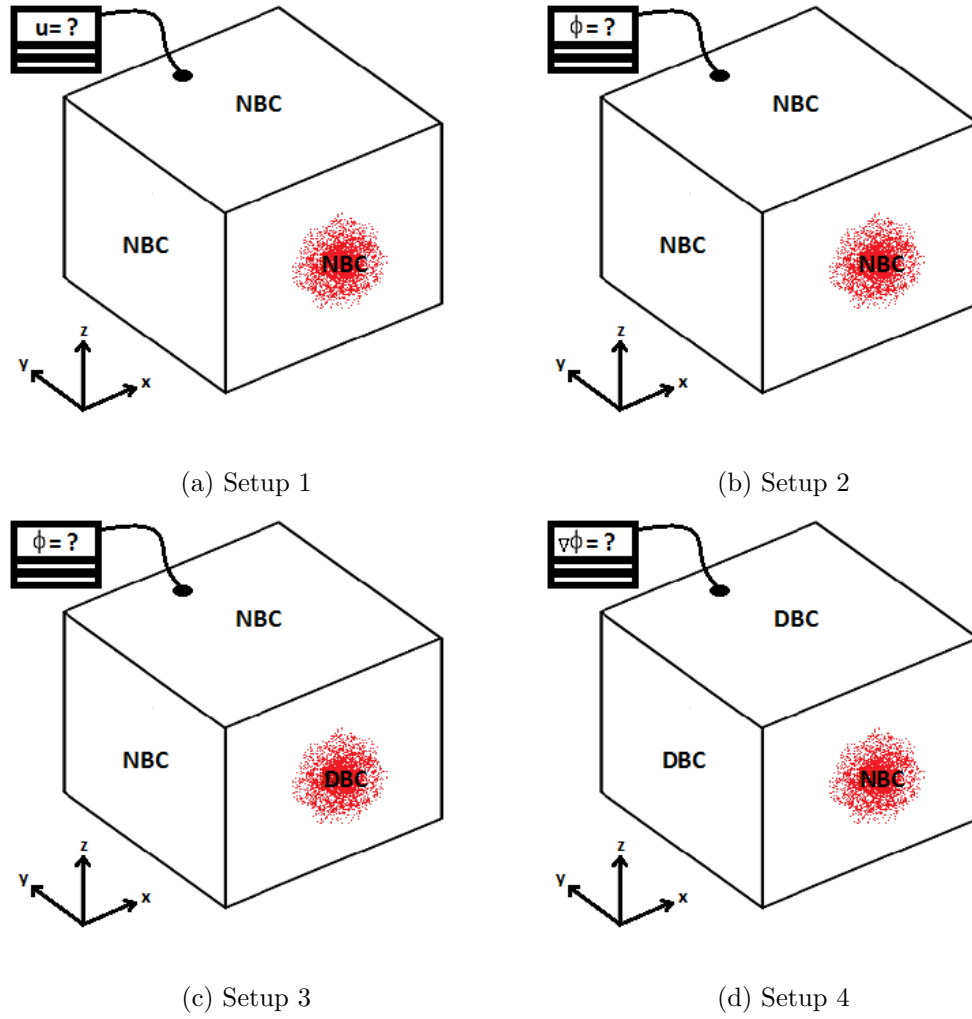
This setup is similar to setup 1, the difference is the governing equation. Similarly, the wave is induced with a NBC and on the other surfaces natural NBC are applied. The computed solution is now the volumetric dilatation instead of the displacement field.

Setup 3

Compared to setup 2, only the boundary condition on the surfaces where the wave is induced, is changed. The wave is initiated with a DBC. The other surfaces stay constrained by natural NBC. As synthetic sensor signals, the volumetric change is recorded on the surface.

Setup 4

Setup 4 is the complement to setup 3. Instead of a DBC, the wave is initiated with a NBC and instead of natural NBC, all the other surfaces are constrained with DBC. In that case, the derivative of the volumetric change in direction of the surface normal is recorded.



Setup	basic equation	Wave initiation Γ_{load}	BC on Γ	sensor measurement
1	Equation of motion	NBC	NBC	\mathbf{u} displacement
2	Wave equation	NBC	NBC	ϕ volumetric change
3	Wave equation	DBC	NBC	ϕ volumetric change
4	Wave equation	NBC	DBC	$\nabla\phi$ gradient vol.change

Figure 3.1: Different simulation setups with the equation of motion and the wave equation

3.2 Simulations with the different boundary condition setups

Before the launching of the simulations, the wave initiation has to be regarded. For computer simulations, it is convenient to avoid discontinuities. For that reason, a smooth load function is needed to describe the boundary value of the surface where the wave is initiated.

The wave is initiated in the middle of the front surfaces with a two-dimensional Gauss function in space (3.2):

$$g(x) = A_s \cdot \exp\left(\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma_s^2}\right) .$$

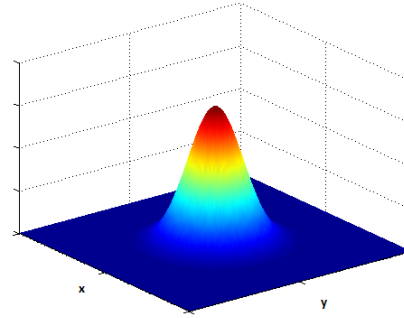


Figure 3.2: Two-dimensional Gauss function in space with amplitude A_s , peak location (x_0, y_0) and standard deviation σ_s

A_s is a constant that controls the height of the peak, whereas x_0 and y_0 determine the location of the peak. The width of the Gaussian bell can be set with the standard deviation σ_s . In the following simulations σ_s is equal to the twentieth part of an edge of the cube. In order to obtain a continuous impuls over space and time this Gauss function is multiplied with the derivative of a Gauss function in time, see figure 3.3:

$$h(t) = A_t \cdot -(t - t_0) \cdot \exp\left(\frac{(t - t_0)^2}{2\sigma_t^2}\right) .$$

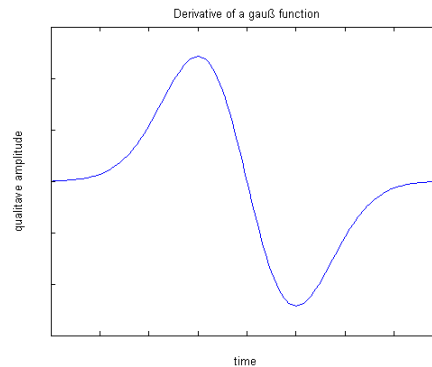


Figure 3.3: Derivative of an one-dimensional Gauss function in time with amplitude A_t , peak location t_0 and standard deviation σ_t

The final boundary value is then composed as:

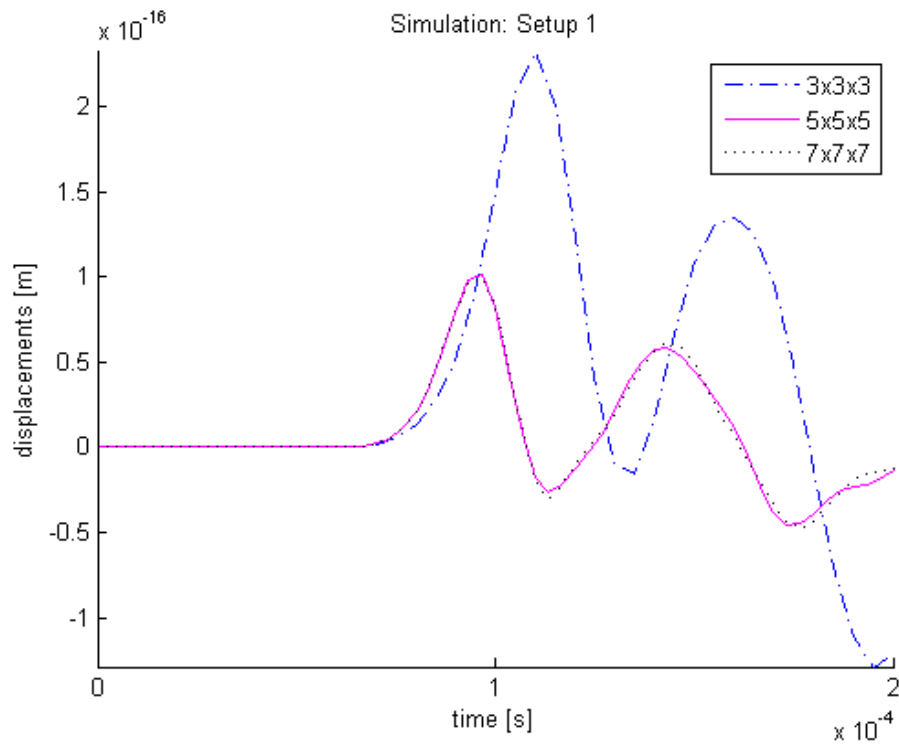
$$\theta(x, t) = g(x) \cdot h(t) .$$

3.2.1 Convergence study

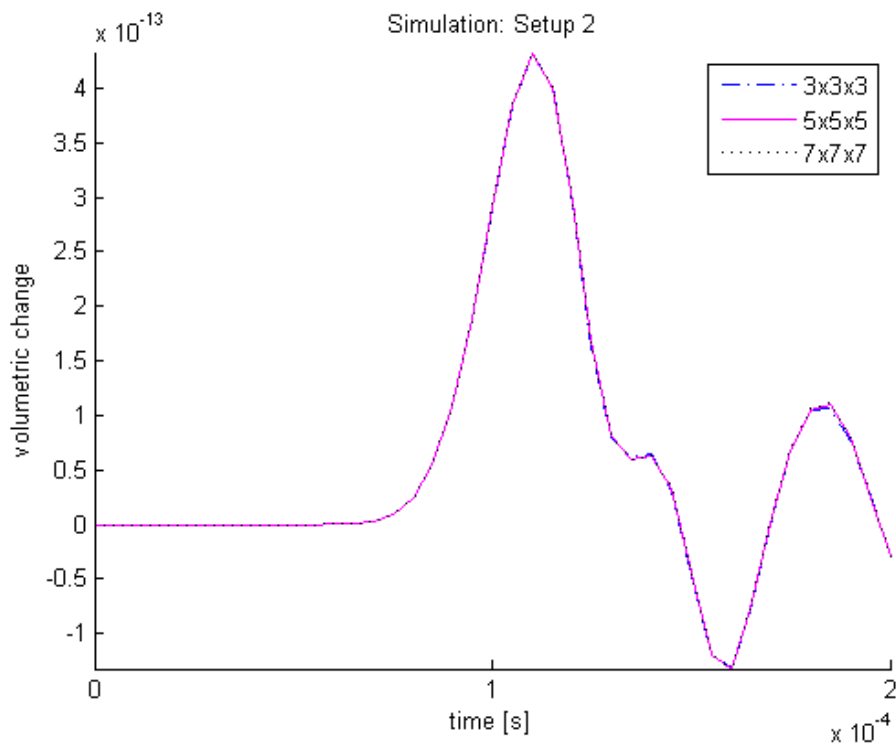
Knowing that the computer calculates an approximated solution of the derived equations, it is necessary to ensure the convergence of the computed solutions with different discretizations.

