FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Homogenization of Rapidly Varying Coefficients in Seismic Simulations

Marius Bohn

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Homogenization of Rapidly Varying Coefficients in Seismic Simulations

Homogenisierung von stark variierenden Koeffizienten für seismische Simulationen

Author:	Marius Bohn
Supervisor:	Univ.-Prof. Dr. Michael Bader
Advisor:	Sebastian Wolf, M.Sc.
Date:	15.07.2022

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 15.07.2022 Marius Bohn

Acknowledgements

I would like to thank Prof. Dr. Michael Bader for giving me the opportunity to write this bachelor's thesis. I would also like to express my sincere thanks to my advisor Sebastian Wolf for his tireless commitment in teaching me the fundamentals of seismology, numerical quadrature and the SeisSol simulator. He was a great help in interpreting simulation results and presented ever new ideas for further extension and testing of the homogenization approach for material parameters.

Abstract

The accuracy of seismic simulations largely depends on a correct description of the media in which seismic waves propagate. Sampling material parameters only at each mesh element's barycenter supposedly leads to different kinds of systematic errors in the simulation output. Homogenization has proven to be an effective means of properly averaging distributions of rapidly varying coefficients over arbitrary areas or volumes. This thesis proposes a homogenization approach for distributions of material parameters in elastic, viscoelastic and poroelastic media. The technique is derived and evaluated on a variety of mesh geometries and parameter configurations. In general, it is found to be useful, even though some scenarios and coefficients do not obviously benefit from homogenization. Compared to the state of the art implementation in SeisSol, homogenization produces results of at least equal or superior accuracy at hardly any added cost. Especially in scenarios involving layered material distributions, the homogenized approach delivers outputs that are much more in line with the expected results.

Die Genauigkeit seismischer Simulationen hängt maßgeblich von einer korrekten Beschreibung der Materialien ab, in denen sich Erdbebenwellen ausbreiten. Werden die Materialparameter nur im Volumenschwerpunkt jedes Gitterelements abgetastet, ist mit verschiedenen Arten von systematischen Fehlern bei den Simulationen zu rechnen. Die Homogenisierung bietet sich als geeignetes mathematisches Instrument an, um Verteilungen von stark variierenden Koeffizienten über beliebige Flächen oder Volumina zu mitteln. In dieser Arbeit wird ein Ansatz zur Homogenisierung für Verteilungen von Materialparametern in elastischen, viskoelastischen und poroelastischen Materialien vorgestellt. Das Verfahren wird hergeleitet und auf verschiedenen Gittern und Konfigurationen der Parameter untersucht. Es erweist sich grundsätzlich als sinnvoll, wenngleich einige Szenarien und Koeffizienten nicht spürbar von der Homogenisierung profitieren. Im Vergleich mit der aktuellen Implementierung in SeisSol liefert die Homogenisierung Ergebnisse mit mindestens gleicher oder besserer Genauigkeit bei vernachlässigbaren zusätzlichen Kosten. Insbesondere bei unstetigen Verläufen der Materialparameter errechnet der homogenisierte Ansatz Resultate, die wesentlich besser mit den erwarteten Ergebnissen übereinstimmen.

Contents

1 Introduction

On a large time scale, the overall risk of suffering physical or material damage from the destruction caused by an earthquake at an arbitrary place is comparably low. However, in some geologically exposed areas, seismic activity occasionally leads to devastating losses of human life and property. Particularly in developing regions, this might be related to an insufficient political awareness of the risks associated with earthquakes or merely to a lack of financial resources to protect against it. Therefore, the natural hazard imposed by earthquakes is investigated by seismology in order to understand, model and predict the impacts of seismic activity. This provides a better baseline for evaluating different scenarios to guide investments and raise the awareness of hazards among the population.

The Japanese policy for the forecast of seismic events in the second half of the 20th century serves as an example of an unsuccessful attempt to predict the dynamics of earthquakes [SW03, Section 1.2]. Albeit serious doubts expressed by seismologists, a great effort has been put into establishing a mechanism to predict earthquakes which assumed that seismic activity is preceded by some recordable signs. At the latest when the severe quake in Kobe occurred without prior notice in 1995, it was clear that this theory is no longer tenable. Consequently, it is crucial to enhance the accuracy of seismic simulations wherever possible. Only then can they provide a reliable means of assessing the risks of living and building in a certain area.

On that note, this thesis introduces a homogenization technique for coefficients used to describe solid materials for the simulation of seismic wave propagation in SeisSol. The homogenization approach will be derived mathematically, implemented and evaluated based on actual simulation results. This includes a detailed comparison against the existing implementation. The idea for this concept goes back to LeVeque [LeV04, Section 9.14] who describes an approach for the homogenization of rapidly varying coefficients based on an analogy to acoustic media. He provides an example of wave propagation in discontinuous, layered materials which is also relevant for seismic applications.

In the context of seismic simulations, some important terms need to be defined upfront. A homogenization technique tries to form a proper average of an arbitrary coefficient c that is distributed over one or more dimensions [Sé20]. The distribution is usually assumed to be heterogeneous. Therefore, in the simplest case, a coefficient is sampled at N points x_i and averaged arithmetically. In real applications, often more complex ways of averaging a particular coefficient are required. Equation 1.1 [Sé20] shows how the sum converges to a volume integral for $N \to \infty$,

$$\frac{1}{N}\sum_{i=1}^{N} c(x_i) \xrightarrow[N\to\infty]{} \frac{1}{V}\int_V c(x)\,dx. \tag{1.1}$$

On a computer, however, it holds $N < \infty$ and a volume average needs to be approximated numerically by a weighted sum of discrete samples. The coefficients that are to be homogenized within this work describe certain properties of solid materials. Therefore, they will be referred to as material parameters throughout this thesis. The propagation of seismic waves will be discussed for elastic, viscoelastic and poroelastic media only.

A seismic source or fault denotes the location at which an earthquake originates. It may be approximated as point-shaped, known to be planar or have an irregular shape. A receiver is a spot at or below the earth's surface where seismic waves can be recorded.

Up to now, material parameters in SeisSol have been averaged by taking their value at each mesh element's barycenter. It is clear, however, that this technique does not take into account most of the actual material parameter distribution inside an element's volume. It is unable to resolve complex parameter profiles that may be rapidly changing within a single element. This provides reason to believe that the material parameters used for simulations in SeisSol might be subject to some systematic errors for certain configurations.

As a result, it is suspected that the output generated by SeisSol is not as accurate as it could be using homogenization. Thus, the approach proposed in this thesis has the potential to enhance the precision of seismic simulations. This allows to better understand how seismic waves propagate in certain media. As indicated in the beginning, more accurate predictions of hazards associated with earthquakes are crucial to reliably forecast risks for certain regions. Insurance providers, for example, would also benefit from a more precise base for calculating their rates and own risks.

Therefore, the research question this work tries to answer is whether homogenization of rapidly varying material parameters can increase the accuracy of seismic simulation outputs. Two hypotheses are formulated to help answer this question: First, it is suspected that barycenter averaging leads to systematic errors for certain distributions of material parameters. Moreover, we assume that homogenization can only deliver results of at least equal or superior accuracy than the state of the art. That is, also considering the higher computational effort, it should not be harmful to employ the homogenization approach.

The main matter is structured in four parts. Chapter 2 provides a theoretical background to the calculation and simulation of seismic wave propagation. It focuses on the elastic wave equations, how to derive and solve them. Seismic waves and different kinds of materials as well as the SeisSol simulator will also be detailed. Chapter 3 moves on to explain the numerical homogenization approach for material parameters. This includes both its mathematical derivation and implementation in SeisSol code using numerical quadrature. The simulation results obtained for different scenarios and parameter distributions using the novel approach will be compared against the base implementation and discussed in Chapter 4. This allows to draw certain conclusions about the hypotheses and the research question given above. Conclusions will be presented in Chapter 5.

With regard to the methodology employed for this work, especially numerical quadrature needs to be highlighted. It is a method to numerically compute integrals as found in the homogenization formulas. Section 3.3 gives a detailed account of quadrature. Furthermore, the homogenization approach will be implemented in SeisSol using C++ code. Working with SeisSol and analyzing seismograms generated by the simulator also forms an essential part of this thesis.

2 Simulation of Seismic Wave Propagation

Before proceeding to examine and evaluate the homogenization technique for material parameters, it will be necessary to review some fundamentals of seismic wave propagation and simulation. In the first section, the focus is on building a mathematical foundation of seismic wave theory. Section 2.2 explains a numerical solution to the elastic wave equations and their application in the SeisSol simulator.

2.1 Introduction to Seismic Wave Theory

The section below introduces to material elasticity and the corresponding concepts of stress and strain. Based on that, the elastic wave equations will be discussed and important properties of seismic waves will be presented. A reference of the different types of materials relevant for this thesis will also be given.

2.1.1 Stress and Strain

LeVeque [LeV04, Section 2.12] explains that when a solid undergoes a small deformation as a result of an external force, the theory of linear elasticity can be applied. It assumes that the restoring forces in the material are approximately linear to the degree of deformation. Correspondingly, a solid is called elastic if it returns to its original state after all forces have been removed. Larger deformations may cause chemical bonds inside the material to be rearranged. This leads to the solid yielding, resulting in a permanent or plastic deformation. Even higher load can result in structural failure of the material.

In the following, let us consider a point inside a three-dimensional solid material at a reference location (x_1, x_2, x_3). Following LeVeque [LeV04, Section 2.12], its location depending on the reference coordinates at time t is $(X_1(x_1, x_2, x_3, t), X_2(x_1, x_2, x_3, t), X_3(x_1, x_2, x_3, t))$. This allows to define the deformation vector

$$\boldsymbol{u}(x_1, x_2, x_3, t) = \begin{bmatrix} u_1(x_1, x_2, x_3, t) \\ u_2(x_1, x_2, x_3, t) \\ u_3(x_1, x_2, x_3, t) \end{bmatrix} = \begin{bmatrix} X_1(x_1, x_2, x_3, t) \\ X_2(x_1, x_2, x_3, t) \\ X_3(x_1, x_2, x_3, t) \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{2.1}$$

The resulting particle velocity $\boldsymbol{v}$ is the time derivative of the deformation,

$$\boldsymbol{v} = \frac{\partial \boldsymbol{u}}{\partial t}. \tag{2.2}$$

Deformation leads to extension or contraction within a solid which can be described as a strain. LeVeque [LeV04, Section 2.12] introduces the deformation gradient

$$\nabla \boldsymbol{u} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\ \frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3} \end{bmatrix} = \begin{bmatrix} \frac{\partial X_1}{\partial x_1} - 1 & \frac{\partial X_1}{\partial x_2} & \frac{\partial X_1}{\partial x_3} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} - 1 & \frac{\partial X_2}{\partial x_3} \\ \frac{\partial X_3}{\partial x_1} & \frac{\partial X_3}{\partial x_2} & \frac{\partial X_3}{\partial x_3} - 1 \end{bmatrix}. \tag{2.3}$$

Only the symmetric part of $\nabla \boldsymbol{u}$ will be considered in order to remove rotational motion of the solid. This results in the 3 x 3 strain tensor

$$\epsilon = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix}. \tag{2.4}$$

Strain causes stress which represents a material's effort to return to its original form as a result of the deformation. It can also be expressed in terms of a stress tensor

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{2.5}$$

where the first index i of σ_{ij} denotes the plane with normal vector $\boldsymbol{e}_i$ on which traction is exerted in the direction of $\boldsymbol{e}_j$ [SW03, Section 2.3]. Stein and Wysession [SW03, Section 2.3] move on to explain that $\sigma_{11}, \sigma_{22}, \sigma_{33}$ are called normal stresses as they describe traction normal to a plane. The stress tensor is symmetric and all components off the diagonal are known as shear stresses. These describe traction within a plane.
Figure 2.1 shows the components of the stress tensor attached to the respective planes of an infinitely small cube they act upon.

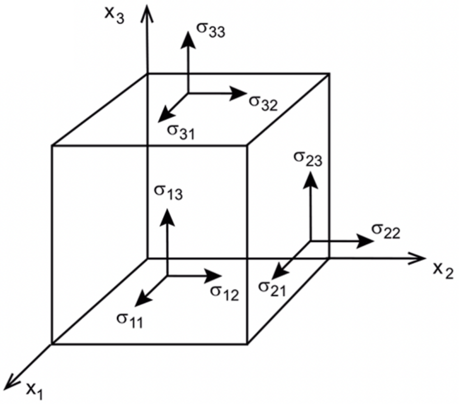


Figure 2.1: Components of the stress tensor at an infinitely small cube. Source: [AHH+21]

Stress relates to strain via Hooke's law which LeVeque [LeV04, Section 22.1] gives by

$$\sigma_{ij} = \sum_{k,l=1}^{3} C_{ijkl}\epsilon_{kl}. \tag{2.6}$$

Considering symmetry of the stress tensor and isotropic materials only, $\boldsymbol{C}$ simplifies to a 6 x 6 matrix [LeV04, Section 22.1]. It mediates between stress and strain via linear combinations of the Lamé parameters λ and μ as seen in Equation 2.7.

$$\begin{bmatrix}\sigma_{11}\\\sigma_{22}\\\sigma_{33}\\\sigma_{12}\\\sigma_{23}\\\sigma_{13}\end{bmatrix} = \begin{bmatrix}\lambda+2\mu & \lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0\\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0\\ 0 & 0 & 0 & 2\mu & 0 & 0\\ 0 & 0 & 0 & 0 & 2\mu & 0\\ 0 & 0 & 0 & 0 & 0 & 2\mu\end{bmatrix}\begin{bmatrix}\epsilon_{11}\\\epsilon_{22}\\\epsilon_{33}\\\epsilon_{12}\\\epsilon_{23}\\\epsilon_{13}\end{bmatrix} \tag{2.7}$$

2.1.2 Seismic Waves

Based on the ideas of stress and strain, the propagation of seismic waves in elastic media can be described. Therefore, the elastic wave equations will be derived for the three-dimensional case following LeVeque [LeV04, Section 22.1].