The results depicted in figure 3.4 and 3.5, suggest that a discretization with $5 \times 5 \times 5$ elements seems to be sufficient, on the condition that the polynomial degree is set to three. The refinement to $7 \times 7 \times 7$ elements does not pay off as the increasing computation time do not improve the solution remarkably. Furthermore the influence of the two-dimensional Gauss function for the wave initiation should be minded. Avoiding enormous gradients in the computed solution that could induce instabilities, the standard deviation σ_s is set to a twentieth part of an edge of the cube.

Similar to the spatial discretization, one should choose a reasonable discretization in time. In section 2.3.2, it is mentioned that it is possible to choose the parameters of the Newmark's scheme such that the method is unconditionally stable. Nevertheless, it should be ensured that the initiation of the wave can be sampled correctly. For this purpose, the standard deviation σ_t in the equation of figure 3.3 is set as twice the timestep size Δt .

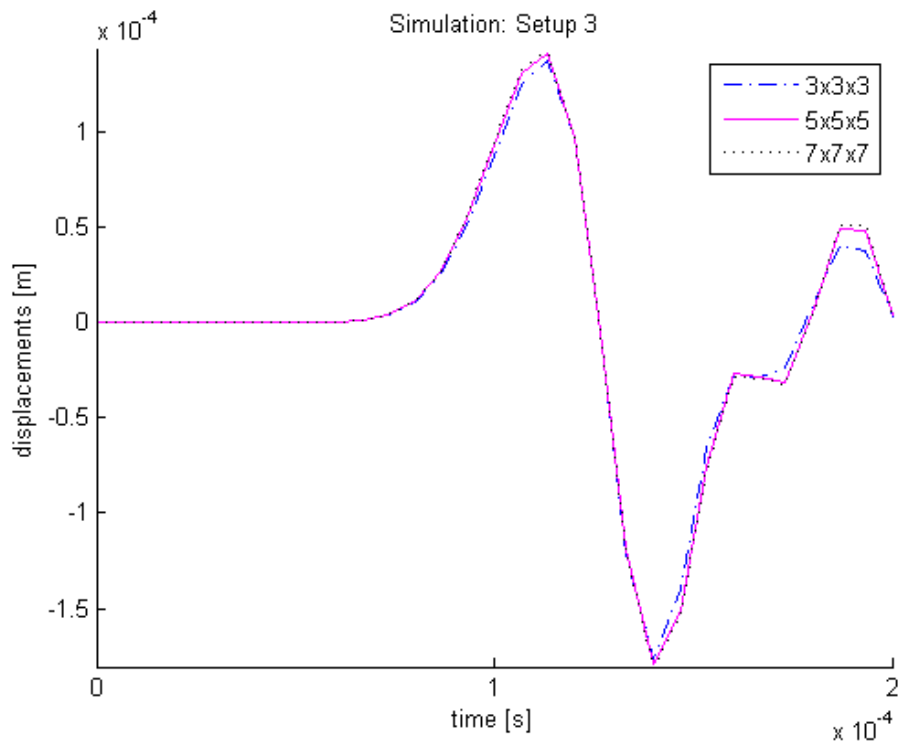


(a) Setup 1

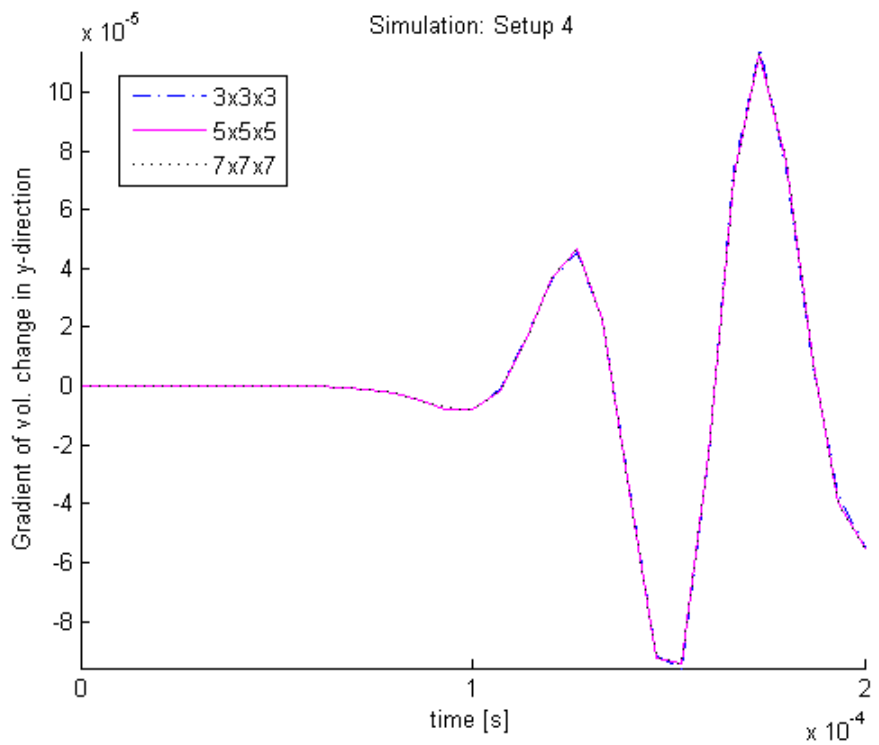


(b) Setup 2

Figure 3.4: Recorded signals of the simulation at coordinates [150 300 30] (at the back of the cube) with different spatial discretizations, polynomial degree $p=3$



(c) Setup 3



(d) Setup 4

Figure 3.5: Recorded signals of the simulation at coordinates [150 300 30] (at the back of the cube) with different spatial discretizations, polynomial degree $p = 3$

3.3 Reproduction of an experiment

3.3.1 Description of the experiment

For later comparison, the different simulation setups in section 3.1 are used to reproduce an experiment that was performed by the chair of Zerstörungsfreie Prüfung at the Technische Universität München, [9].

The body in experiment is a $30 \times 30 \times 23$ [cm] cube made of ultra-high performance concrete (UHPC). The material is assumed to be homogenous with a wave propagation speed as approximated $4750 \frac{m}{s}$. In the experiment, an ultrasonic impuls is applied onto the cube to induce the wave into the cube, see figure 3.6.

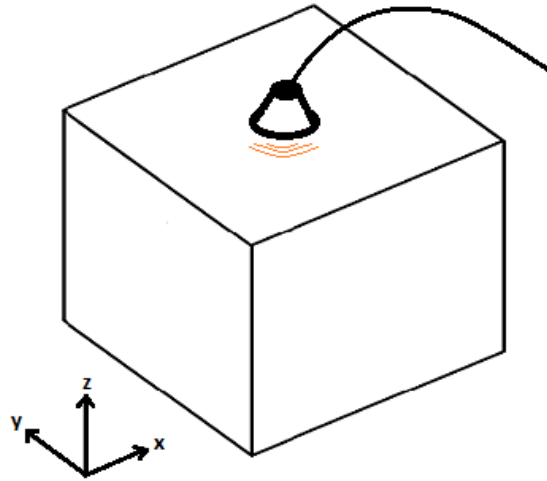
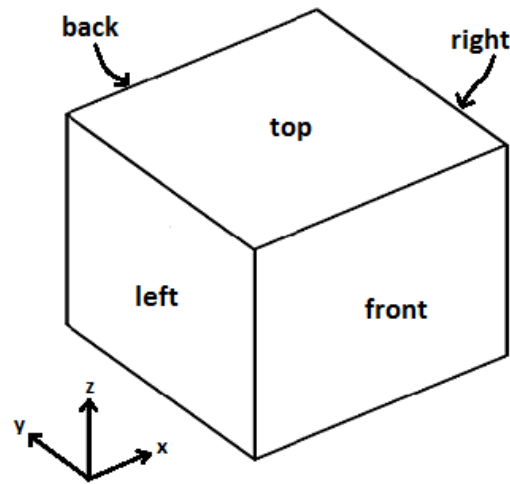


Figure 3.6: Experimental setup

In the experiment, the propagating wave is recorded by several sensors that are placed on the surfaces of the cube. For the simulations, the sensor coordinates declared in [9] are transformed to a right-handed coordinates systems with origin in the front lower corner of the cube, see figure 3.7.

The location of the ultrasonic initiator is on the top surface of the cube. Its precise location varied during several experiments, see table 3.1.

Be aware that the measured signals in the experiment have no units assigned. The amplitude only is measured qualitatively. In all experiments except experiment E2, the measured amplitudes are too high, wherefore the only unclipped measurements are in experiment E2.



Sensors	x in mm	y in mm	z in mm	surface orientation
S1	275	25	230	top
S2	100	100	230	top
S3	0	200	130	left
S4	0	150	30	left
S5	0	25	30	left
S6	25	0	205	front
S7	150	0	30	front
S8	275	0	30	front
S9	25	275	230	top
S10	25	300	30	back
S11	150	300	30	back
S12	275	300	205	back
S13	300	275	30	right
S14	300	150	30	right
S15	300	100	130	right

Figure 3.7: Locations of the different sensors

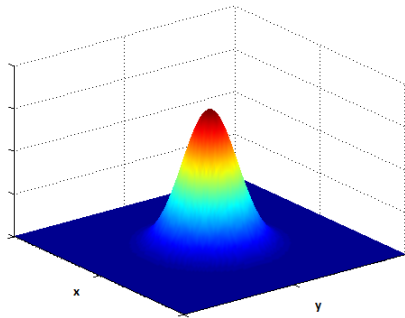
Experiments	x	y	z
E1	200	150	230
E2	150	150	230
E3	100	150	230
E4	100	200	230
E5	100	250	230
E6	150	250	230
E7	200	250	230
E8	200	200	230
E9	200	150	230

Table 3.1: Location of the ultrasonic initiator in different experiments

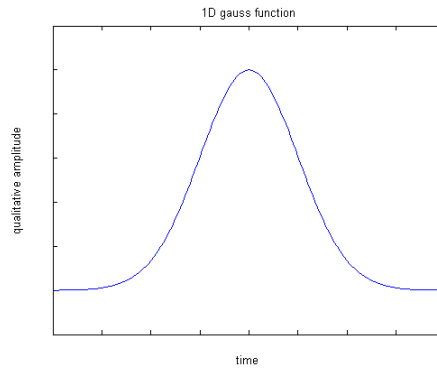
3.3.2 Simulation of the experiment

In this section, the experiment is reproduced to generate the synthetic sensor signals. In [9] there is no exact information about the ultrasonic initiator.

It is assumed that the load function of the ultrasonic initiator can be approximated as the product of a two-dimensional Gauss function in space and one-dimensional Gauss function in time.



(a) 2D Gauss function in space



(b) 1D Gauss function in time

Figure 3.8: Wave initiation for reproduction of the experiment, the load function is obtained by a multiplication of (a) and (b)

The two-dimensional Gauss function in space is known from figure 3.2 and the one-dimensional looks very similar:

$$\tilde{h}(t) = \tilde{A}_t \cdot \exp\left(-\frac{(t - t_0)^2}{2\sigma_t^2}\right).$$

The final boundary value is then composed as:

$$\tilde{\theta}(x, t) = g(x) \cdot \tilde{h}(t) . \quad (3.1)$$

The next step is determining t_0 , the time when the ultrasonic impuls is initiated. Consider experiment E2, the time when the sensor records the first arriving signal is measured. Knowing the distance from the source to the sensor and the wavespeed, it is possible to estimate the initiation time, see table 3.2.

Sensor	distance to source [mm]	first arriving signal [ms]	computed initiation time [ms]
S1	190.1	0.237	0.197
S2	99.5	0.214	0.193
S3	232.2	0.239	0.190
S4	308.9	0.253	0.188
S5	333.2	0.258	0.188
S6	217.1	0.241	0.195
S7	308.9	0.252	0.187
S8	333.2	0.259	0.189
S9	190.1	0.238	0.198
S10	333.2	0.259	0.189
S11	308.9	0.252	0.188
S12	217.1	0.241	0.195
S13	333.2	0.259	0.189
S14	308.9	0.253	0.188
S15	232.2	0.240	0.191

Table 3.2: Computation of the initiation time of the wave for experiment E2

The averaged initiation time is 0.191 ms. Assuming that the measurement of sensors S1, S6, S9 and S12 are defective, the averaged initiation time is 0.189 ms. The fluctuation of the computed initiation for different sensors might be caused by small deviations in the defined sensor coordinates and the actual sensor placements in the experiment. In terms of inevitable errors in measurements of the above quantities the initiation time t_0 of the Experiment E2 is assumed as 0.19 ms.

Beside the initiation time, it is also necessary to define values for the material parameters as density ρ , Young's modulus E and Poisson ratio ν . The only information that is given, is the measured wave propagation speed. The needed material parameters are determined such that they are in the common range for UHPC and agree with the measured wave speed, see table 3.3.

Young's modulus E	36.55 GPa
Density ρ	1800 $\frac{kg}{m^3}$
Poisson ratio ν	0.2
Wave speed α	4750 $\frac{m}{s}$

Table 3.3: Approximated material parameters

As it is not clear which quantity corresponds to the measured amplitudes (displacements, forces, volumetric changes, . . .), several simulations with the different setups from section 3.1 are minuted.

3.3.3 Observations from the simulation of the experiment

Based on the information that is provided about the experiment, it is not possible to reproduce the amplitude. Several assumptions are integrated in the numerical simulation that simplify the reality.

It is assumed that the cube is ideally homogenous and has perfect dimensional metrics. Additionally, the exact material parameters are unknown, as mentioned previously. Also the bearing of the cube is neglected which influences the computed solution of the simulation. The cube must be fixed some how and every contact point influences the wavepropagation inside of the cube. Finally, the ultrasonic impuls that induced the wave at the experiment has to be modeled with a smooth load function.

The complete results of the reproduction of experiment E2 can be examined in attachment A. In the following the most remarkable observations are mentioned.

The first important observation is that in most cases both models, the experiment and the simulation, record the first arriving signal at approximated the same time. In setup 1, setup 2 and setup 4 you can see that the sensors S1, S6, S9 and S12 the synthetic signals from the simulations record an arriving signal earlier than the experimental signals. This observation becomes reasonable if the applied initiation time 0.190 ms is compared to the computed initiation time of those sensors in table 3.2. Sensor S1, S6, S9 and S12 have the largest deviation. The deviation can be compared to the deviation in the arriving signal in table 3.4.

Another interesting observation is that within setup 2 the sensors S1, S6, S9 and S12 approximate the experimental arriving signal very good whereas all other synthetic sensors react a bit later than in the experiment.

	Setup 1 [ms]	Setup 2 [ms]	Setup 3 [ms]	Setup 4 [ms]
S1	-0.005	0.000	-0.004	-0.003
S2	0.000	+0.002	-0.001	-0.001
S3	0.000	+0.001	0.000	0.000
S4	+0.001	-0.001	0.000	+0.001
S5	+0.002	-0.002	-0.001	0.000
S6	-0.003	0.000	-0.003	-0.003
S7	+0.002	-0.001	+0.001	0.000
S8	0.000	-0.001	+0.001	0.000
S9	-0.005	0.000	-0.003	-0.004
S10	0.000	-0.001	0.000	0.000
S11	-0.001	-0.002	0.000	0.000
S12	+0.004	0.000	-0.003	-0.002
S13	0.000	+0.001	0.000	0.000
S14	-0.001	-0.003	0.000	+0.001
S15	0.000	0.000	0.000	0.000

Table 3.4: Error of the arriving signal of different setups, error = arriving signal of synthetic signal - arriving signal of experimental signal

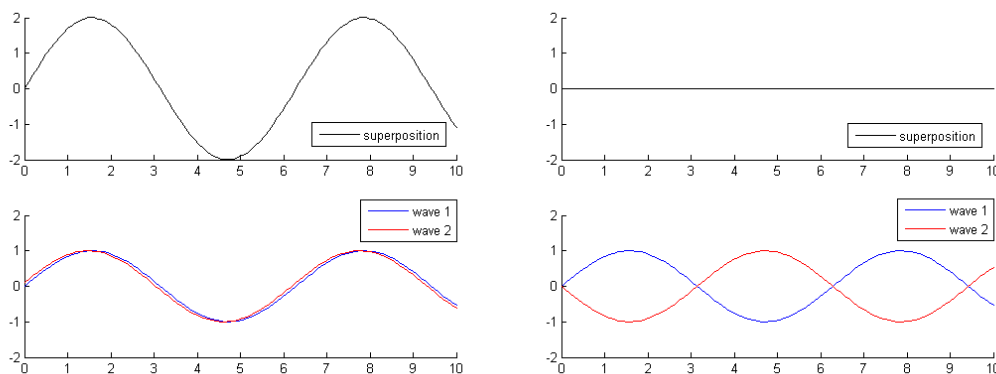
3.4 Influences of the boundary on the reflected wave

Before the time-reversed simulation is performed, influence of different boundary conditions is investigated. For a better understanding how the boundary conditions affects the wave propagation, the reflection of a wave in a simple one-dimensional simulation is observed in figure 3.10.

The Dirichlet boundary cause the wave to travel back with same speed but different polarisation of the amplitude. This leads to a opposite sign of the amplitude after the reflection. This phenomenon can be explained with Newton's third law [4]. The internal force of the wave accelerates the material as the wavefront is passing by. At the end of the rope, the movement is restricted and the internal force need to be balanced with an external force in the opposite direction. This force then induce a wave that travels back with a phase shift of 180 degree.

In contrast to the Dirichlet boundary, the natural Neumann boundary cause no phase shift. This boundary condition restrict the derivative of the solution to be zero. The wave is reflected with the same speed and the same polarisation of the amplitude.

In summary the wave speed stays unaffected during a reflection at the boundary but the phase shift of the wave is influenced. For that reason, it is important to use the same boundary conditions in the time-reversed simulation as in the corresponding forward simulation. Otherwise the location of the origin of the wave might not be found. It could happen that two wave fronts that amplify each other in the forward simulation would compensate each other in the backward simulation, see figure 3.9.



(a) superposition of two identical waves (b) superposition if wave 2 is phaseshifted by 180°

Figure 3.9: Interference of two waves

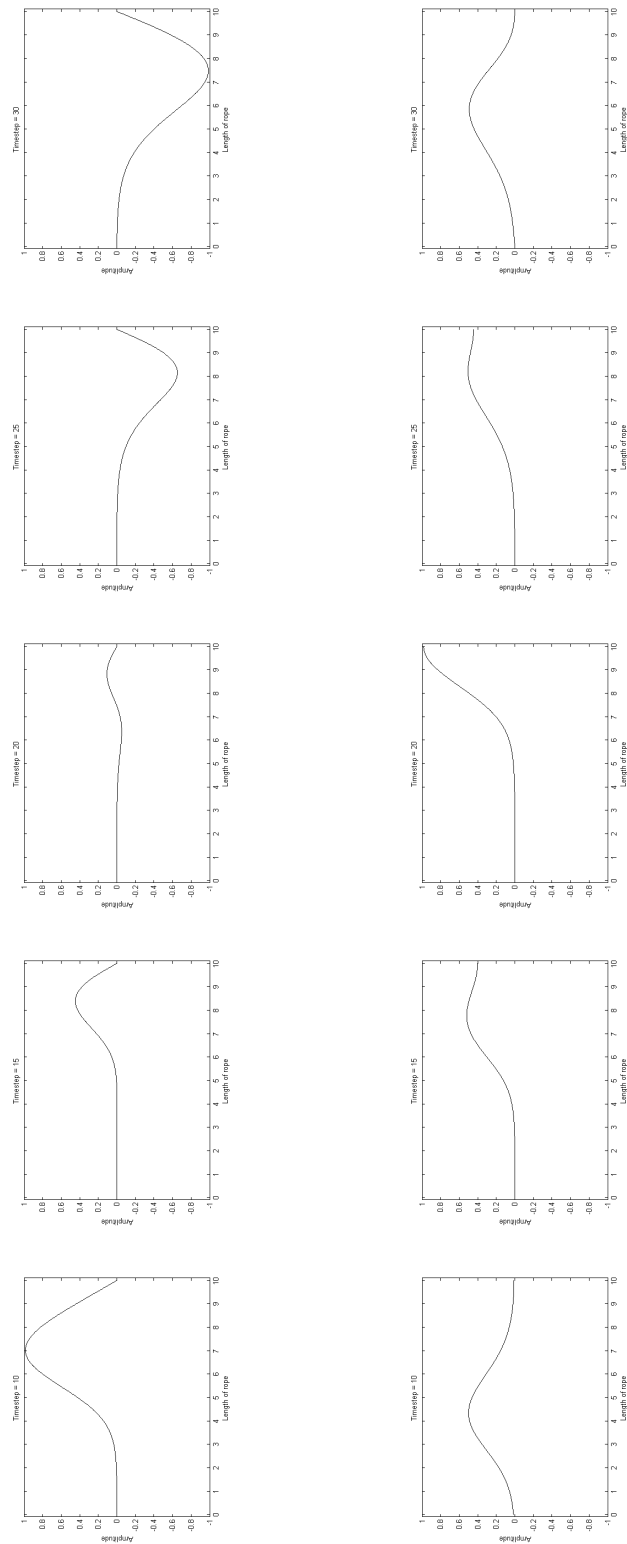


Figure 3.10: Wave reflections with Dirichlet boundary (top) and with natural Neumann boundary (bottom) on the right end of the rope, the wave travels from left to the right end and is then reflected and travels back

3.5 Time-reversed simulations

In a last step, a time-reversed simulation with the equation of motion and the wave equation is performed. The goal is that the wave is traced back until the location of the original source.