Using the definition of the strain tensor as well as Newton's second law, a set of equations for wave motion follows to be

$$\begin{aligned}\frac{\partial \epsilon_{11}}{\partial t} - \frac{\partial v_1}{\partial x_1} &= 0\\ \frac{\partial \epsilon_{22}}{\partial t} - \frac{\partial v_2}{\partial x_2} &= 0\\ \frac{\partial \epsilon_{33}}{\partial t} - \frac{\partial v_3}{\partial x_3} &= 0\\ \frac{\partial \epsilon_{12}}{\partial t} - \frac{1}{2}\left(\frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2}\right) &= 0\\ \frac{\partial \epsilon_{23}}{\partial t} - \frac{1}{2}\left(\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2}\right) &= 0\\ \frac{\partial \epsilon_{13}}{\partial t} - \frac{1}{2}\left(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1}\right) &= 0,\end{aligned} \tag{2.8}$$

$$\begin{aligned}\rho\frac{\partial v_1}{\partial t} - \frac{\partial \sigma_{11}}{\partial x_1} - \frac{\partial \sigma_{12}}{\partial x_2} - \frac{\partial \sigma_{13}}{\partial x_3} &= 0\\ \rho\frac{\partial v_2}{\partial t} - \frac{\partial \sigma_{12}}{\partial x_1} - \frac{\partial \sigma_{22}}{\partial x_2} - \frac{\partial \sigma_{23}}{\partial x_3} &= 0\\ \rho\frac{\partial v_3}{\partial t} - \frac{\partial \sigma_{13}}{\partial x_1} - \frac{\partial \sigma_{23}}{\partial x_2} - \frac{\partial \sigma_{33}}{\partial x_3} &= 0.\end{aligned} \tag{2.9}$$

The equations in 2.8 can be derived from the relation between ϵ and the deformation $\boldsymbol{u}$ defined in Equations 2.3 and 2.4. For the fourth equation, for example, the reasoning is

$$\begin{aligned} & \epsilon_{12} = \frac{1}{2}\left(\frac{\partial X_1}{\partial x_2} + \frac{\partial X_2}{\partial x_1}\right)\\ \Leftrightarrow \quad & \frac{\partial \epsilon_{12}}{\partial t} = \frac{1}{2}\left(\frac{\partial X_1}{\partial x_2 \partial t} + \frac{\partial X_2}{\partial x_1 \partial t}\right)\\ & = \frac{1}{2}\left(\frac{\partial u_1}{\partial x_2 \partial t} + \frac{\partial u_2}{\partial x_1 \partial t}\right)\\ & = \frac{1}{2}\left(\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_1}\right)\end{aligned} \tag{2.10}$$

The final step results from Equation 2.2. For the three equations in 2.9, we apply Newton's second law to an infinitely small element to obtain the net forces in all spatial directions as

$$F_i = \rho \frac{\partial v_i}{\partial t} = \sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_i}. \tag{2.11}$$

The equations of motion from 2.8 need to be inserted into the expressions for the six components of σ from Equation 2.7. Forming their time derivative results in

$$\begin{aligned}
\frac{\partial \sigma_{11}}{\partial t} - (\lambda + 2\mu)\frac{\partial v_1}{\partial x_1} - \lambda \frac{\partial v_2}{\partial x_2} - \lambda \frac{\partial v_3}{\partial x_3} &= 0 \\
\frac{\partial \sigma_{22}}{\partial t} - \lambda \frac{\partial v_1}{\partial x_1} - (\lambda + 2\mu)\frac{\partial v_2}{\partial x_2} - \lambda \frac{\partial v_3}{\partial x_3} &= 0 \\
\frac{\partial \sigma_{33}}{\partial t} - \lambda \frac{\partial v_1}{\partial x_1} - \lambda \frac{\partial v_2}{\partial x_2} - (\lambda + 2\mu)\frac{\partial v_3}{\partial x_3} &= 0 \\
\frac{\partial \sigma_{12}}{\partial t} - \mu \left(\frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2} \right) &= 0 \\
\frac{\partial \sigma_{23}}{\partial t} - \mu \left(\frac{\partial v_2}{\partial x_3} + \frac{\partial v_3}{\partial x_2} \right) &= 0 \\
\frac{\partial \sigma_{13}}{\partial t} - \mu \left(\frac{\partial v_1}{\partial x_3} + \frac{\partial v_3}{\partial x_1} \right) &= 0.
\end{aligned} \tag{2.12}$$

Equations 2.9 and 2.12 together are called the elastic wave equations.
LeVeque [LeV04, Section 2.12] recalls that there are two types of seismic waves, namely P and S waves which stands for either "primary" and "secondary" or "pressure" and "shear" waves. Their main difference is the direction of oscillation in relation to the direction the wave propagates in. A P wave compresses and stretches the material along the path it travels while S waves cause the material to move perpendicular to it [KBH02, Section 3.3]. This is illustrated in Figure 2.2.

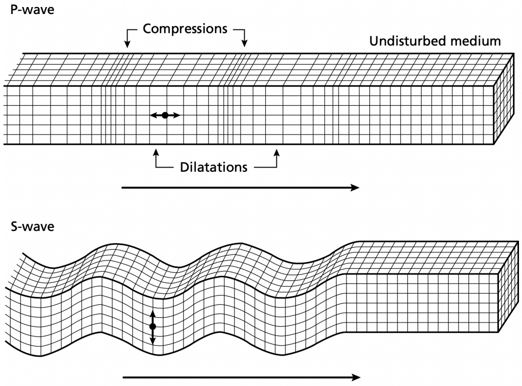
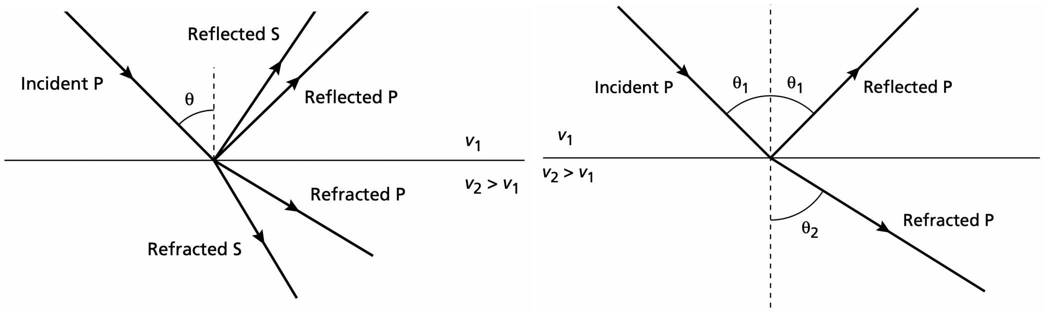


Figure 2.2: Oscillations caused by P and S waves relative to their direction of propagation. Source: [KBH02]

Both types of waves travel at different speeds which Kearey et al. [KBH02, Section 3.3] give by

$$v_p = \sqrt{\frac{K + \frac{4}{3}\mu}{\rho}} = \sqrt{\frac{\lambda + 2\mu}{\rho}}, v_s = \sqrt{\frac{\mu}{\rho}}. \tag{2.13}$$

The velocity of S waves is independent of the bulk modulus K because an S wave causes no compression [SW03, Section 2.4]. In general, P waves travel faster than S waves because the elastic moduli K and μ are usually positive. The resulting difference in time between the arrivals of P and S waves at receivers allows to locate the source of an earthquake.
When a wave hits an interface in a layered medium, a reflected and a transmitted wave are generated [KBH02, Section 3.6]. For P waves, additional reflected and transmitted S waves occur as shown in Figure 2.3.



(a) Reflected and transmitted P and S waves for P wave hitting a material interface.

(b) Reflected and transmitted P waves and associated incidence angles.

Figure 2.3: Reflected and transmitted P and S waves generated when a P wave hits a material interface. Source: [KBH02]

Kearey at al. [KBH02, Section 3.6] argue that the directions of the newly generated waves follow Snell's law. For the relation between angles enclosed with the vertical and speeds in two neighboring layers it holds that

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2}. \tag{2.14}$$

As a result, the P wave is reflected at the same angle at which it hit the interface. Both the reflected and transmitted S waves have a lower angle enclosed with the vertical than the incoming P wave due to their overall lower velocity. The angle of the refracted P wave depends on v_2.

2.1.3 Seismic Faults

An earthquake generally originates at a fault. A fault is an area of fractures between neighboring solid blocks in the ground [Uni22]. This allows the blocks to move along the fault. It is characterized by the angle it encloses with the earth's surface, called the dip, and the direction in which movement occurs along the fault, named slip. Using these definitions, faults can be divided into three categories [Uni22]:

1. Normal fault: Downward slip of the block above the fault

2. Reverse (thrust) fault: Upward slip of the block above the fault
3. Strike-slip fault: Both blocks slide horizontally along the fault

Stein and Wysession [SW03, Section 2.3] note that, as a good approximation, a fault originates along the plane with the highest shear stress in a solid.

2.1.4 Materials and Parameters

In the following, the materials a solid may be composed of will be divided into several categories based on their macroscopic structure. The kinds of materials relevant for this thesis are elastic, viscoelastic and poroelastic media. Plastic and anisotropic materials will be briefly addressed in Chapter 5. They will not be investigated further.

Elastic Materials

As was mentioned in Subsection 2.1.1, purely elastic media behave according to the theory of linear elasticity. Like a spring in the one-dimensional case, a three-dimensional elastic solid experiences restoring forces linear to its deformation [LeV04, Section 2.12]. The elastic approximation is valid only up to a certain point beyond which any material suffers lasting deformation.

Elastic media can be described by means of elastic moduli. These express different relations of certain components of stress to the resulting strain. Kearey et al. [KBH02, Section 3.2] provide definitions of the elastic moduli:

- Young's modulus E describes elongation of a material when a force is applied in longitudinal direction. This is accompanied by tapering of the solid perpendicular to the direction of longitudinal strain. Poisson's ratio ν relates lateral and longitudinal strain for this mode of deformation.
- Under pressure, a volume is compressed along all axes. Here the bulk modulus K gives the ratio of volumetric stress and strain.
- The second Lamé parameter μ is also called shear modulus. It captures how rigidly a material responds when a shear stress is applied [SW03, Section 2.3].
- The axial modulus Ψ is less relevant. Similar to Young's modulus, it relates longitudinal stress and strain, but without lateral contraction.

Figure 2.4 provides a vivid description of the elastic moduli.

Stein and Wysession [SW03, Section 2.3] explain that in seismic applications, however, the properties of elastic media are mostly characterized by the Lamé parameters λ and μ only. Together with density ρ, they provide a complete description of elastic behavior. The first Lamé parameter λ has no physical interpretation. According to Stein and Wysession, it relates to the other elastic moduli via

$$\lambda = K - \frac{2}{3}\mu = \frac{E\nu}{(1+\nu)(1-2\nu)}. \tag{2.15}$$

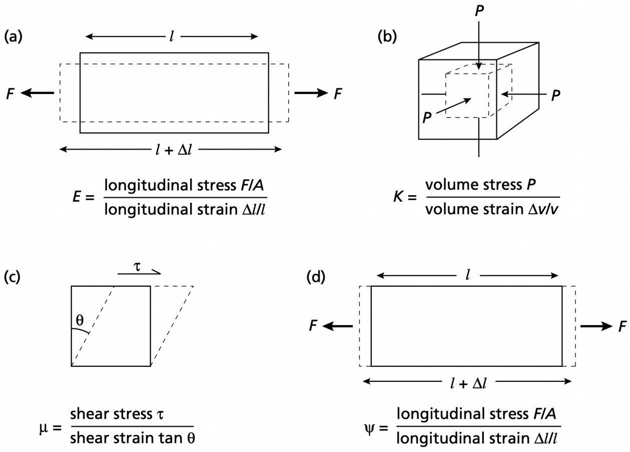

(a)
l
F
F
l + Δl
E = longitudinal stress F/A / longitudinal strain Δl/l
(b)
P
P
P
K = volume stress P / volume strain Δv/v
(c)
τ
θ
μ = shear stress τ / shear strain tan θ
(d)
l
F
F
l + Δl
ψ = longitudinal stress F/A / longitudinal strain Δl/l

Figure 2.4: Visualizations of the elastic moduli. (a) Young's modulus, (b) bulk modulus, (c) shear modulus, (d) axial modulus. Source: [KBH02]

Viscoelastic Materials

Viscoelastic media combine viscous and elastic characteristics. In seismology, the description of viscoelasticity relies on the elastic parameters λ, μ and ρ as well as on additional quality factors Q_p and Q_s for P and S waves, respectively. They are also known as "seismic Q".
Morosov [Mor19] argues that multiple interpretations and analogies exist to capture the meaning of seismic Q. They all agree on the fact that Q_p and Q_s are used to model the attenuation of seismic waves. The most vivid definition is that of "intended interpretational Q" [Mor19]. It identifies the need to consider wave decay in viscoelastic media as a result of heterogeneities like fractures, aquifers or different temperatures in the material. This causes some of the energy contained in seismic waves to be dissipated by friction.
We speak of "intended" because this interpretation is not helpful to quantify seismic Q. No physical quantity exists to measure for example the degree of fracturing of a rock. Therefore, other theories exist which assign the quality factor a mathematical meaning. Morosov names an analogy to a resonator or an interpretation of Q as the complex argument of the bulk modulus K in the frequency domain. It is difficult, however, to make assumptions about the frequency domain prior to an actual simulation run.

Poroelastic Materials

Porous media can be described by the theory of poroelasticity. Poroelastic materials have both a solid and a fluid component. In contrast to viscoelastic media, the viscous phase is always arranged regularly in form of a pore space that permeates the solid and contains a fluid [WGU$^+$22, Section 3.1].
Wolf et al. [WGU$^+$22, Section 3.1] list the parameters used to describe poroelastic materials. The densities of the solid and the fluid phase, ρ_S and ρ_F, are considered separately. The same applies to the bulk moduli K_S and K_F. The Lamé parameters λ_M and μ_M govern the elastic behavior of the solid part. It is also referred to as the solid matrix, hence the subscript M. Wolf et al. identify four more coefficients specific to poroelastic media:

- The matrix permeability κ describes the resistance the fluid faces as it passes through the pores.

- Matrix porosity Φ denotes the share of the entire volume taken up by the pores.
- Two points in the fluid phase are not connected via a straight link but through the pores which have an irregular shape. Therefore, the matrix tortuosity T captures the ratio of the length of a connection via pores to the direct link between two points in the fluid medium.
- Fluid viscosity ν is a well known parameter used to describe how rigidly a liquid responds to deformation.

Overview of Materials and Parameters

The table below provides a reference to the material parameters actually used for the description of different media in SeisSol. For elastic materials, for example, only density and both Lamé parameters are used. As discussed above, these are sufficient to fully describe elastic behavior.

Parameter	Description	Elastic	Viscoelastic	Poroelastic
ρ	Density	●	●	●
λ	1. Lamé Parameter	●	●	●
μ	2. Lamé Parameter	●	●	●
$Q_{p/s}$	Seismic Q		●	
K	Bulk Modulus			●
κ	Matrix Permeability			●
Φ	Matrix Porosity			●
T	Matrix Tortuosity			●
ν	Fluid Viscosity			●

Table 2.1: Overview of material parameters and their relevance for elastic, viscoelastic and poroelastic media. For the latter, density and bulk modulus are considered separately for the fluid and solid phases of a medium.

2.2 Solution to the Elastic Wave Equations

Turning now to the usage of the elastic wave equations in seismic simulations. The first subsection presents the ADER-DG scheme as a numerical solution to the elastic wave equations. It is followed by an introduction to the SeisSol simulator and its interaction with the easi library used to initialize material parameters.

2.2.1 ADER-DG Numerical Scheme

In order to simulate seismic wave propagation, a numerical solution to the elastic wave equations is required. Following Dumbser and Käser [DK06], the Arbitrary high-order DERivatives (ADER) approach combined with the Discontinuous Galerkin (DG) discretization scheme will be applied to the three-dimensional elastic wave equations. The ADER-DG approach allows for an arbitrarily high accuracy of the solution. It is a single-step scheme and does not depend on intermediate calculations like for example Runge-Kutta. Since its

entire derivation is rather complex, ADER-DG will be presented in a simplified way.
At the heart of DG is its ability to handle discontinuities at the boundaries between neighboring elements [KP21, Chapter 1]. It introduces the concept of numerical fluxes between elements to compensate for the error that would otherwise occur without forcing continuity. Dumbser and Käser move on to explain that the DG approach approximates the solution for each tetrahedral element by means of polynomial basis functions $\Phi_l(x)$ up to degree N. For the numerical solution Q to the elastic wave equations it holds

$$Q(x,t) = \sum_{l=1}^{N} \hat{Q}_l(t)\Phi_l(x) \tag{2.16}$$

where Q is approximated as a linear combination of time-independent basis functions $\Phi_l(x)$ with space-independent degrees of freedom $\hat{Q}_l(t)$.
The elastic wave equations can also be expressed as a single term

$$\frac{\partial Q}{\partial t} + A\frac{\partial Q}{\partial x_1} + B\frac{\partial Q}{\partial x_2} + C\frac{\partial Q}{\partial x_3} = 0 \tag{2.17}$$

using matrices A, B and C. We multiply above equation by a test function Φ_k and form its integral over a tetrahedral volume V, which yields

$$\begin{aligned} &\int_V \Phi_k \left(\frac{\partial Q}{\partial t} + A\frac{\partial Q}{\partial x_1} + B\frac{\partial Q}{\partial x_2} + C\frac{\partial Q}{\partial x_3}\right) dV = 0 \\ \Leftrightarrow \quad &\int_V \Phi_k \frac{\partial Q}{\partial t}\, dV + \int_V \Phi_k \left(A\frac{\partial Q}{\partial x_1} + B\frac{\partial Q}{\partial x_2} + C\frac{\partial Q}{\partial x_3}\right) dV = 0. \end{aligned} \tag{2.18}$$

Equation 2.18 is integrated by parts while we introduce a numerical flux Ψ_k as explained above to obtain

$$\int_V \Phi_k \frac{\partial Q}{\partial t}\, dV + \int_{\partial V} \Phi_k \Psi_k \, dS - \int_V \frac{\partial \Phi_k}{\partial x_1} AQ + \frac{\partial \Phi_k}{\partial x_2} BQ + \frac{\partial \Phi_k}{\partial x_3} CQ \, dV = 0. \tag{2.19}$$

Inserting an expression for the flux Ψ_k between two neighboring elements leads to a large term for the DG formulation which we omit here for the sake of brevity. It is computationally efficient to substitute the irregular tetrahedral volume by that of a reference tetrahedron. This allows to evaluate and store some of the integrals contained in the DG expression once for an entire simulation run.
Additionally, the ADER time-discretization needs to be performed. By applying the Cauchy-Kowalevski procedure in the transformed $\xi\eta\zeta$-system with rearranged matrices A^*, B^* and C^*, one obtains

$$\frac{\partial Q}{\partial^k t} = (-1)^k \left(A^* \frac{\partial}{\partial \xi} + B^* \frac{\partial}{\partial \eta} + C^* \frac{\partial}{\partial \zeta}\right)^k Q. \tag{2.20}$$

Dumbser and Käser develop the solution to Equation 2.17 in a Taylor series which yields

$$Q(x,t) = \sum_{k=0}^{N} \frac{t^k}{k!} \frac{\partial Q(x,0)}{\partial^k t}. \tag{2.21}$$

Inserting Equation 2.20 into Equation 2.21, the time-derivatives can be replaced by space-derivatives. Moreover, we insert Equation 2.16 to get the result

$$\begin{aligned} Q(x,t) &= \sum_{k=0}^{N} \frac{t^k}{k!}(-1)^k \left(A^* \frac{\partial}{\partial \xi} + B^* \frac{\partial}{\partial \eta} + C^* \frac{\partial}{\partial \zeta} \right)^k Q(x,0) \\ &= \sum_{k=0}^{N} \frac{t^k}{k!}(-1)^k \left(A^* \frac{\partial}{\partial \xi} + B^* \frac{\partial}{\partial \eta} + C^* \frac{\partial}{\partial \zeta} \right)^k \hat{Q}_l(0)\Phi_l(x). \end{aligned} \tag{2.22}$$

The final idea is to evaluate the Taylor series from Equation 2.22 at time $t + \tau$ to obtain a solution for an advanced time step τ.

2.2.2 SeisSol and easi

The homogenization approach for material parameters developed in this thesis is implemented in the SeisSol simulation package. SeisSol is a "scientific software for the numerical simulation of seismic wave phenomena and earthquake dynamics" [Tea22d]. At the heart of SeisSol is an implementation of the ADER-DG solution to the elastic wave equations presented in the previous subsection.
The corresponding documentation [Tea22c] highlights its most important features: The software uses meshes composed of tetrahedral elements to model highly complex three-dimensional topography and seismic fault geometries. The ADER-DG approach allows for an arbitrarily high order of accuracy in both space and time. SeisSol can handle all kinds of elastic as well as plastic and anisotropic materials. Two- and three-dimensional output can be generated and visualized for the evolution of multiple parameters over time. If receivers are defined, SeisSol generates artificial seismograms at their locations. The SeisSol code is open-source and available on GitHub [Tea22b].

SeisSol is designed for a highly parallelized execution, typically on high-performance computing clusters. Therefore, it relies on MPI combined with OpenMP to form a hybrid parallelization scheme [Tea22c]. The Message Passing Interface (MPI) is a standard that defines communication between multiple processes [BM06, Section 3.3]. The processes exchanging messages may be part of a single machine and / or distributed over multiple nodes. This can be used as a means of implicit synchronization to split up and coordinate the execution of tasks between different processes [Mar21].
OpenMP, on the contrary, is an API which allows to instruct a suitable compiler to apply multi-threading to certain parts of code via dedicated pragmas [BM06, Section 3.4]. Apart from threading features integral to a programming language's standard library, a program may be parallelized using either MPI or OpenMP exclusively. Under certain circumstances, however, it can be beneficial to combine both concepts and form a hybrid approach [INT17, Section 1]. This is possible by connecting multiple nodes via a common MPI interface and simultaneously parallelizing the tasks executed on every single node using OpenMP directives.
A noticeable advantage of combining MPI with OpenMP is a reduced memory consumption because of the administrative overhead required by MPI to keep processes aware of the data handled by others [INT17, Section 2.1]. Replacing MPI processes on a node by OpenMP

threads which share their memory can help reduce the overhead. Moreover, it is possible to achieve better scaling performance at high node counts with hybrid MPI and OpenMP for some applications [INT17, Section 2.2]. Nonetheless, scaling of this approach is also limited at some point due to the overhead caused by parallelization. SeisSol shows high scaling performance for realistic scenarios on modern HPC systems [KUU^{+}21, Section 6.3].
Like any type of parallel programming, a successful usage of OpenMP and MPI requires code that is suitable for parallelization in the first place. A program needs to be studied carefully to avoid for example race conditions or idle OpenMP threads consuming performance in sections which can only be run sequentially anyway [INT17, Section 3].

SeisSol uses the easi library to initialize a multi-dimensional rheological model. The easi documentation [Uph22] explains how a model is generated: The overall aim is to assign a vector of material parameters to an arbitrary point in space. Applied to SeisSol, this means that prior to the actual simulation every mesh element needs to be assigned a vector of parameters. It is used as a constant description of the material characteristics for the entire volume occupied by an element. easi allows to define and connect maps and filters in YAML files. A combination of them is used to model a function which returns material parameter values for certain points in space.
Listing 2.1 contains an exemplary easi model to showcase the use of maps and filters with the help of the easi documentation [Uph22]. The Any filter in line 1 accepts all points and represents a root node. The GroupFilter only allows points which are tagged with 6,7 or 8 in the mesh. These are assigned constant values for the elastic material parameters ρ, μ and λ using a ConstantMap. All other points are processed in the FunctionMap in line 10. It defines ImpalaJIT [Fas22] functions for all three material parameters depending on a point's coordinates. More types of maps and filters are available.

```
 1  !Any
 2  components:
 3    - !GroupFilter
 4      groups: [6,7,8]
 5      components: !ConstantMap
 6        map:
 7          rho:      2500.0
 8          mu:          0.0
 9          lambda:  1.96e10
10    - !FunctionMap
11      map:
12        rho:     return 1600. + 59.5*pow(y,1./3.);
13        mu:      return 0.;
14        lambda:  return pow(260. + 30*sqrt(y), 2.) * (1600. + 59.5*pow(y,1./3.));
```

Listing 2.1: easi example 5_function.yaml. Source: [Tea22a]

For later reference it should be noted that SeisSol interacts with easi models by means of corresponding classes. The easi::Component class represents an easi model loaded and parsed from a YAML file. A seissol::initializers::QueryGenerator defines the points at which material parameters should be retrieved. They are stored in an easi::Query. An easi::Component can be evaluated using the query which yields the desired output of material parameters.

3 Numerical Homogenization Approach

In the following pages, the mathematical background developed for the homogenization of elastic and viscoelastic material parameters alongside its implementation in SeisSol will be presented. Poroelasticity will be covered in theory only.
First, the existing implementation used for averaging parameters will be given as a reference, followed by a detailed account of the newly derived homogenization approach. Furthermore, an introduction to Gaussian quadrature in the context of evaluating integrals for homogenization is provided. These theoretical considerations form the basis for application of the actual homogenization technique in SeisSol code in the final section.

3.1 Current Approach

As indicated previously, material parameters used to be sampled at a single point per element, namely at its barycenter $\boldsymbol{b}$. For non-regular tetrahedra with vertices $\boldsymbol{v_0} = [x_0, y_0, z_0]^T$, $\boldsymbol{v_1} = [x_1, y_1, z_1]^T$, $\boldsymbol{v_2} = [x_2, y_2, z_2]^T$, $\boldsymbol{v_3} = [x_3, y_3, z_3]^T$ as encountered in terrain meshes, the barycenter location can be calculated as

$$\boldsymbol{b} = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} \frac{x_0+x_1+x_2+x_3}{4} \\ \frac{y_0+y_1+y_2+y_3}{4} \\ \frac{z_0+z_1+z_2+z_3}{4} \end{bmatrix}. \tag{3.1}$$

While this approach delivers exact results for homogeneous media, it is subject to errors of increasing severity for any more complex parameter distribution. It will be shown in Chapter 4 that such cases include, among others, layered materials with discontinuities and some parameter distributions described by trigonometric functions.

```
easi::Query seissol::initializers::ElementBarycentreGenerator::generate() const {
  std::vector<Element> const& elements = m_meshReader.getElements();
  std::vector<Vertex> const& vertices = m_meshReader.getVertices();

  easi::Query query(elements.size(), 3);
  for (unsigned elem = 0; elem < elements.size(); ++elem) {
    // Compute barycentre for each element
    for (unsigned dim = 0; dim < 3; ++dim) {
      query.x(elem,dim) = vertices[ elements[elem].vertices[0] ].coords[dim];
    }
    for (unsigned vertex = 1; vertex < 4; ++vertex) {
      for (unsigned dim = 0; dim < 3; ++dim) {
        query.x(elem,dim) += vertices[ elements[elem].vertices[vertex] ].coords[dim];
      }
    }
    for (unsigned dim = 0; dim < 3; ++dim) {
      query.x(elem,dim) *= 0.25;
    }
    /*...*/
  }
  return query;
}
```

Listing 3.1: Generation of an easi query using barycenter sampling. Source: SeisSol, ParameterDB.cpp, ll. 59-81

Listing 3.1 shows calculation of the barycenter of all elements contained in a terrain mesh following Equation 3.1. The results are entered into a new easi query which is returned. Upon evaluation of the easi query at a later stage, parameter values are fetched and stored in the material parameter database without further processing.
In contrast to this rather simple yet often imprecise approach, a more sophisticated technique for averaging material parameters will be presented in the next section.

3.2 Homogenization Approach

In the following, we will move on to describe in greater detail how the steps required for homogenization of material parameters in elastic, viscoelastic and poroelastic media can be derived. The respective parameters to be averaged have been introduced in Subsection 2.1.4.