The theory derived in chapter 2 holds for both, the forward and the backward simulation. Simulating the wave propagating back to its origin, the solution field and its first and second derivatives of the last timestep of the forward simulation has to be known. If the initial conditions are adapted the same Implementation for the time-reversal is used as for the forward simulation. Ensure that the wave travels in reversed time the sign of the derivatives has to be inverted.

3.5.1 Time-reversed simulation with the equation of motion

The first time-reversed simulation is set with the equation of motion. The setup for the forward simulation can be seen in figure 3.11. The wave was initiated in the middle of the top surface. In the time-reversed simulation all surfaces where constrained by natural NBC. The material parameters were changed to obtain a better observable wave propagation. In figure 3.12, the wave that focuses back to its origin is visualized.

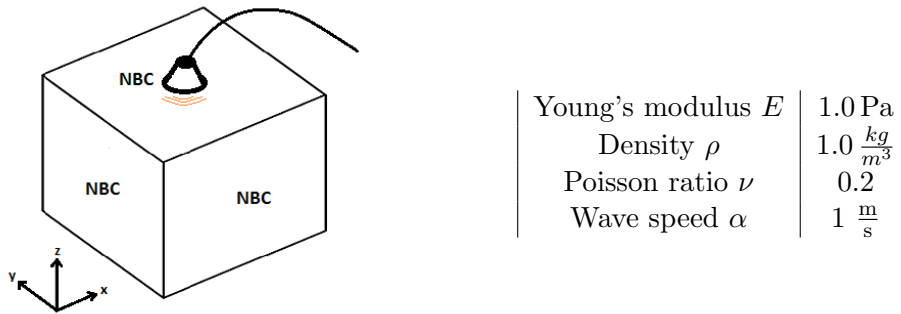
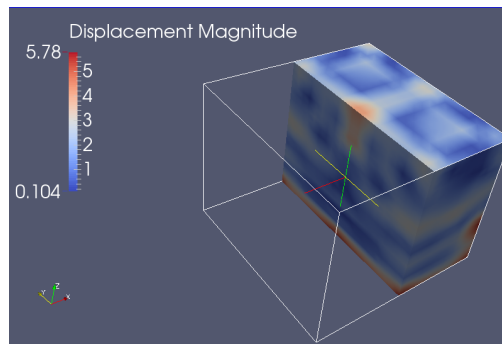


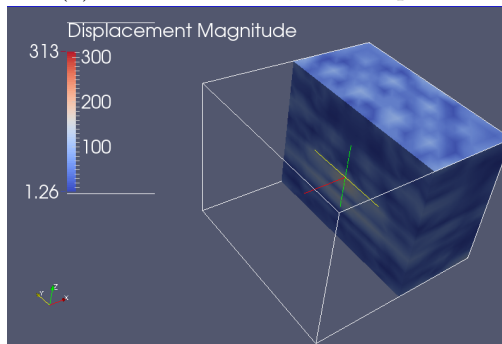
Figure 3.11: Setup of the forward simulation with the equation of motion

In the equation of motion, the solution of the next timestep depends on the displacement, velocity and acceleration of the previous timestep:

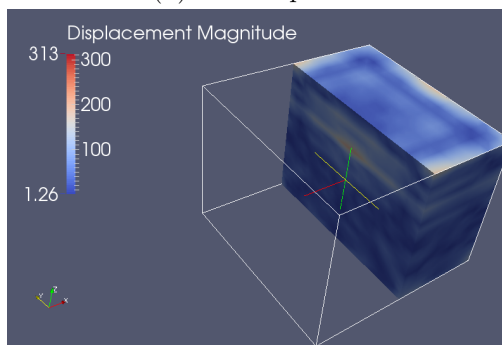
$$\begin{aligned}
 \overbrace{\left(\frac{1}{\beta\Delta t^2}M + K\right)}^{K_{effdyn}} D_{n+1} &= \left(\frac{1}{\beta\Delta t^2}M\right)D_n \\
 &+ \left(\frac{1}{\beta\Delta t}M\right)V_n \\
 &+ \left(\frac{1-2\beta}{2\beta}M\right)A_n \\
 &+ F(t_{n+1}) .
 \end{aligned}$$



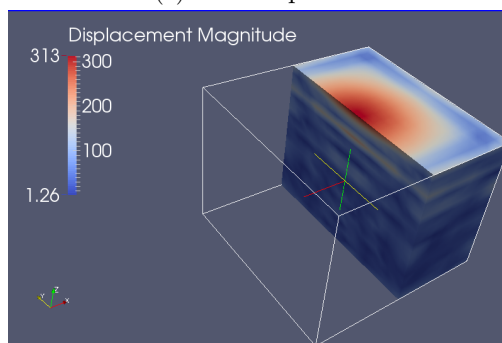
(a) Initial condition, time step = 0



(b) time step = 13



(c) time step = 29



(d) time step = 44, wave propagates back to its origin

Figure 3.12: Time reversed simulation with the equation of motion, observing the resulting displacement with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$

3.5.2 Time-reversed simulation with the wave equation

The second time-reversed simulation is set with the wave equation. The simulation setup of the forward simulation is different, see figure 3.13. In this example, the wave is initiated next to a corner on the top surface. This location was chosen to receive an unsymmetrical wave propagation.

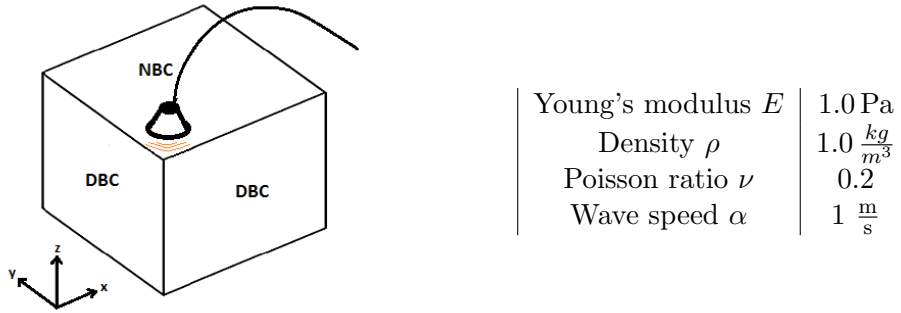


Figure 3.13: Setup of the forward simulation with the wave equation

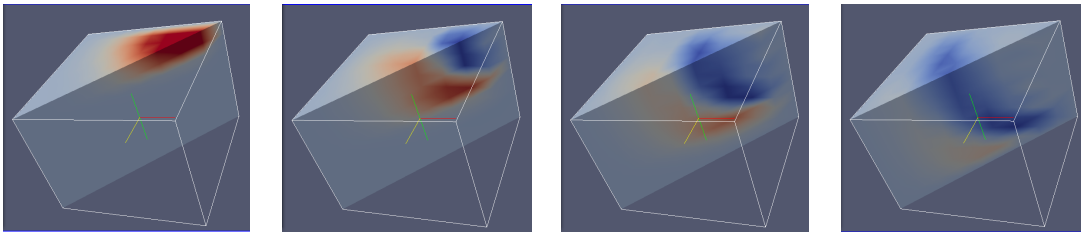


Figure 3.14: Forward simulation with the wave equation

In case of the wave equation, the volumetric change and its first and second derivatives with respect to time are calculated. Also be aware that the mass matrix \tilde{M} , stiffness matrix \tilde{K} and loadvector \tilde{F} are not identical to the mass matrix M , stiffness matrix K and loadvector F in the discretized equation of motion:

$$\begin{aligned}
 \overbrace{\left(\frac{1}{\beta \Delta t^2} \tilde{M} + \tilde{K} \right)}^{\tilde{K}_{effdyn}} \phi_{n+1} &= \left(\frac{1}{\beta \Delta t^2} \tilde{M} \right) \phi_n \\
 &+ \left(\frac{1}{\beta \Delta t} \tilde{M} \right) \dot{\phi}_n \\
 &+ \left(\frac{1 - 2\beta}{2\beta} \tilde{M} \right) \ddot{\phi}_n \\
 &+ \tilde{F}(t_{n+1}) .
 \end{aligned}$$

In figure 3.15 and 3.16, the solution of the forward simulation is compared to the solution of the backward simulation of the wave equation.