3.2.1 Elastic Media

In the case of elastic materials, only three material parameters need to be averaged, namely the density ρ as well as the Lamé parameters λ and μ. A bar above a coefficient denotes a parameter's homogenized value.
For homogenization of ρ within a mesh element, LeVeque [LeV04, Section 9.14] and Moczo et al. [MKV$^+$02, p. 6] consider it sufficient to use its arithmetic average to obtain the average material density

$$\bar{\rho} = \frac{1}{|V|} \int_V \rho(x)\, dx. \tag{3.2}$$

LeVeque uses an analogy to acoustic wave propagation. Therefore, he proposes an approach using a harmonic average for the bulk modulus K, which can also be applied to the shear modulus μ [LeV04, Section 9.14]. In order to show that this substitution is valid, let us consider the stress-strain relation

$$\sigma = K\epsilon \tag{3.3}$$

which rewrites to Hooke's law as

$$\epsilon = \frac{1}{K}\sigma. \tag{3.4}$$

LeVeque applies an arithmetic average to the strain ϵ within the integral boundaries $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. Replace this by an arbitrary tetrahedron volume V and use Equation 3.4 to obtain

$$\begin{aligned} \bar{\epsilon} &= \frac{1}{|V|} \int_V \epsilon(x)\, dx \\ &= \left(\frac{1}{|V|} \int_V \frac{1}{K(x)}\, dx \right) \sigma \\ &= \frac{1}{\bar{K}} \sigma. \end{aligned} \tag{3.5}$$

The three-dimensional extensional and shear stress-strain relations for elasticity are also given by LeVeque [LeV04, Section 22.1]:

$$\begin{aligned} \epsilon_{11} &= \frac{1}{E}\sigma_{11} - \frac{\nu}{E}\sigma_{22} - \frac{\nu}{E}\sigma_{33} \\ \epsilon_{22} &= -\frac{\nu}{E}\sigma_{11} + \frac{1}{E}\sigma_{22} - \frac{\nu}{E}\sigma_{33} \\ \epsilon_{33} &= -\frac{\nu}{E}\sigma_{11} - \frac{\nu}{E}\sigma_{22} + \frac{1}{E}\sigma_{33}, \end{aligned} \tag{3.6}$$

$$\epsilon_{12} = \frac{1}{2\mu}\sigma_{12},\ \epsilon_{23} = \frac{1}{2\mu}\sigma_{23},\ \epsilon_{13} = \frac{1}{2\mu}\sigma_{13}. \tag{3.7}$$

With the same reasoning as for acoustics in Equation 3.5 as well as the analogy of Equation 3.4 to Equation 3.7 apart from a constant factor of $\frac{1}{2}$, it is justified to apply a harmonic average to μ to obtain the average shear modulus

$$\frac{1}{\bar{\mu}} = \frac{1}{|V|} \int_V \frac{1}{\mu(x)}\, dx. \tag{3.8}$$

It is slightly more complex to derive a homogenized first Lamé parameter λ. LeVeque [LeV04, Section 22.1] recalls the relationship between shear modulus μ, Young's modulus E and Poisson ratio ν to be

$$\mu = \frac{E}{2(1+\nu)} \tag{3.9}$$

$$\Leftrightarrow \frac{1}{2\mu} = \frac{1+\nu}{E}. \tag{3.10}$$

Rephrase Equation 3.6 and use Equation 3.10 to obtain

$$\begin{aligned} \epsilon_{ii} &= \frac{1+\nu}{E}\sigma_{ii} - \frac{\nu}{E}(\sigma_{11} + \sigma_{22} + \sigma_{33}) \\ &= \frac{1}{2\mu}\sigma_{ii} - \frac{\nu}{E}(\sigma_{11} + \sigma_{22} + \sigma_{33}) \end{aligned} \tag{3.11}$$

for $i \in \{1, 2, 3\}$. Averaging Equation 3.11 yields

$$\begin{aligned} &\frac{1}{2\bar{\mu}}\sigma_{ii} - \frac{\bar{\nu}}{\bar{E}}(\sigma_{11} + \sigma_{22} + \sigma_{33}) \\ &\qquad = \left(\frac{1}{|V|} \int_V \frac{1}{2\mu(x)}\, dx \right) \sigma_{ii} - \left(\frac{1}{|V|} \int_V \frac{\nu(x)}{E(x)}\, dx \right) (\sigma_{11} + \sigma_{22} + \sigma_{33}), \end{aligned} \tag{3.12}$$

which can be reduced to calculate the homogenized ratio of ν and E using its arithmetic average as

$$X := \frac{\bar{\nu}(x)}{\bar{E}(x)} = \frac{1}{|V|} \int_V \frac{\nu(x)}{E(x)}\, dx. \tag{3.13}$$

LeVeque [LeV04, Section 22.1] gives the relationships between E as well as ν and Lamé parameters λ, μ as

$$E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu},\ \nu = \frac{1}{2}\left(\frac{\lambda}{\lambda + \mu}\right). \tag{3.14}$$

For the ratio of ν and E it follows that

$$\frac{\nu(x)}{E(x)} = \frac{\lambda(x)}{2\mu(x)\left(3\lambda(x) + 2\mu(x)\right)}. \tag{3.15}$$

It remains to insert Equation 3.15 into 3.13 and to solve for $\bar{\lambda}$ to obtain

$$\begin{aligned} && X &= \frac{\bar{\lambda}}{2\bar{\mu}\left(3\bar{\lambda} + 2\bar{\mu}\right)} \\ \Leftrightarrow && \bar{\lambda} &= 2\bar{\mu}X\left(3\bar{\lambda} + 2\bar{\mu}\right) \\ \Leftrightarrow && \bar{\lambda}\left(1 - 6\bar{\mu}X\right) &= 4\bar{\mu}^2 X \\ \Leftrightarrow && \bar{\lambda} &= \frac{4\bar{\mu}^2 X}{1 - 6\bar{\mu}X}. \end{aligned} \tag{3.16}$$

3.2.2 Alternative Calculation of 1. Lamé Parameter

The first Lamé parameter λ can also be derived in a simpler way. In this regard, Moczo et al. [MKV$^+$02, p. 12] point out the relationship between the bulk modulus K and Lamé parameters μ and λ,

$$K = \lambda + \frac{2}{3}\mu. \tag{3.17}$$

Harmonic averaging yields

$$X := \frac{1}{\bar{K}(x)} = \frac{1}{|V|} \int_V \frac{1}{\lambda(x) + \frac{2}{3}\mu(x)}\, dx \tag{3.18}$$

$$= \frac{1}{\bar{\lambda}(x) + \frac{2}{3}\bar{\mu}(x)}. \tag{3.19}$$

It remains to solve above equation for $\bar{\lambda}$ to obtain a simpler result compared to deriving $\bar{\lambda}$ from the averaged $\frac{\nu}{E}$:

$$\bar{\lambda} = \frac{1}{X} - \frac{2}{3}\bar{\mu}. \tag{3.20}$$

Running simulations on elastic media with rapidly varying λ shows how both approaches deliver virtually equivalent results.

3.2.3 Viscoelastic Media

Above calculations for elastic media can be readily extended to also cover the viscoelastic case featuring additional parameters Q_p and Q_s. As it has been explained in Subsection 2.1.4, these denote seismic quality factors of P and S waves.
In order not to exceed the scope of this thesis, we apply an arithmetic average for homogenization of these parameters,

$$\bar{Q} = \frac{1}{|V|} \int_V Q(x)\, dx. \tag{3.21}$$

As it will be seen in the evaluation of the corresponding results, this turns out to be an appropriate choice since the effect of homogenization on Q_p and Q_s is negligible compared to barycenter averaging.

3.2.4 Poroelastic Media

Poroelastic media represent another possible structure of soil that is relevant for seismic simulations. For homogenization of these materials, all of the respective material parameters presented in Subsection 2.1.4 need to be properly averaged.
Both the solid and the fluid density, ρ_S and ρ_F, can be homogenized using an arithmetic average as for the previously discussed materials. Equation 3.2 gives the correct homogenization formula for an arbitrary density distribution. Similarly, the Lamé parameters μ_M and λ_M can be averaged using Equations 3.8 and 3.16, respectively.
For the solid and fluid bulk moduli, K_S and K_F, LeVeque [LeV04, Section 9.14] uses a harmonic average to obtain the homogenized bulk modulus

$$\bar{K} = \left(\frac{1}{|V|} \int_V \frac{1}{K(x)}\, dx \right)^{-1}. \tag{3.22}$$

Moreover, Cheng [Che16, Section 1.2] gives the overall bulk modulus K of a porous medium by

$$K = \left(\frac{1}{K_S} + \frac{\Phi}{K_F} \right)^{-1} \tag{3.23}$$

where Φ denotes the matrix porosity. It is used to weight the fluid bulk modulus by the fraction of the total volume occupied by the fluid.
A poroelastic medium is also characterized by its matrix permeability κ. Multiple models exist to average permeability for different layouts of heterogeneous porous materials. Glover and Luo [GL20, Section 6.2] propose an arithmetic average in case the different material phases are arranged side by side in the direction of flow. Should the permeabilities be connected in series instead, a harmonic average is preferred. If no assumptions can be made about flow direction or composition of the porous material, a geometric average of the phases' permeabilities is proposed. All means are weighted by the volume fractions occupied by the different materials.
Regarding matrix porosity Φ, an arithmetic average seems to be a sound choice. Glover and Luo [GL20, Section 3] calculate mean porosity values for binary grain mixtures using

spherical approximations for the grains. They consider two cases where either many small grains fit between large ones or large grains displace smaller ones. The homogenization of both can be generalized best using an arithmetic average.
Zharbossyn et al. [ZBB+20, Section 2.1] also use an arithmetic average for the approximation of tortuosity T of a ternary particle mixture in a porous medium. In our application, T describes the ratio of the length of a connection via pores to the direct link between two points in the medium. The mean of two distances is calculated using its arithmetic average. Therefore, its application to the homogenization of matrix tortuosity is reasonable, too.
Grunberg and Nissan [GN49] provide a formula for the averaging of viscosity for a liquid mixture consisting of two components,

$$\bar{\nu} = x_1 \ln \nu_1 + x_2 \ln \nu_2 \tag{3.24}$$

where x_i denotes the molar mass fraction and ν_i the viscosity of the i-th component. This can be generalized to a continuous viscosity distribution which yields the homogenized fluid viscosity

$$\bar{\nu} = \frac{1}{|V|} \int_V \ln \nu(x)\, dx. \tag{3.25}$$

In summary, it has been shown that all three elastic material parameters can be homogenized by use of arithmetic and harmonic averages in Equations 3.2, 3.8 and 3.16. For $\bar{\rho}$ and $\bar{\mu}$ simple averaged expressions have been derived while for $\bar{\lambda}$ an additional step via $\frac{\nu}{E}$ was necessary. Arithmetic averaging of the viscoelastic parameters Q_p and Q_s has been proposed and an alternative means of calculating a homogenized $\bar{\lambda}$ was provided. For all poroelastic material parameters a homogenization approach has been given as well.

3.3 Introduction to Gaussian Quadrature

As indicated in the previous section, homogenized computation of material parameters requires the evaluation of multiple integrals per element. Material parameters are given as functions of up to three dimensions and, in general, no assumptions can be made about their distributions. Consequently, it will be necessary to compute the numerical values of their integrals using a suitable quadrature technique.
The basic idea of numerical integration is the approximation of a definite integral of an arbitrary function $f(x)$ using a weighted sum of values of f at certain points x_k called nodes. Stroud [Str74, Section 3.1] defines the general quadrature problem as

$$\int_a^b w(x) f(x)\, dx = \sum_{k=0}^{n} \lambda_k f(x_k) + E[f], \tag{3.26}$$

where

$$\begin{aligned} w(x) &: \text{weight function}, \\ \lambda_k &: \text{weights}, \\ x_k &: \text{nodes}, \\ E[f] &: \text{approximation error}. \end{aligned}$$

For computational purposes $E[f]$ will be ignored since it is unknown. Thus, Equation 3.26 becomes an approximation,

$$\int_a^b w(x)f(x)\,dx \approx \sum_{k=0}^{n} \lambda_k f(x_k). \tag{3.27}$$

Stroud assumes that the integral exists and that $f(x)$ is defined at x_k, $k \in \{0, \ldots, n\}$. This is required to choose the nodes and weights in any possible way that minimizes the approximation error. Several approaches exist for a suitable choice of $x_0, \ldots, x_n$ and $\lambda_0, \ldots, \lambda_n$ resulting in different degrees of Equation 3.27. A quadrature formula of degree d results in $E[f] = 0$ for all polynomials of degree $\leq$ d, and $E[f] \neq 0$ for higher polynomial degrees [Str74, Section 3.1].

For the purpose of this thesis we will focus on Gaussian quadrature. Stroud [Str74, Section 3.6] elaborates that a Gaussian quadrature formula has the same form as Equation 3.27 at degree $2n - 1$ which is the highest possible degree for n nodes. The nodes are chosen to be the roots of an orthogonal polynomial $P_n(x)$. For the choice of P_n again several approaches exist, among which the Jacobi polynomials $P_n^{\alpha,\beta}(x)$ are used in the implementation. The resulting quadrature formula for $[a, b] = [-1, 1]$ and $w(x) = (1-x)^\alpha(1+x)^\beta$ where $\alpha, \beta > -1$ is called Gauss-Jacobi quadrature. It has the form

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta f(x)\,dx \approx \sum_{k=0}^{n} \lambda_k f(x_k), \tag{3.28}$$

where $x_0, \ldots, x_n$ are the roots of the n-th Jacobi polynomial. For $\lambda_0, \ldots, \lambda_n$ multiple formulations exist. See for example Brzezinski [Brz18, p. 3] or Hale and Townsend [HT13, p. 2] for a formula which slightly differs from the one used in SeisSol. The simulator uses

$$\lambda_k = -\frac{2n+\alpha+\beta+2}{n+\alpha+\beta+1} \frac{(n+\alpha)!(n+\beta)!}{(n+\alpha+\beta)!(n+1)!} \frac{2^{\alpha+\beta}}{P_n^{\alpha,\beta\prime}(x_k) P_{n+1}^{\alpha,\beta}(x_k)}. \tag{3.29}$$

In order to use Gauss-Jacobi quadrature for the homogenization of material parameters in SeisSol, it has to be applied to a three-dimensional integral over a tetrahedral mesh element.

Stroud [Str71, Section 2.4] proposes a corresponding approximation for an integral of a function $f(x, y, z)$ over a tetrahedral volume. In our case f represents a material parameter distribution. We substitute the integrand by a monomial which can be integrated exactly,

$$I := \int_0^1 \int_0^{1-x} \int_0^{1-x-y} f(x,y,z)\,dzdydx = \int_0^1 \int_0^{1-x} \int_0^{1-x-y} x^\alpha y^\beta z^\gamma\,dzdydx. \tag{3.30}$$

Stroud applies the transformation

$$\begin{aligned} x &= u \\ y &= v(1-u) \\ z &= w(1-v)(1-u), \end{aligned} \tag{3.31}$$

which has the Jacobian determinant of

$$J = (1-u)^2(1-v). \tag{3.32}$$

Let us apply above equations to Equation 3.30 in order to obtain

$$I = \int_0^1 \int_0^1 \int_0^1 (1-u)^{\beta+\gamma+2}(1-v)^{\gamma+1} u^\alpha v^\beta w^\gamma \, dudvdw. \tag{3.33}$$

Above equation can be divided into three separate integrals, namely

$$I_u := \int_0^1 (1-u)^2 (1-u)^{\beta+\gamma} u^\alpha \, du = \int_0^1 (1-u)^2 f(u) \, du, \tag{3.34}$$

$$I_v := \int_0^1 (1-v)(1-v)^\gamma v^\beta \, dv = \int_0^1 (1-v) g(v) \, dv, \tag{3.35}$$

$$I_w := \int_0^1 w^\gamma \, dw = \int_0^1 h(w) \, dw. \tag{3.36}$$

Now Gauss-Jacobi quadrature can be applied using $\beta = 0$ for all equations as well as $\alpha = 2$ for Equation 3.34, $\alpha = 1$ for 3.35 and $\alpha = 0$ for 3.36. This results in quadrature points x_u, x_v, x_w as well as weights λ_u, λ_v, λ_w for all three dimensions.
Stroud composes a conical product out of the resulting quadrature points and weights to obtain the final points $\boldsymbol{x}$ and weights λ as

$$\boldsymbol{x}_{i,j,k} = \begin{bmatrix} x_{u_{i,j,k}} \\ x_{v_{i,j,k}}(1 - x_{u_{i,j,k}}) \\ x_{w_{i,j,k}}(1 - x_{u_{i,j,k}})(1 - x_{v_{i,j,k}}) \end{bmatrix}, \tag{3.37}$$

$$\lambda_{i,j,k} = \lambda_{u_{i,j,k}} \lambda_{v_{i,j,k}} \lambda_{w_{i,j,k}}. \tag{3.38}$$

Using Equations 3.37 and 3.38, Equation 3.30 can be evaluated numerically for a material parameter function f over a tetrahedral mesh element.
Listing 3.2 shows the implementation of Stroud's approach in SeisSol, which will be used in Section 3.4.

```cpp
inline void TetrahedronQuadrature(double (*points)[3], double* weights, unsigned int n) {
  /*...*/

  auto points0 = std::vector<double>(n);
  auto points1 = std::vector<double>(n);
  auto points2 = std::vector<double>(n);

  auto weights0 = std::vector<double>(n);
  auto weights1 = std::vector<double>(n);
  auto weights2 = std::vector<double>(n);

  // Get the Gauss-Jacobi positions and weights.
  GaussJacobi(points0.data(), weights0.data(), n, 2, 0);
  GaussJacobi(points1.data(), weights1.data(), n, 1, 0);
  GaussJacobi(points2.data(), weights2.data(), n, 0, 0);

  // Shift and rescale positions because Stroud
  // integrates over [0,1] and gaujac of num. recipes
  // considers [-1,1].
20    for (size_t i = 0; i < n; ++i) {
21      points0[i] = 0.5 * points0[i] + 0.5;
22      points1[i] = 0.5 * points1[i] + 0.5;
23      points2[i] = 0.5 * points2[i] + 0.5;
24
25      weights0[i] = 0.5 * 0.5 * 0.5 * weights0[i];
26      weights1[i] = 0.5 * 0.5 * weights1[i];
27      weights2[i] = 0.5 * weights2[i];
28    }
29
30    for (size_t i = 0; i < n; ++i) {
31      for (size_t j = 0; j < n; ++j) {
32        for (size_t k = 0; k < n; ++k) {
33          const auto curIndex = i * n * n + j * n + k;
34          points[curIndex][0] = points0[i];
35          points[curIndex][1] = points1[j] * (1 - points0[i]);
36          points[curIndex][2] = points2[k] * (1 - points1[j]) *
37            (1 - points0[i]);
38          weights[curIndex] = weights0[i] * weights1[j] * weights2[k];
39        }
40      }
41    }
42    /*...*/
43  }
```

Listing 3.2: Computation of quadrature points and weights for tetrahedral volumes. Source: SeisSol, Quadrature.h, ll. 156-211

In lines 13 - 15 Gauss-Jacobi quadrature is applied to Equations 3.34 - 3.36 with the corresponding choices for α and β. As previously stated, Gauss-Jacobi quadrature approximates integrals over $[a, b] = [-1, 1]$. Consequently, the resulting quadrature points and weights need to be rescaled since Stroud's approach assumes $[a, b] = [0, 1]$. The conical product is constructed in lines 30 - 41, which yields the desired points and weights for quadrature over a tetrahedral volume.

3.4 Implementation

In the previous sections, a new homogenization approach for elastic and viscoelastic material parameters as well as mathematical prerequisites regarding numerical quadrature have been defined. In the following, their actual implementation in SeisSol will be detailed.

All homogenization functionality is realized inside a new ElementAverageGenerator class. Just like the basic ElementBarycentreGenerator it inherits from QueryGenerator and thus provides methods for the generation and evaluation of an easi query. Additionally, it needs members for storing and handling quadrature points and weights as well as element volumes. The desired quadrature degree is controlled via two macros and was chosen to be 4 in all simulations generated for this thesis, which results in $4^3 = 64$ quadrature points per element.

```
#define QUAD_DEG 4
#define NUM_QUADPOINTS (QUAD_DEG * QUAD_DEG * QUAD_DEG)
```

Listing 3.3: Quadrature degree macros.

```
class seissol::initializers::ElementAverageGenerator : public seissol::initializers::QueryGenerator {
public:
  explicit ElementAverageGenerator(MeshReader const& meshReader);
  virtual easi::Query generate() const;
  std::vector<double> elementVolumes();
  std::vector<double> getElemVolumes() const { return m_elemVolumes; };
  std::array<double, NUM_QUADPOINTS> getQuadratureWeights() const { return m_quadratureWeights; };
private:
  MeshReader const& m_meshReader;
  std::vector<double> m_elemVolumes;
  std::array<double, NUM_QUADPOINTS> m_quadratureWeights;
  std::array<std::array<double,3>, NUM_QUADPOINTS> m_quadraturePoints;
};
```

Listing 3.4: ElementAverageGenerator class declaration.

In order to calculate homogenized parameter values in a three-dimensional space, it is generally required to divide by the corresponding volume which specifies the integral bounds. In this application, parameters are integrated over tetrahedral elements with vertices as defined in Section 3.1. Their volume is known to be calculated as

$$V = \frac{1}{6}|(\boldsymbol{v_1} - \boldsymbol{v_0}) \cdot ((\boldsymbol{v_2} - \boldsymbol{v_0}) \times (\boldsymbol{v_3} - \boldsymbol{v_0}))|. \tag{3.39}$$

Element volumes are computed step by step using Equation 3.39 during initialization of the ElementAverageGenerator class as shown in Listing 3.5.

```
std::vector<double> seissol::initializers::ElementAverageGenerator::elementVolumes() {
    /* Local declarations */
    // Compute a := (v_1-v_0), b := (v_2-v_0) and c := (v_3-v_0)
    for (unsigned elem = 0; elem < elements.size(); ++elem) {
        for (int i = 0; i < 3; ++i) {
          a[i] = vertices[ elements[elem].vertices[1] ].coords[i] - vertices[elements[elem].vertices[0] ].coords[i];
          b[i] = vertices[ elements[elem].vertices[2] ].coords[i] - vertices[elements[elem].vertices[0] ].coords[i];
          c[i] = vertices[ elements[elem].vertices[3] ].coords[i] - vertices[elements[elem].vertices[0] ].coords[i];
        }
        // Cross product
        bxc[0] = b[1] * c[2] - b[2] * c[1];
        bxc[1] = b[2] * c[0] - b[0] * c[2];
        bxc[2] = b[0] * c[1] - b[1] * c[0];
        // Dot product
        for (int i = 0; i < 3; ++i) {
          elemVolumes[elem] += a[i] * bxc[i];
        }
        elemVolumes[elem] = abs(elemVolumes[elem]) / 6;
        return elemVolumes;
    }
}
```

Listing 3.5: Computation of tetrahedral element volumes.

Apart from element volumes, quadrature points and weights are computed during construction, too. Listing 3.6 shows how this relies on the TetrahedronQuadrature method already present in SeisSol. It implements the computation of quadrature points and weights for a reference tetrahedron of edge length 1 according to the approach presented in Section 3.3.

```
seissol::initializers::ElementAverageGenerator::ElementAverageGenerator(MeshReader const& meshReader)
  : m_meshReader(meshReader)
  {
    // Generate subpoints and weights in reference tetrahedron using Gaussian quadrature
    double quadraturePoints[NUM_QUADPOINTS][3];
    double quadratureWeights[NUM_QUADPOINTS];
    seissol::quadrature::TetrahedronQuadrature(quadraturePoints, quadratureWeights, QUAD_DEG);
    /* Initialize const class members with results */
    // Initialize element volumes
    m_elemVolumes = elementVolumes();
  }
```

Listing 3.6: ElementAverageGenerator constructor.

Generation of an easi query in ElementAverageGenerator involves an additional step compared to the barycentered implementation.

For each element, NUM_QUADPOINTS points instead of just one have to be sampled. The TetrahedronQuadrature method used in Listing 3.6 returns quadrature points inside a reference tetrahedron of edge length 1 as illustrated in Figure 3.1. These have to be transformed to global coordinates for all tetrahedra using the corresponding transformation method already present in SeisSol as seen in Listing 3.7, line 12. The results are stored and grouped in a matrix inside the easi query.

```
1  easi::Query seissol::initializers::ElementAverageGenerator::generate() const {
2    std::vector<Element> const& elements = m_meshReader.getElements();
3    std::vector<Vertex> const& vertices = m_meshReader.getVertices();
4
5    // Generate query using subpoints for each element
6    easi::Query query(elements.size() * NUM_QUADPOINTS, 3);
7
8    // Transform subpoints to global coordinates for all elements
9    for (unsigned elem = 0; elem < elements.size(); ++elem) {
10     for (unsigned i = 0; i < NUM_QUADPOINTS; ++i) {
11       std::array<double, 3> xyz{};
12       seissol::transformations::tetrahedronReferenceToGlobal(/*Vertices*/, m_quadraturePoints[i].data(), xyz.data());
13       for (unsigned dim = 0; dim < 3; ++dim) {
14         query.x(elem * NUM_QUADPOINTS + i,dim) = xyz[dim];
15       }
16       // Group
17       query.group(elem * NUM_QUADPOINTS + i) = elements[elem].group;
18     }
19   }
20
21   return query;
22 }
```

Listing 3.7: Generation of an easi query for homogenization.

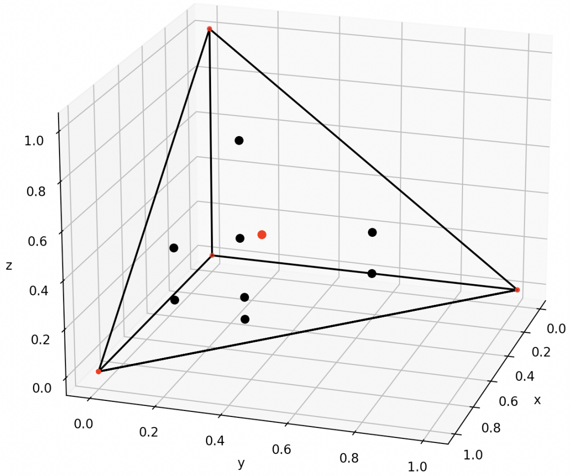


Figure 3.1: Barycenter (red) and subpoints (black) for quadrature degree 2 inside a reference tetrahedron of edge length 1.

The added homogenization functionality seamlessly integrates with the existing easi query evaluation code. The given query generator is dynamically downcast to its actual implementation at runtime. In case an ElementAverageGenerator is supplied, homogenization is applied. Otherwise, the parameter values are stored directly in the material parameter database.

```
1 if (const ElementAverageGenerator* gen = dynamic_cast<const
      ElementAverageGenerator*>(&queryGen)) {
2     /* Homogenization */
3 } else {
4     // Usual behavior without homogenization
5     for (unsigned i = 0; i < numPoints; ++i) {
6       m_materials->at(i) = seissol::model::ElasticMaterial(elasticMaterials[i
            ]);
7     }
8 }
```

Listing 3.8: Dynamic cast to choose between homogenized and normal implementations.

Listing 3.9 shows the homogenization branch represented by line 2 in Listing 3.8. std::vector<ElasticMaterial> materialsMean contains the homogenized material parameter values. Quadrature weights need to be upscaled by a factor of $\frac{element\,volume}{reference\,volume}$ with a reference tetrahedron volume of $\frac{1}{6}$. Gaussian quadrature is applied for each parameter by implementing Equation 3.27 using the points and weights obtained from the TetrahedronQuadrature method. Lines 8 and 9 show quadrature of Equations 3.2 and 3.8 respectively. Line 11 implements quadrature of Equation 3.15 inserted into 3.13.

```
1 /* Local declarations */
2
3 // Approximate element volume integrals using Gaussian quadrature
4 for (unsigned i = 0; i < numPoints; ++i) {
5   // Scale up quadrature weights by (element volume / reference volume)
6   double quadWeight = 6 * elemVolumes[i / NUM_QUADPOINTS] * quadratureWeights[
        i % NUM_QUADPOINTS];
7   // Integrate rho and (1 / mu)
```

```
  materialsMean[i / NUM_QUADPOINTS].rho += elasticMaterials[i].rho * quadWeight;
  materialsMean[i / NUM_QUADPOINTS].mu += 1 / elasticMaterials[i].mu * quadWeight;
  // Integrate (nu / E) to obtain lambda
  vERatioMean[i / NUM_QUADPOINTS] += elasticMaterials[i].lambda / (2 * elasticMaterials[i].mu * (3 * elasticMaterials[i].lambda + 2 * elasticMaterials[i].mu)) * quadWeight;
}
```

Listing 3.9: Computation of homogenized material parameters - 1.

The resulting integrals are divided by the respective element volumes and in case of harmonic averages, the reciprocal is taken. It remains to compute the homogenized $\bar{\lambda}$ using Equation 3.16 and to store the results in the material parameter database as shown in Listing 3.10.

```
// Divide by volumes to obtain parameter mean values and store them in m_materials
for (unsigned i = 0; i < numElems; ++i) {
  materialsMean[i].rho /= elemVolumes[i];
  materialsMean[i].mu /= elemVolumes[i];
  vERatioMean[i] /= elemVolumes[i];
  // Harmonic average is used for mu, so take the reciprocal
  materialsMean[i].mu = 1 / materialsMean[i].mu;
  // Derive lambda from averaged mu and (nu / E)
  materialsMean[i].lambda = (4 * pow(materialsMean[i].mu, 2) * vERatioMean[i]) / (1 - 6 * materialsMean[i].mu * vERatioMean[i]);

  m_materials->at(i) = seissol::model::ElasticMaterial(materialsMean[i]);
}
```

Listing 3.10: Computation of homogenized material parameters - 2.

This chapter has reviewed all key aspects of a numerical homogenization approach for material parameters in the context of seismic simulations. In the first section, the state of the art has been introduced as a simple but insufficiently accurate means of approximating complex parameter distributions. The mathematical foundations for homogenization have been presented in Section 3.2. In the following section, Gaussian quadrature has been defined and applied in an approach by Stroud to compute the numerical value of integrals over tetrahedral volumes. Based on these prerequisites, the final section has been dedicated to the implementation of the homogenization approach in SeisSol.