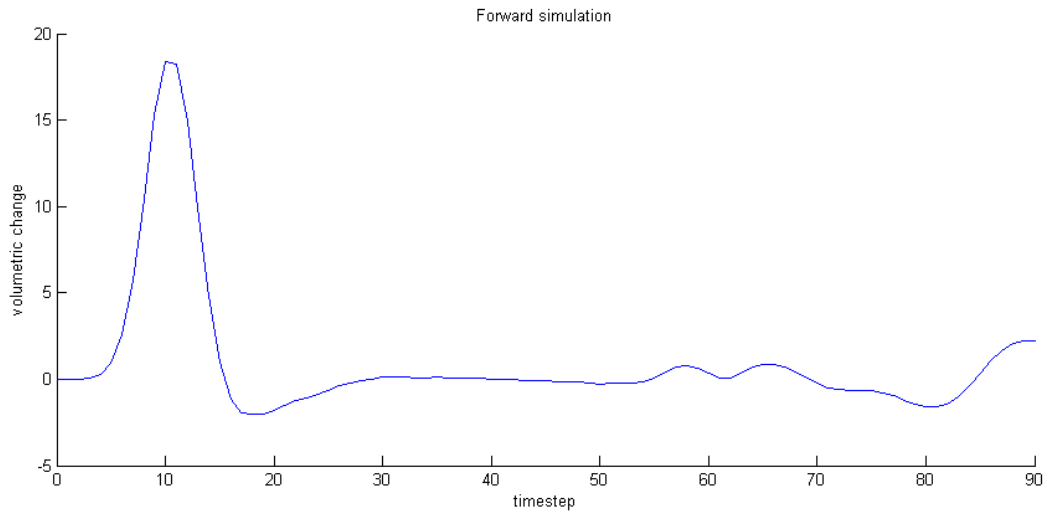


Figure 3.15: Solution of forward simulation with the wave equation at location where the wave was initiated (coordinates $x = 0.03$, $y = 0.03$, $z = 0.23$ [m]) with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$

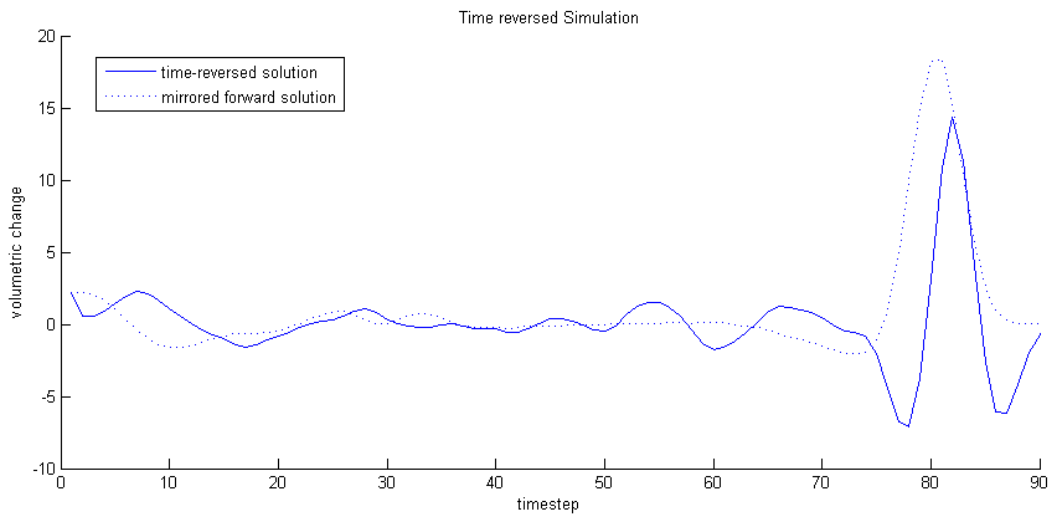


Figure 3.16: Solution of timereversed simulation with the wave equation compared to the mirrored forward simulation at location where the wave was initiated (coordinates $x = 0.03$, $y = 0.03$, $z = 0.23$ [m]) with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$

Chapter 4

Conclusions

The main task of this thesis was the localisation of an acoustic source with two different models. For this purpose the existing finite element code from [10] was extended to solve and postprocess the equation of motion and the wave equation in three-dimensional domains. An overview of the implemented functionalities can be extracted from attachment B.

Before starting with the implementation, the fundamental equations were introduced. Therefore, the strong and weak formulation of both equations were deduced in Chapter 2. Furthermore, the connection between both equations was shown. Then the finite element ansatz was explained and applied on the weak formulations of the equations. The solution of both equations varies in time wherefore the Newmark's scheme for temporal discretization was introduced and implemented. For later visualizations several postprocess functionalities were appended.

In a next step, the implementation was used to reproduce an experiment with different simulation setups. The experiment was performed by the Chair of Zerstörungsfreie Prüfung of the Technische Universität München [9]. The first arriving signal at different sensor locations could be reproduced. Due to a lack of precised information about the experimental setup the amplitude of the simulation signals could not be compared to the experimental results.

In the end, the wave propagation was simulated in reversed time. To show that waves can traced back to its origin the backward simulation is performed with the equation of motion and the wave equation. For the time-reversed simulation, information about the forward simulation has to be known. In this thesis, the backward simulation was performed knowing the solution at the end of the forward simulation in the whole domain. In general, it is not necessary to know the whole solutionfield to find the origin of the wave [2]. Further projects could investigate how far the information about the forward simulation can be reduced to perform a satisfactory backward simulation.

In conclusion, wave propagation can be simulated with the equation of motion and the wave equation. The main difference is the computed solutionfield. The reasons for that are the different fundamental concepts. The basis of the equation of motion is a force balance. A force in a certain direction can not be balanced with a force in an orthogonal direction. Therefore, in three dimensional domains the solution field has three components.

The wave equation solves for the volumetric change which is a scalar even if the computation domain is three-dimensional. Based on this fact, the computation with the equation of motion is more expensive, provided that the same discretization is applied on both equations. In the equation of motion, the solution is a vector with three components which shows up with three times more degrees of freedom.

As a last point, the post processing is an important issue. Especially the visualization of solution fields in three-dimensional domains is not trivial and exhausts the most part of the computation capacity. For visualizing the solution in three dimensions, the solution was saved in vtk-files and exported to the open-source application Paraview.

Based on the implementations that were accomplished in the course of this thesis, not only wave propagation but also general elasto-dynamic issues can be covered with the finite element code from [10].