In the chapter that follows, the simulation results generated using homogenized material parameters will be presented and evaluated compared to results obtained using the existing implementation.

4 Evaluation of Results

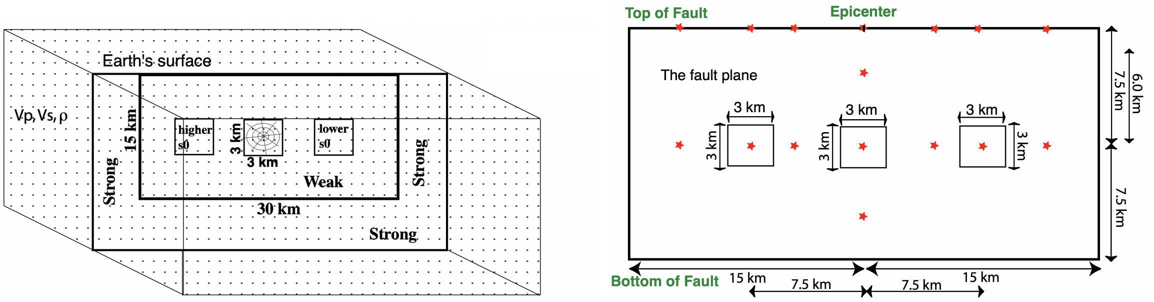
In the following pages, the numerical homogenization approach for material parameters will be validated and benchmarked on a variety of seismic examples. All simulations are based on either the Southern California Earthquake Center (SCEC) TPV5 scenario [Sou05a] or the SISMOWINE WP2_LOH1 [Sei06] point source example. The following section gives a brief introduction to both cases and their usage for the purpose of this thesis.

4.1 Seismic Scenarios

With regard to the seismic scenarios employed during testing, it should be noted that some simulations rely on the predefined meshes and material files while others use strongly altered versions of them. This serves to highlight certain cases where the base implementation is particularly inaccurate or subject to systematic errors. Only the default scenarios will be detailed upfront. Where changes to the underlying meshes and / or material configurations have been made, these will be referred to in the respective sections.

4.1.1 SCEC TPV5

TPV5 features a "vertical right-lateral strike-slip planar fault that resides in a halfspace" [Sou05a]. The problem specification moves on to explain that rupture is allowed on the fault plane whose lateral and bottom edges are represented by strength barriers. The nucleation point is centered on the fault and framed by a small square nucleation patch. The nucleation patch has an initial shear stress higher than the initial static yield stress which leads to nucleation and onset of failure across the fault plane [Sou05a].
Two square patches are located between the nucleation patch and both of the fault's lateral boundaries. They feature higher and lower initial shear stress conditions, respectively, than the remaining part of the fault [Sou05a]. The exact geometry is illustrated in Figure 4.1.



(a) 3D overview of TPV5 geometry featuring the planar fault.

(b) TPV5 fault geometry including patches and fault receiver locations.

Figure 4.1: Views of SCEC TPV5 overall and fault geometries. Source: [Sou05b]

4.1.2 WP2_LOH1

The WP2_LOH1 scenario is divided into a homogeneous half-space and a thin layer with different material properties on top. The earthquake originates at a point source inside the half-space, slightly below the top layer [Sei06].

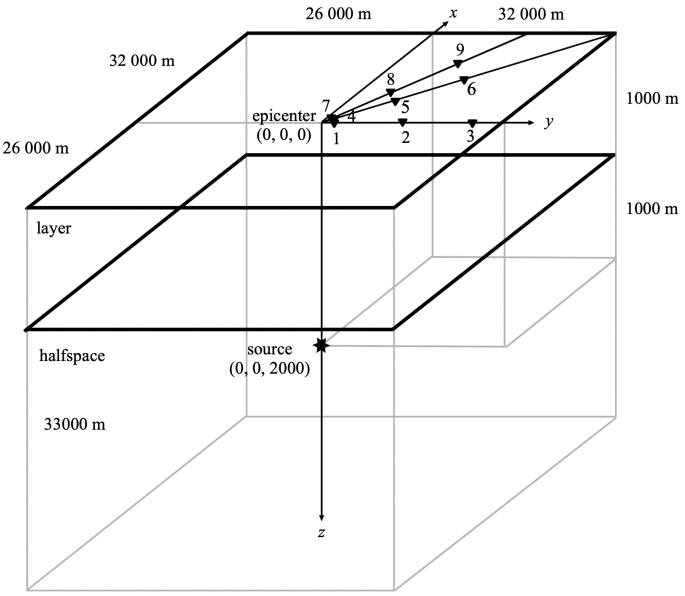


Figure 4.2: 3D overview of WP2_LOH1 geometry. Source: [Sei06]

Turning now to the evaluation of simulations conducted using above seismic scenarios.

4.2 Layered Density Distributions

The section below aims to compare the performance of homogenization against the current barycenter approach using multiple discontinuous, layered density distributions. In the first subsection, the material interfaces are widely spaced. The second subsection tries to eliminate the effects of reflections at material boundaries by using a very fine separation instead.

4.2.1 Coarse Layer Separation

In the following, a single setup is used to run a set of four simulations in both a TPV5 and a WP2_LOH1 scenario.
Figure 4.3 shows the layered density distribution developed for this purpose. The corresponding easi file keeps the Lamé parameters constant and varies density in increments of $400\,\frac{kg}{m^3}$ every $2\,km$ of depth. It starts at $1200\,\frac{kg}{m^3}$ at the earth's surface and increases up to $4000\,\frac{kg}{m^3}$ at $16\,km$. There the stepped progression starts again at $1200\,\frac{kg}{m^3}$ and stops at a depth of $30\,km$. Below, density remains constant.

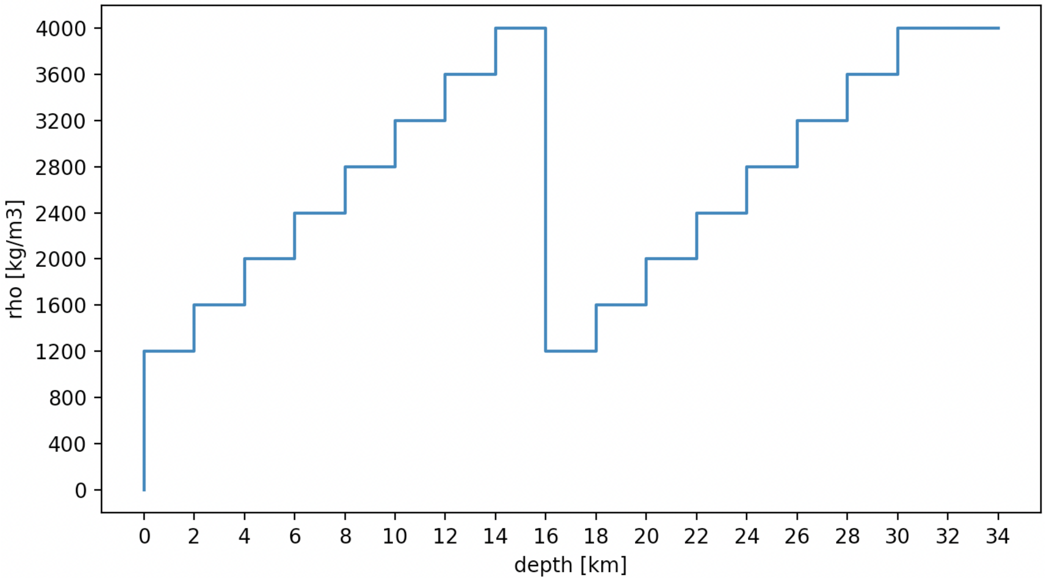


Figure 4.3: Density distribution over depth for the layered density profiles of TVP5 and WP2_LOH1.

The objective is to compare the performance of homogenization on an unstructured, comparatively low-resolution mesh against the base implementation running on both the same and other preferential mesh configurations. These include the unstructured mesh at a higher resolution as well as a mesh with planar interfaces aligned with the material boundaries. The latter ensures that no elements are intersecting any material boundaries. This yields an exact result at any mesh resolution.

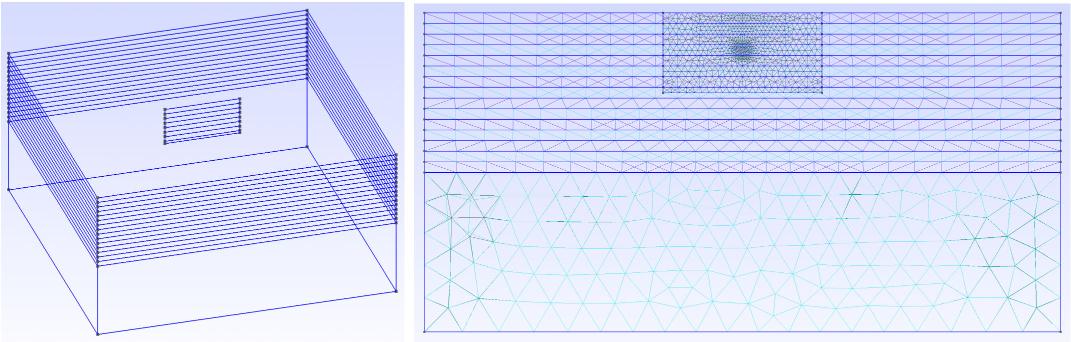
SCEC TPV5

Table 4.1 details the element resolutions in different areas of the three mesh types for TPV5. "Coarse" denotes the unstructured, low-resolution mesh. "Fine" refers to the same mesh at a higher resolution and "Layered" describes the mesh with material aligned layers. The columns contain the edge length of tetrahedra in the unstructured volume, on the fault plane and on the nucleation patch. The overall resolution increases from coarse over layered to fine.

	Normal Length [m]	Fault Length [m]	Patch Length [m]
Coarse	$25 \cdot 10^4$	600	300
Layered		200	
Fine	$25 \cdot 10^3$		

Table 4.1: Element edge lengths for three different configurations of TPV5.

Figure 4.4 gives an impression of the layered TPV5 scenario in the Gmsh finite element mesh generator [GR09]. The interface boundaries on the outside surfaces and on the centered fault are visible. In the meshed 2D view, one can see how the elements' edges are aligned with the interfaces and do not intersect them, also on the fault. It shows the higher resolution nucleation patch at the center of the fault as well. Nucleation occurs close to the earth's surface and we are interested in the solution at the surface. Therefore, the lower halfspace of the TPV5 environment is less relevant for the comparison of wave propagation and thus not layered. The easi configuration assumes constant material parameters in this area.



(a) 3D view of layered TPV5 scenario in Gmsh.

(b) Cross-section at the fault of layered TPV5 scenario with mesh output in Gmsh.

Figure 4.4: 3D view and cross-section of layered TPV5 scenario in Gmsh.

With respect to the errors induced by the different mesh configurations, the base implementation running on a layered mesh can be considered the exact result. All material parameters are constant between neighboring interfaces. Since no element intersects an interface, no averaging needs to be performed and the material parameters inside every cell are correct.

For the remaining setups it holds that errors generally only occur for all elements intersecting a material boundary. In these cases, some kind of averaging is required.

The largest error is to be expected from the base implementation on a coarse mesh. Since an element's barycenter is located in either of both layers, either of the corresponding values for density will be sampled. Only the homogenized implementation is able to consider the proportions of an element below and above an interface. This yields a proper average of an accuracy proportional to the quadrature degree selected.

The base implementation running on a fine mesh should deliver more accurate results than on the coarse mesh because the overall element edge length is lower. As a result, more yet smaller elements have intersections with material boundaries. This should result in an inaccurate density being sampled for a lower fraction of the entire volume.

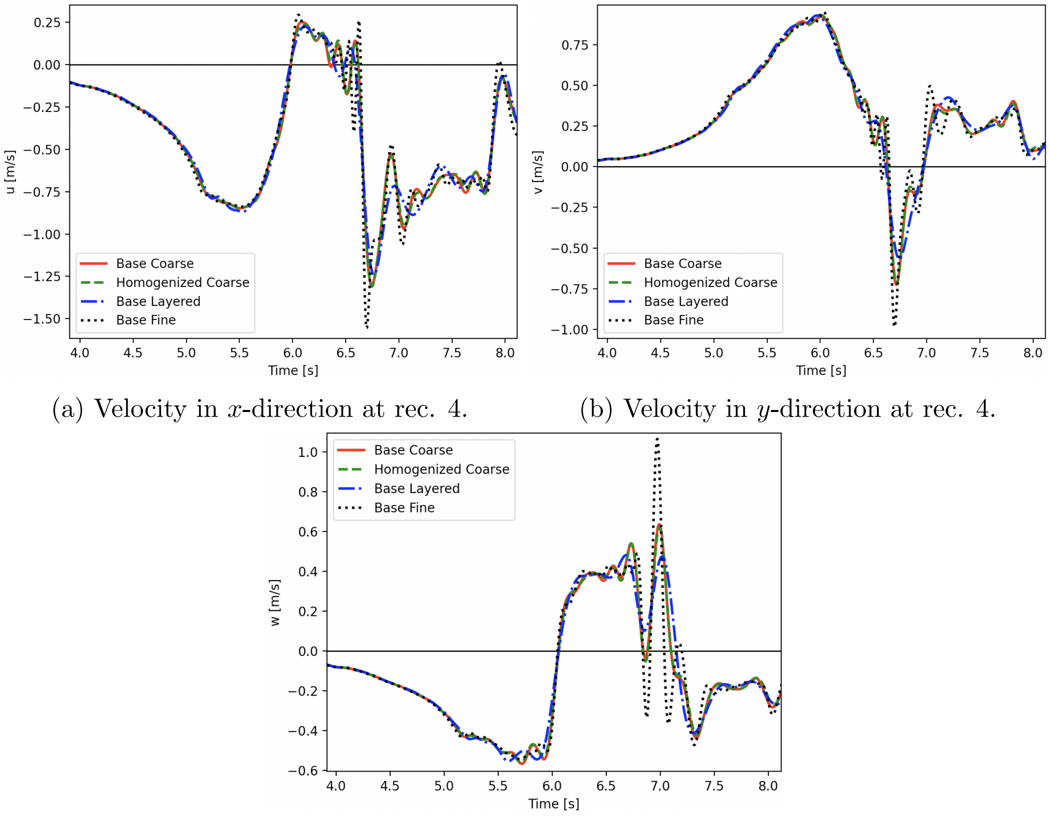
Figure 4.5 contains the particle velocity output at receiver 4 for all simulation setups detailed above. Figure 4.6 gives a close-up view for one second at the same receiver. In general, one can see that all four curves do not deviate significantly from each other for the majority of time. Nonetheless, especially in some areas with strong fluctuations the divergence is larger.

It is striking that the solid and dashed curves for the base and homogenized implementations on a coarse mesh almost perfectly line up at all times. At 2000 m per layer, the interfaces are far apart. As a result, even on the coarse mesh the number and size of elements intersecting a material boundary is relatively low. The associated approximation error for density of these elements has virtually no visible impact on wave propagation.

However, both curves still deviate from the exact solution in some areas. This suggests that there might be differences in the reflection and transmission of waves at material boundaries between both meshes. The different arrangement of elements might have a greater impact on the reflection characteristics than the averaging of density itself.

A higher mesh resolution, however, does not seem to be helpful either. Especially between 6.6 s and 7.0 s in Figure 4.6, the dotted curve from the base implementation on a fine mesh deviates even more from the layered mesh result. Again, this hints at the relevance of reflection and transmission behavior of waves for the accuracy of results.

In summary, homogenization of density does not deliver better results than the barycenter implementation in this layered material setup of TPV5. Even without homogenization, the results are already very accurate most of the time. A potential improvement does not seem to be achievable via better averaging of density. Instead, the reflection characteristic of waves at material boundaries is likely to have a greater impact. At the same time, a higher resolution mesh leads to the curves diverging further away from the desired result. All other receivers support the above findings.

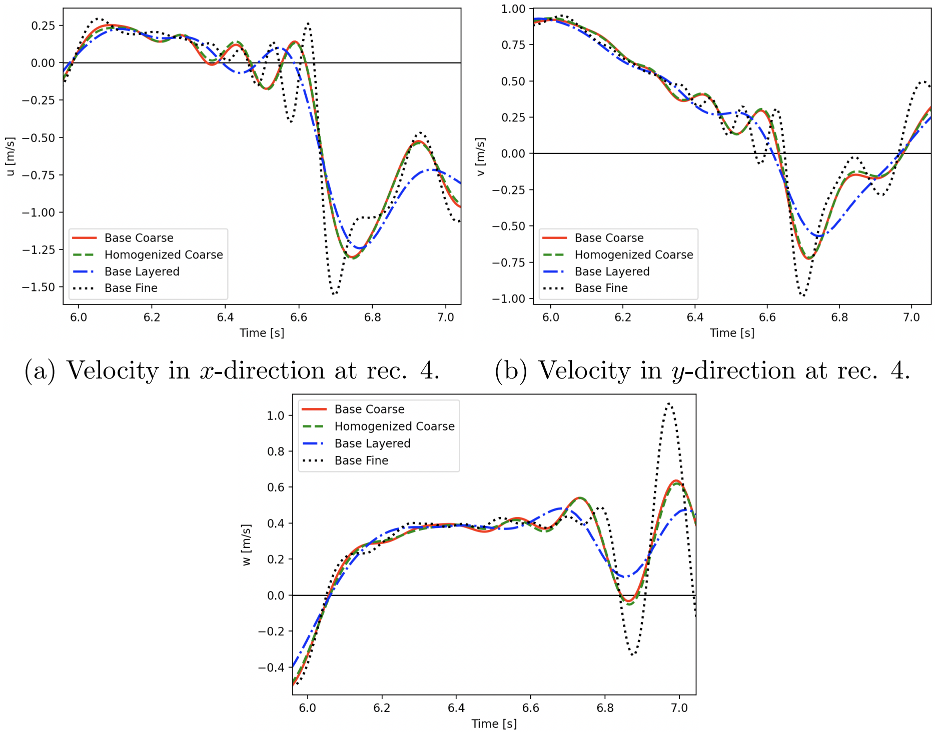


(a) Velocity in x-direction at rec. 4.

(b) Velocity in y-direction at rec. 4.

(c) Velocity in z-direction at rec. 4.

Figure 4.5: TPV5 velocity output at receiver 4 for layered density on 2 km thick layers.



(a) Velocity in x-direction at rec. 4.

(b) Velocity in y-direction at rec. 4.

(c) Velocity in z-direction at rec. 4.

Figure 4.6: TPV5 detailed velocity output at receiver 4 for layered density on 2 km thick layers.

WP2_LOH1

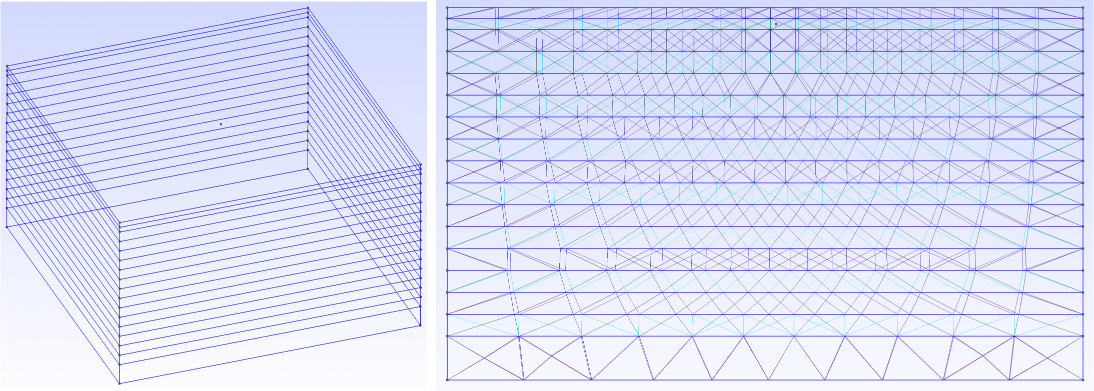
As indicated previously, the same experiment is repeated in a WP2_LOH1 scenario. It adopts the density distribution from Figure 4.3 without change. Moreover, the same types of mesh configurations but with different element sizes are available.

Table 4.2 contains the element resolutions of the meshes used for WP2_LOH1. Here, the second and third columns contain the resolutions inside and outside the refined box of LOH1. "Coarse" and "Layered" share the same element edge lengths because this results in the minimum possible number of elements. "Fine" uses the preconfigured settings.

	Normal Length [m]	Box VIn [m]	Box VOut [m]
Coarse	50000	2000	30000
Layered			
Fine	5000	200	3000

Table 4.2: Element edge lengths for three different configurations of WP2_LOH1.

In Figure 4.7 one can see the layered WP2_LOH1 mesh in Gmsh. It has only about half the depth of TPV5. As a result, the interfaces divide almost the entire volume into sections of 2 km thickness. Characteristic of WP2_LOH1 is its additional 1 km thick layer right below the earth's surface. Again, the elements' edges in the 2D view are nicely aligned with the interfaces. The 3D view also shows the point source at its center and close to the surface.

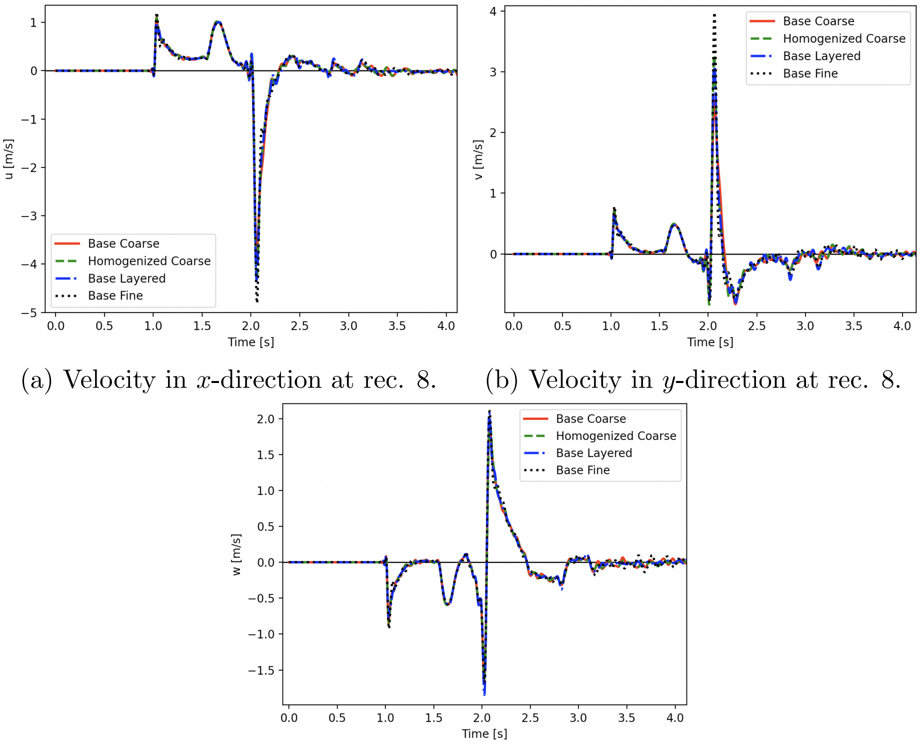


(a) 3D view of layered WP2_LOH1 scenario in Gmsh.

(b) Cross-section of layered WP2_LOH1 scenario with mesh output in Gmsh.

Figure 4.7: 3D view and cross-section of layered WP2_LOH1 scenario in Gmsh.

Similar results compared to the TPV5 case can be expected. The base implementation running on a layered mesh will produce an exact density approximation for having no elements intersecting any material boundaries. The TPV5 run has shown no major differences in outputs between the other setups, which is likely to be similar for WP2_LOH1.



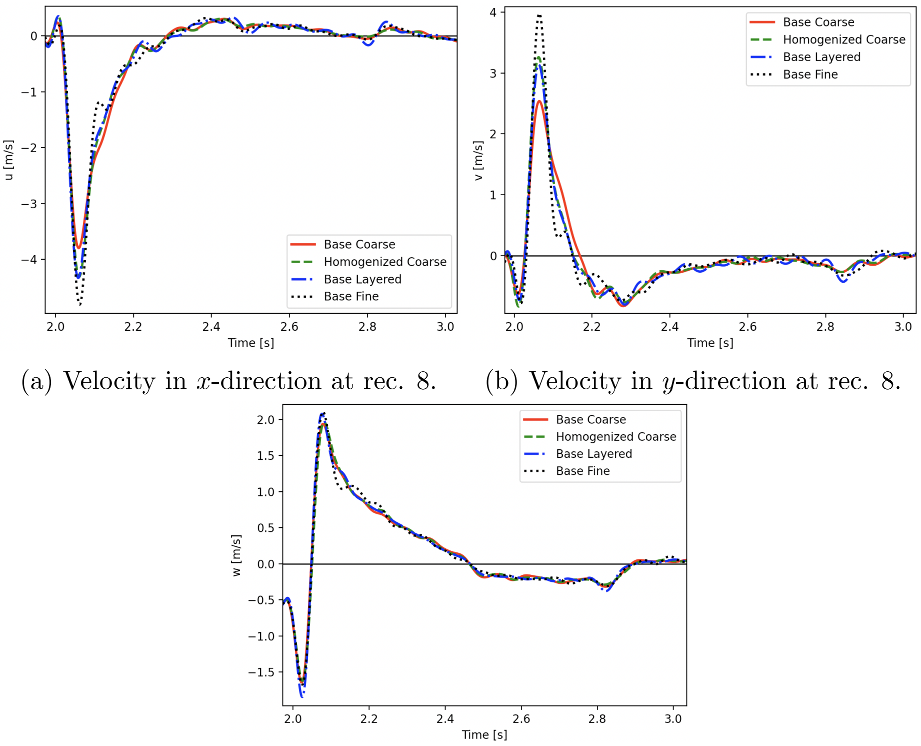
(a) Velocity in x-direction at rec. 8.

(b) Velocity in y-direction at rec. 8.

(c) Velocity in z-direction at rec. 8.

Figure 4.8: WP2_LOH1 velocity output at receiver 8 for layered density on 2 km thick layers.

The curves in Figure 4.8 for the particle velocity at receiver 8 are largely congruent. Figure 4.9 gives a more detailed view of one second at this receiver. Just like with TPV5, the strongest divergence between all solutions is visible when their curves are spiking. Interestingly, the homogenized implementation generates a curve that is more offset from the base implementation on a coarse mesh for WP2_LOH1. It is closest of all to the layered mesh result at and around the peak in velocity at 2.05 s in x- and y-directions. For the remainder of time, all solutions are again too similar to make out any one that best approaches the ideal result. However, analogously to TPV5 the dotted curve for the base implementation on a fine mesh seems to be the most inaccurate for the majority of time.
Due to the overall higher element size of WP2_LOH1, it might be more relevant to properly average density for elements intersecting material boundaries. The barycenter implementation is not able to produce accurate velocity outputs at both low and high mesh resolutions. At a higher resolution, differences in reflection and transmission of the wave at material boundaries are noticeable. This leads to more frequent over- and undershooting of the exact solution. Notice for example the smaller peaks on the dotted curves at 2.15 s which are not present on the other outputs.



(a) Velocity in x-direction at rec. 8.

(b) Velocity in y-direction at rec. 8.

(c) Velocity in z-direction at rec. 8.

Figure 4.9: WP2_LOH1 detailed velocity output at receiver 8 for layered density on 2 km thick layers.

In this subsection, multiple simulations featuring different mesh setups and a layered density material have been run in two different scenarios. It has been shown that the possible improvements using homogenization of material parameters are rather limited for

wide material layers. The second simulation using WP2_LOH1 has suggested a weak link may exist between mesh element size and a more accurate solution using homogenization compared to the barycenter approach.
The following sections will move on to investigate some more specifically constructed cases to highlight the areas in which homogenization can bring more significant improvements.

4.2.2 Fine Layer Separation

As was pointed out in Subsection 2.1.2, only part of a seismic wave is transmitted while the remainder is reflected at a material boundary. Reflection only occurs if the distance between material layers is higher than the wavelength.
Therefore, another experiment is conducted with material layers of alternating density separated considerably tighter than the average element size. On a large scale, the material can be considered nearly homogeneous. The objective is to compare both implementations' capability to resemble the nearly homogeneous density profile at a low level of wave reflections.
Three simulations are run on a standard WP2_LOH1 scenario at a fixed mesh resolution. It uses an elastic material configuration with density alternating between $\rho_{min} = 2000\,\frac{kg}{m^3}$ and $\rho_{max} = 4000\,\frac{kg}{m^3}$ every $50\,m$ in z-direction. Listing 4.1 shows a suitable easi setup with fixed Lamé parameters and a floored sin-function used to periodically alternate density with depth.

```
!FunctionMap
map:
  rho:     return 2000 + floor(sin(0.062831853071796 * z) + 1) * 2000;
  mu:      return 30000000000;
  lambda:  return 20000000000;
```

Listing 4.1: easi configuration for alternating density on $50\,m$ thick layers.