Appendix A

Sensor signals of the experiment **E2** and synthetic sensor signals

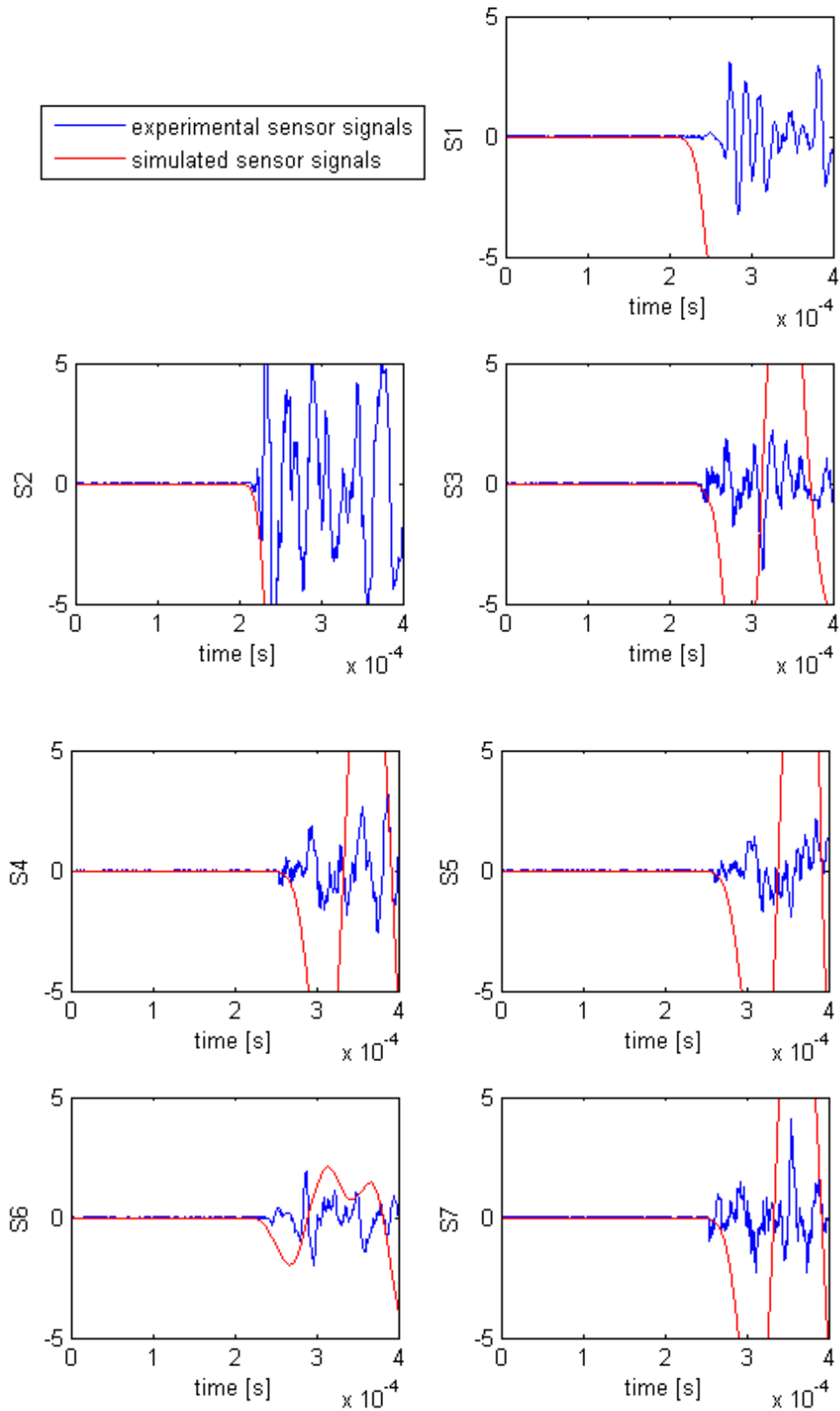


Figure A.1: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 1

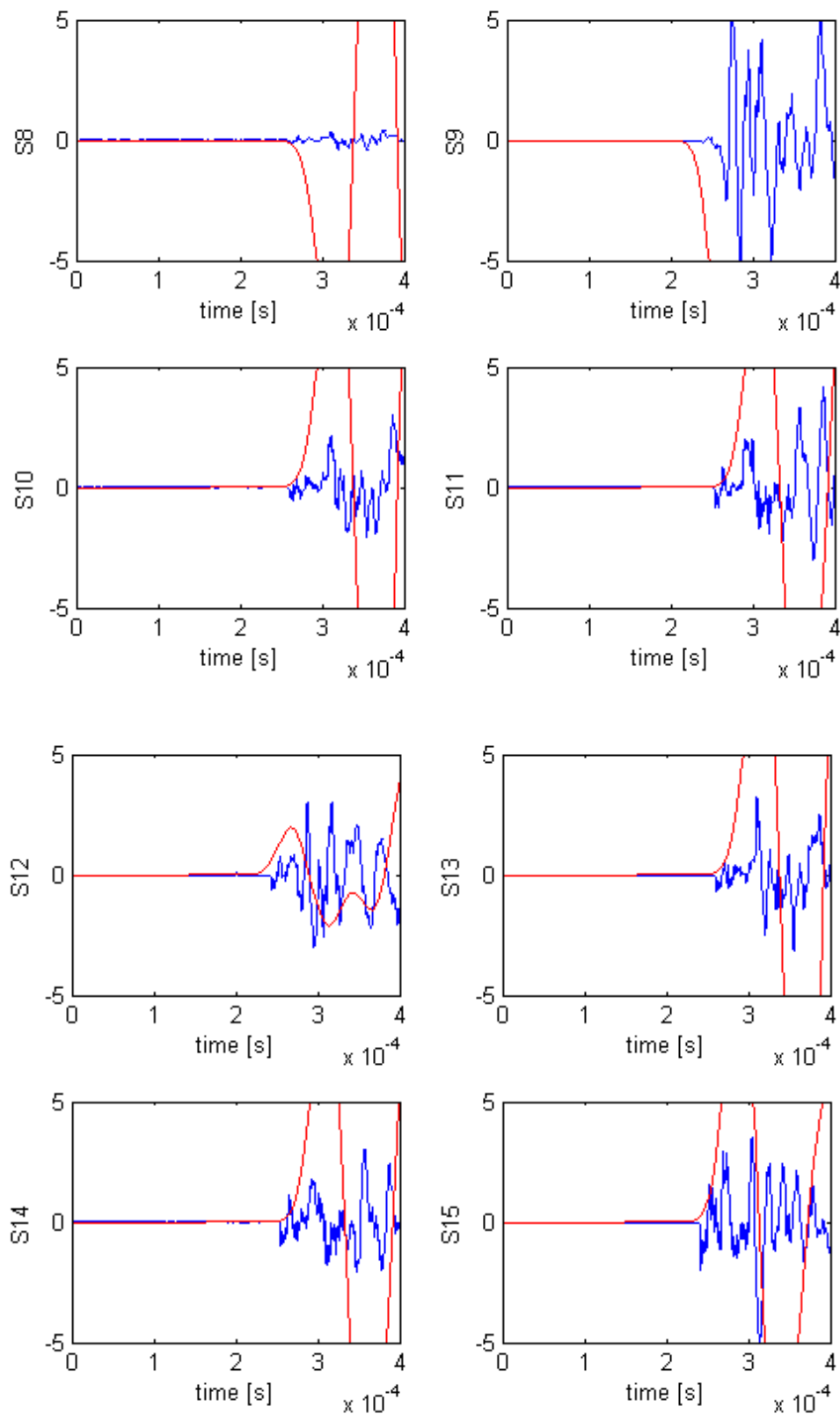


Figure A.2: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 1

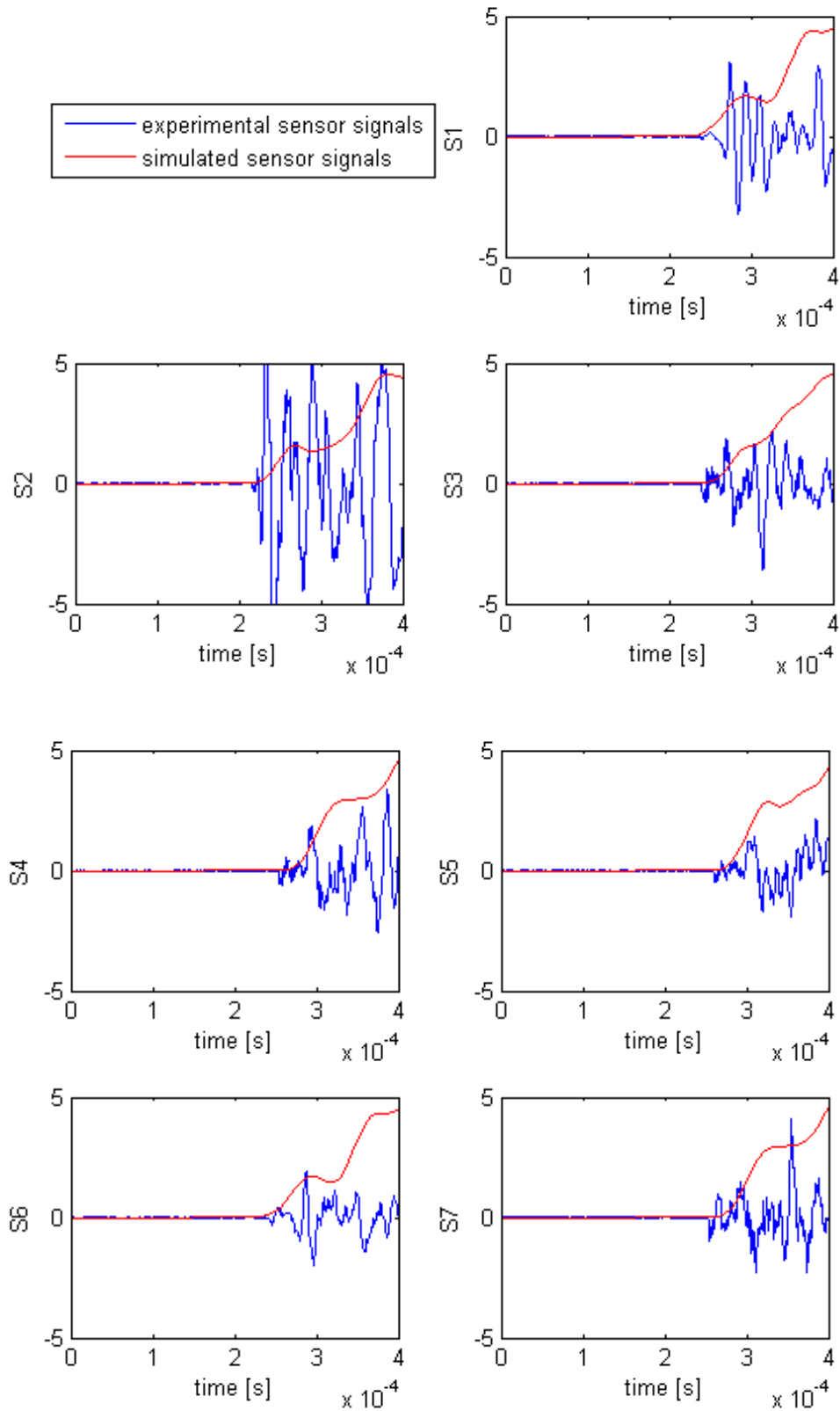


Figure A.3: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 2

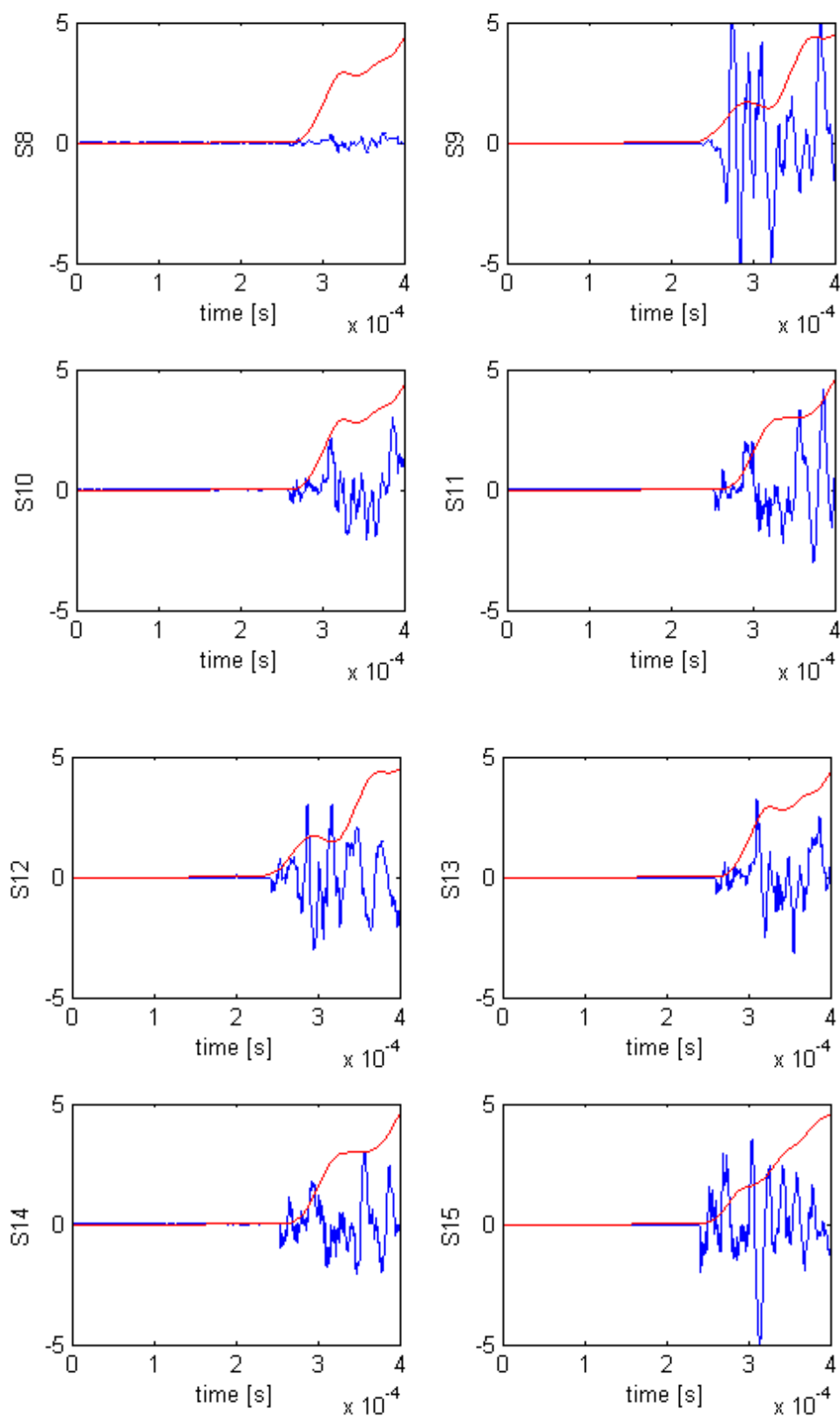


Figure A.4: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 2

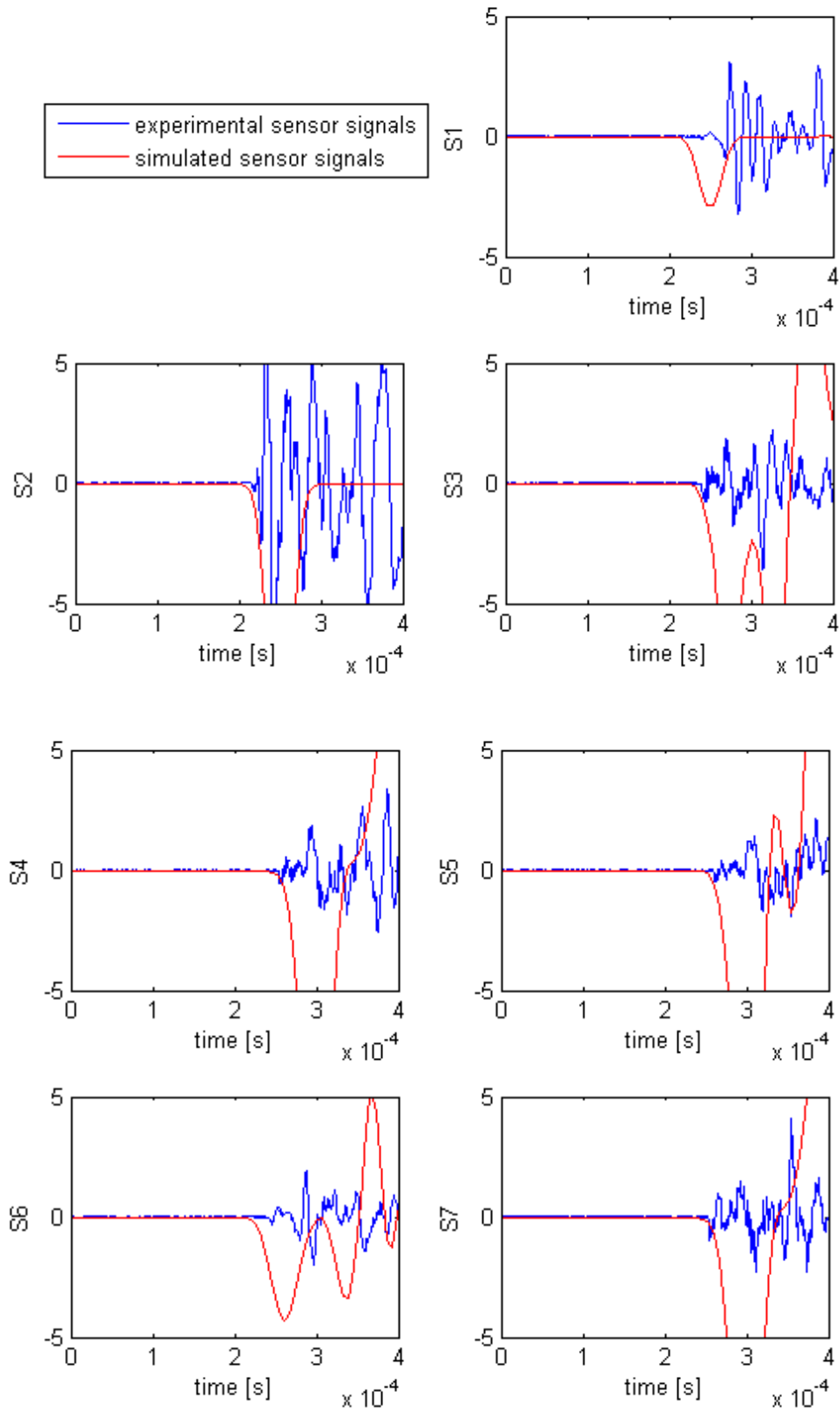


Figure A.5: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 3

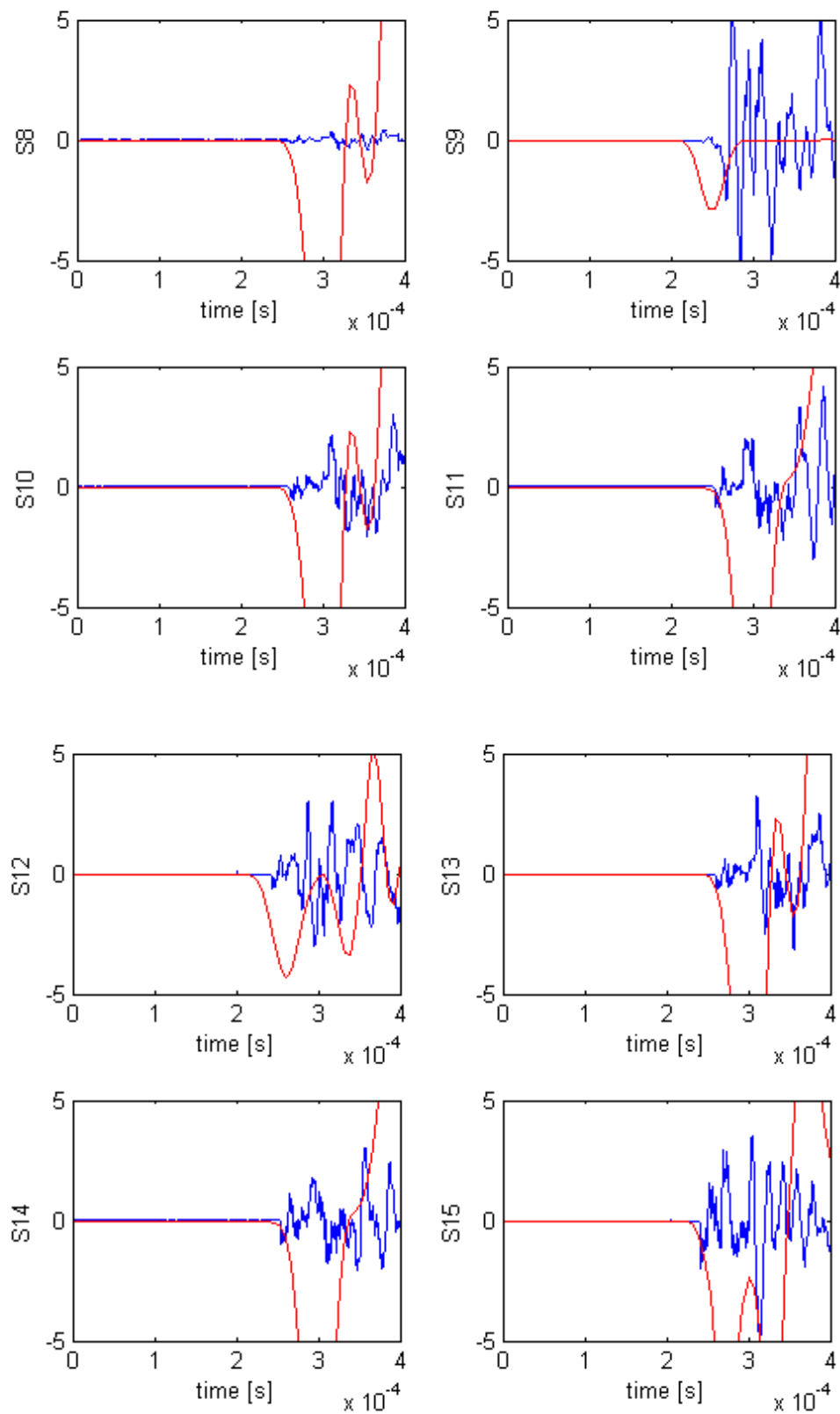


Figure A.6: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 3

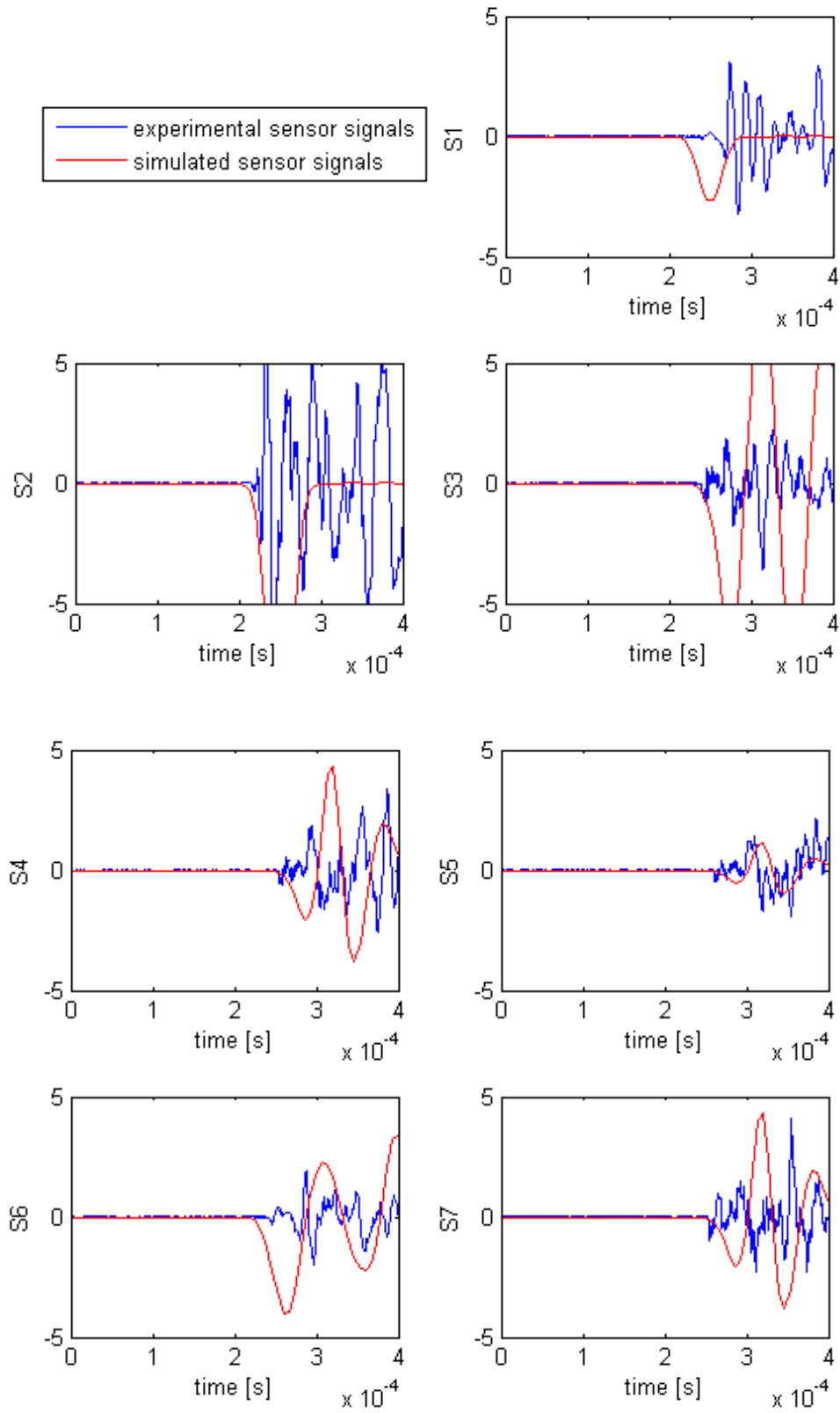


Figure A.7: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 4

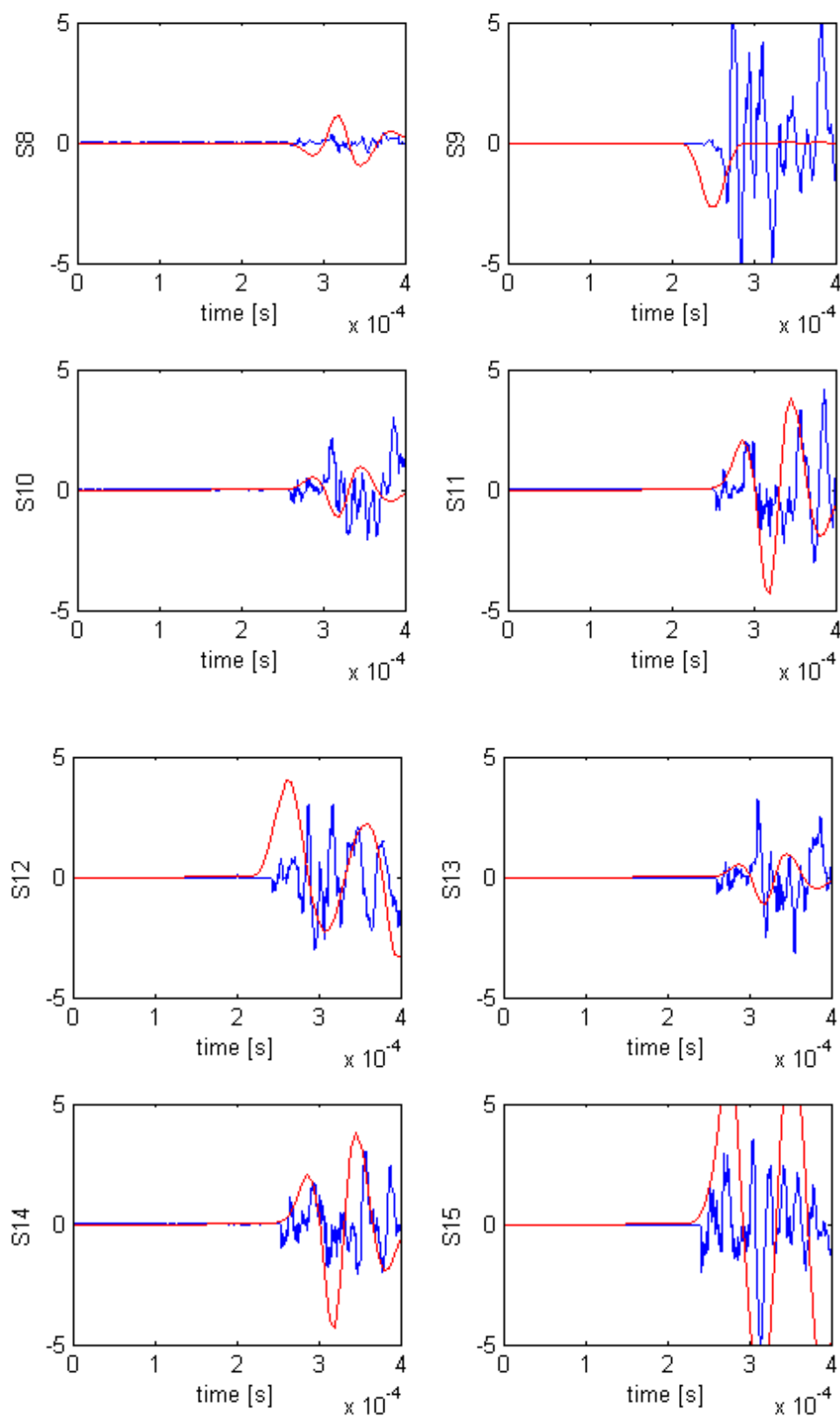


Figure A.8: Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 4

Appendix B

List of implemented functionalities

For a better understanding how the finite element code was extended, it is recommended to take a look at [10] to get a brief overview how the finite element code is hierarchically structured. The complete source code of the implementations can be inspected in attachment C.

Covering time-dependent problems in forward and backward simulations two timers were implemented with the following classes:

- TimeState (Time-handling)
- ReversedTimeState (Time-handling) .

The first expression is the name of the class and the expression in brackets describes the location in the structure of the whole finite element code, see [10].

Based on the fact that the time-dependencies were not covered in the first place, a class was written that integrates the time into originally time-independent functionalities:

- SpaceTimeFunction (Time-handling) .

After the time handling was added the Newmark's scheme was implemented:

- Newmark (Analysis) .

To use the already implemented functions for the calculation of mass matrices, stiffness matrices and load vectors, a new material with different material parameters was constructed:

- Hooke3DWaveEquation (Material) .

As mentioned previously, the wave equation solves for a scalar solution field, independent of the dimensions of the computation domain. For that reason, a new three-dimensional element type was installed:

- ScalarHexa (Element)
- ElementFactoryForScalarHexa (Element-Factory) .

Additionally, the post processing was added several functionalities.

In section 3.1, it is described that not only the solution but also the derivative of the solution is observed. For that reason, a new class that provides the ability of post processing the derivative of the solution had to be implemented:

- DerivativeSolutionVector (Point-Processor) .

To visualize the solution in three-dimensional domains, the open source tool Paraview was used. For that purpose, the following classes were implemented to save the computed solution field in VTK-files:

- VisualPostProcessingFactoryForVTK (Post-Processing-Factory)
- FEMesh3DforVTK (Post-Processing-Mesh-Factory)
- SolutionAsVTKVisualizer (Visualizer) .

For the reason, that the visualization of the solution in the whole domain could be very expensive the functionality to post process only single points in the domain was added:

- VisualPostProcessingFactoryForSensor (Post-Processing-Factory)
- OnlySome Points (Post-Processing-Mesh-Factory)
- SolutionAtSensors (Visualizer) .

Appendix C

Compact disc

The attached disc contains:

- source code of the implemented functionalities
- this documentation in pdf-format .

List of Figures

2.1	Stress resultants of an infinitesimal volume [8]	4
2.2	Stress balance in x-direction [8]	4
2.3	Decomposition of the domain into equally finite element, each element is discretized with two linear basis functions [8]	13
2.4	1D basis functions [10]	14
2.5	2D basis functions [10]	14
2.6	Structure of combined 1D-modes [10]	15
3.1	Different simulation setups with the equation of motion and the wave equation	25
3.2	Two-dimensional Gauss function in space with amplitude A_s , peak location (x_0, y_0) and standard deviation σ_s	26
3.3	Derivative of an one-dimensional Gauss function in time with amplitude A_t , peak location t_0 and standard deviation σ_t	26
3.4	Recorded signals of the simulation at coordinates [150 300 30] (at the back of the cube) with different spatial discretizations, polynomial degree $p=3$	28
3.5	Recorded signals of the simulation at coordinates [150 300 30] (at the back of the cube) with different spatial discretizations, polynomial degree $p = 3$	29
3.6	Experimental setup	30
3.7	Locations of the different sensors	31
3.8	Wave initiation for reproduction of the experiment, the load function is obtained by a multiplication of (a) and (b)	32
3.9	Interference of two waves	36
3.10	Wave reflections with Dirichlet boundary (top) and with natural Neumann boundary (bottom) on the right end of the rope, the wave travels from left to the right end and is then reflected and travels back	37
3.11	Setup of the forward simulation with the equation of motion	38

3.12	Time reversed simulation with the equation of motion, observing the resulting displacement with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$	39
3.13	Setup of the forward simulation with the wave equation	40
3.14	Forward simulation with the wave equation	40
3.15	Solution of forward simulation with the wave equation at location where the wave was initiated (coordinates $x = 0.03$, $y = 0.03$, $z = 0.23$ [m]) with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$	41
3.16	Solution of timereversed simulation with the wave equation compared to the mirrored forward simulation at location where the wave was initiated (coordinates $x = 0.03$, $y = 0.03$, $z = 0.23$ [m]) with a FEM-discretization with $5 \times 5 \times 5$ elements and polynomial degree $p = 3$	41
A.1	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 1	46
A.2	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 1	47
A.3	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 2	48
A.4	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 2	49
A.5	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 3	50
A.6	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 3	51
A.7	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 4	52
A.8	Comparison of experimental sensor signals compared to the synthetic sensor signals of the simulation with setup 4	53

List of Tables

2.1	Gauss points and weights for numerical integration	18
3.1	Location of the ultrasonic initiator in different experiments	32
3.2	Computation of the initiation time of the wave for experiment E2	33
3.3	Approximated material parameters	33
3.4	Error of the arriving signal of different setups, error = arriving signal of synthetic signal - arriving signal of experimental signal	35

Bibliography

- [1] Definition of a wave. <http://en.wikipedia.org/w/index.php?title=Wave&oldid=623392394>, Aug. 2014. Page Version ID: 623392394.
- [2] M. Fink. Time-reversed acoustics. *Scientific American*, Nov. 1999.
- [3] J. Fischer. Simulation of compression waves within human femurs using Finite-Element methods, Aug. 2013.
- [4] T. H. Heaton. Seismology. Lecture notes, Calofornia Institute Of Technology.
- [5] H. Kaltenbach. Grundlagen der numerischen strömungsmechanik. Lecture notes, Lehrstuhl für Aerodynamik und Strömungsmechanik, Technische Universität München, 2013.
- [6] E. Rank. Algorithms of partial differential equations. Lecture notes, Technische Universität München, 2013.
- [7] B. A. Szabó and I. Babuska. *Finite element analysis*. John Wiley & Sons, New York, 1991.
- [8] W. A. Wall and A. Popp. Computational solid dynamics (MSE). Lecture notes, Institut for Computational Mechanics, Technische Universität München, 2012.
- [9] U. Wengener. Lokalisierung von Abplatzungen aufgrund von Brandeinwirkung an kleinen Betonprobekörpern, Aug. 2013.
- [10] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, J. Parvizian, M. Ruess, D. Schillinger, and E. Rank. FCMLab: a finite cell research toolbox for MATLAB. *Advances in Engineering Software*, 74:49–63, Aug. 2014.