As a reference, a third simulation uses the base implementation and the constant average density of $3000\,\frac{kg}{m^3}$.
Let us recall from Subsection 2.1.2 that the velocities of P and S waves are given by

$$v_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}, v_s = \sqrt{\frac{\mu}{\rho}}. \tag{4.1}$$

v_s is always smaller than v_p. Therefore, v_s needs to be minimized by inserting the maximum density that occurs which yields

$$v_{s,min} = \sqrt{\frac{\mu}{\rho_{max}}} = \sqrt{\frac{3.0 \cdot 10^{10}\,\frac{N}{m^2}}{4000\,\frac{kg}{m^3}}} = 2738.61\frac{m}{s}. \tag{4.2}$$

Empirical analysis of the spectra of the three simulations in Figure 4.10 shows that frequency components above $50\,Hz$ can be considered negligible.

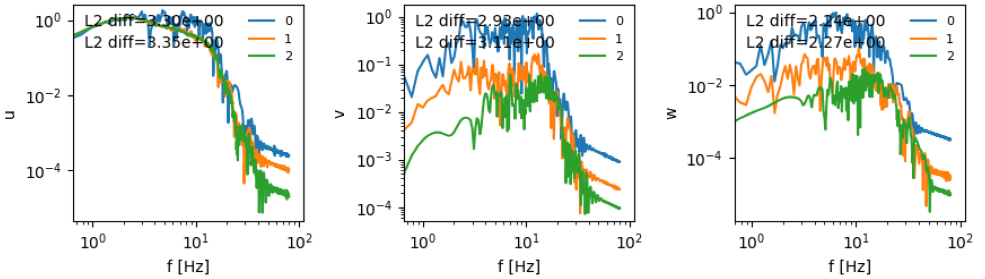
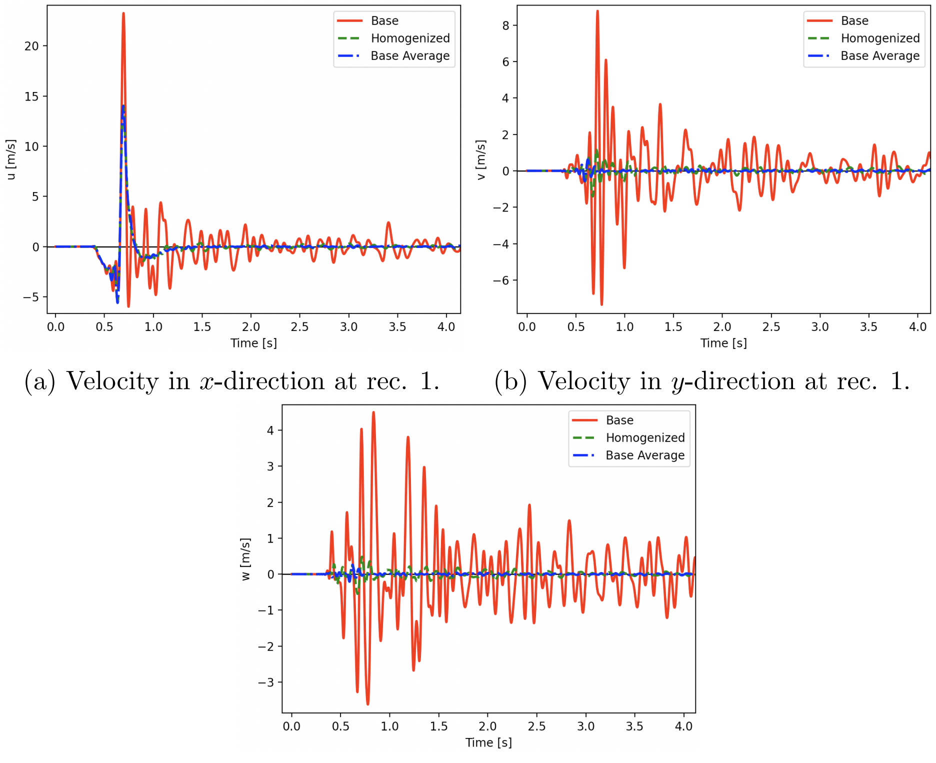


Figure 4.10: WP2_LOH1 frequency spectra for alternating density on 50 m thick layers.

The resulting minimum wavelength is given by

$$\lambda_{s,min} = \frac{v_{s,min}}{f_{max}} = \frac{2738.61\frac{m}{s}}{50\frac{1}{s}} = 54.77m \tag{4.3}$$

which is greater than the chosen layer thickness.



(a) Velocity in x-direction at rec. 1.

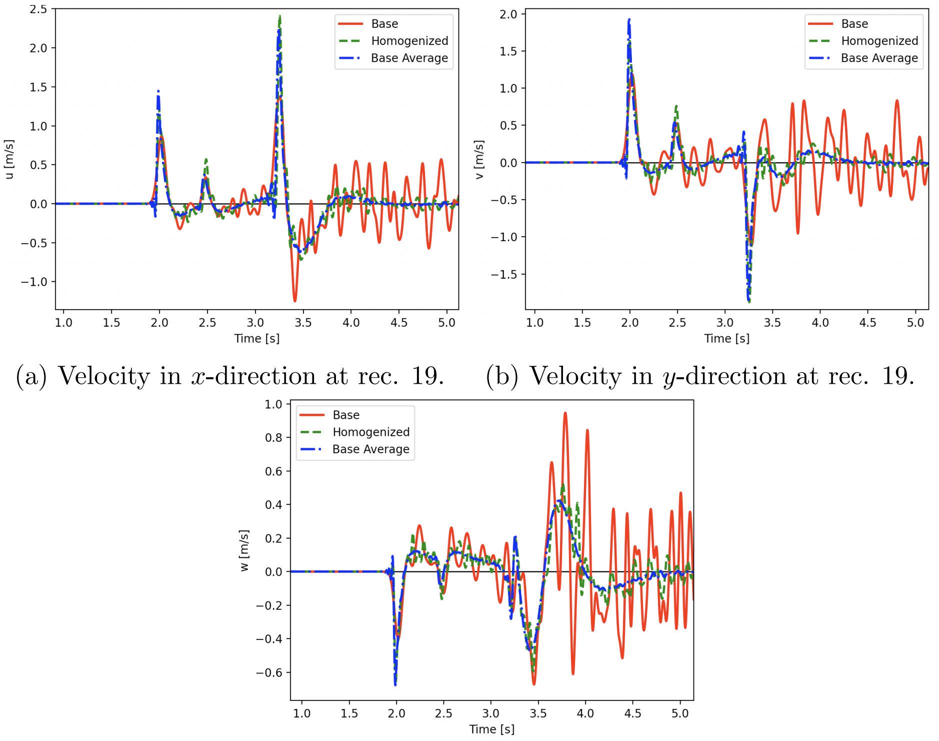
(b) Velocity in y-direction at rec. 1.

(c) Velocity in z-direction at rec. 1.

Figure 4.11: WP2_LOH1 velocity output at receiver 1 for alternating density on 50 m thick layers.

Figure 4.11 shows material velocities at receiver 1 in all three spatial directions. Compared to the homogenized implementation and the base implementation using a constant average density, the base implementation on the layered material stands out due to high-amplitude

oscillations. The homogenized and base average curves for v and w tend towards 0 while the base implementation shows diffuse fluctuations.
An element's barycenter is located in a layer with either of the two possible values for density. As a result, the density sampled for an element by the base implementation is subject to a systematic error of $+/-33.3\,\%$ of the average value. This can lead to a material boundary between two neighboring elements with different wave speeds. As indicated above, the wave is partially reflected in all directions at these interfaces. The oscillations in particle velocity are a visible result of these reflections. Moreover, they prove incapacity of the barycenter implementation to resolve discontinuities in material parameter distributions below the size of an element. The homogenized implementation delivers results very close to the base implementation with a constant average density by using multiple sampling points per element.
Particle velocities at receiver 19 are depicted in Figure 4.12, which substantiates above findings from the analysis of Figure 4.11. Strong oscillations in the barycenter implementation output are present consistently across all receivers. Nonetheless, the homogenized results at receiver 19 also contain some more visible fluctuations compared to receiver 1. This can be attributed to the fact that the accuracy of homogenization depends on the number of quadrature points. In the setup used here at 64 samples per element, homogenized densities still differ from the exact average by up to $5\,\%$. As a result, minor discontinuities in density are present in the homogenized output, too.



(a) Velocity in x-direction at rec. 19.

(b) Velocity in y-direction at rec. 19.

(c) Velocity in z-direction at rec. 19.

Figure 4.12: WP2_LOH1 velocity output at receiver 19 for alternating density on $50\,m$ thick layers.

Moving on now to investigate some deficiencies of the barycenter implementation compared to homogenization for material parameters described by continuous periodic functions.

4.3 Periodic Density Distributions

So far this chapter has focused on discontinuous, layered density distributions only. The following section will compare and discuss the performance of homogenization for material parameter distributions described by trigonometric functions. These are used in place of low-order polynomials with the advantage of being able to adapt their periodicity to mesh layer thickness.
In all simulations, a layered TPV5 scenario at a mesh resolution slightly lower than that from Subsection 4.2.1 is employed to compare the homogenized against the base implementation. The reasoning behind the low resolution is that apart from the higher resolution area around the fault, it can be observed that this way there is just one layer of elements between two mesh interfaces in most of the generated volume. Even though elements are structured irregularly in the actual mesh, it still holds that for a single layer of tetrahedra each of the enclosing interfaces contains at least one of each element's vertices. Also see the top-left and top-right corners in the previous layered TPV5 mesh from Figure 4.4b) for a two-dimensional visualization of such an arrangement.

4.3.1 Sine Density Distribution

A first simulation run uses an elastic material configuration with density given by

$$\rho(z) = 5000 \left| sin\left(\frac{\pi}{2000}z\right)\right| + 500. \tag{4.4}$$

Listing 4.2 shows the corresponding easi configuration with fixed Lamé parameters and a $|\sin(z)|$-function used to periodically alter density with depth.

```
!FunctionMap
map:
  rho:     return 5000 * abs(sin(0.001570796326795 * z)) + 500;
  mu:      return 32038120320;
  lambda:  return 32043759360;
```

Listing 4.2: easi configuration for periodic density using $|\sin(z)|$.

It can be seen in Figure 4.13 that half of the sine's period length coincides with a mesh layer of 2000 m thickness.

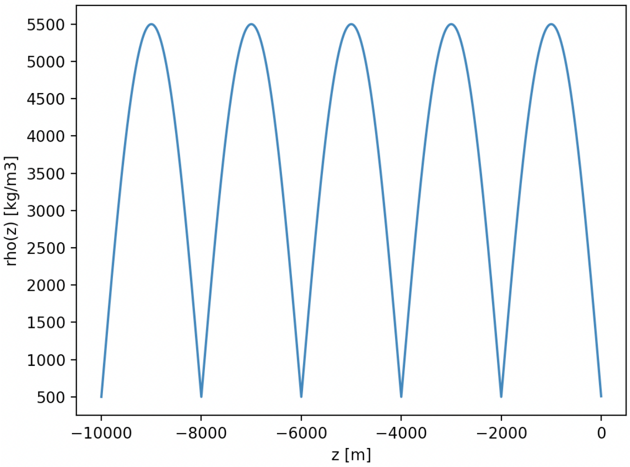


Figure 4.13: Density distribution for the first $10\,km$ of depth using a $|\sin(z)|$ function on a layered TPV5 scenario.

Also taking into account the above preliminary remarks regarding the mesh layout, let us look at a simple example to consider what to expect from the results:
We visualize a reference tetrahedron with edge length scaled up to $2000\,m$ so that it just fits between two interfaces. The element's volume is calculated using Equation 3.39 as

$$V_{ref} = \frac{1}{6}\left|\begin{bmatrix}2000\\0\\0\end{bmatrix}\cdot\left(\begin{bmatrix}0\\2000\\0\end{bmatrix}\times\begin{bmatrix}0\\0\\2000\end{bmatrix}\right)\right| = 1.\bar{3}\cdot 10^9\,m^3. \tag{4.5}$$

Considering Equation 4.4, the element's average density is given by

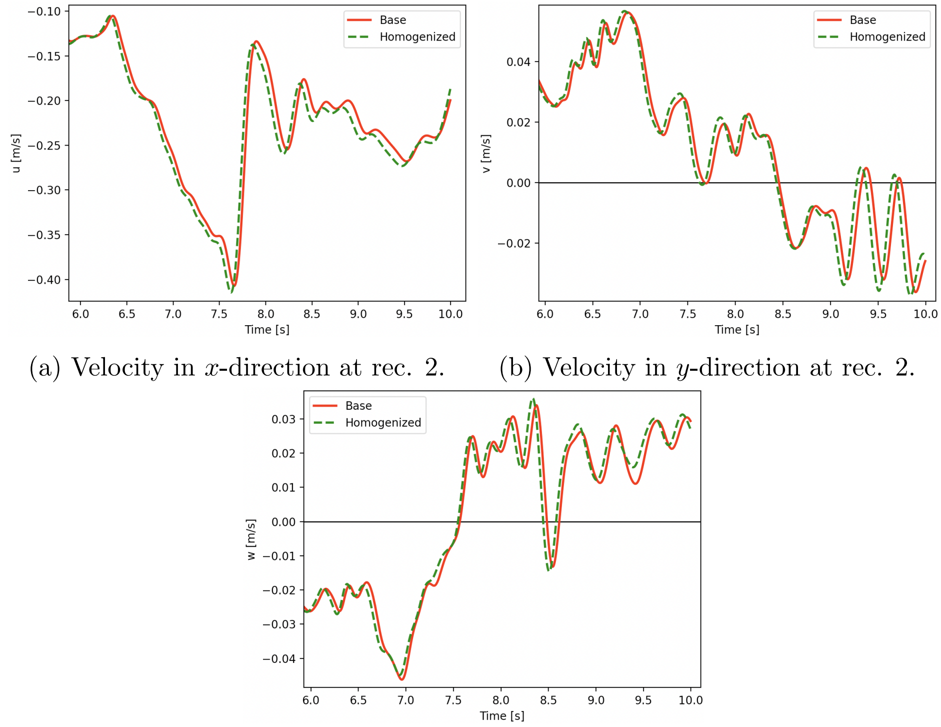
$$\begin{aligned}\bar{\rho}_{ref} &= \frac{1}{|V_{ref}|}\int_0^{2000}\int_0^{2000-x}\int_0^{2000-x-y} \rho(z)\,dzdydx\\ &= \frac{4.45274\cdot 10^{12}}{1.\bar{3}\cdot 10^9}\,\frac{kg}{m^3}\\ &= 3339.55\,\frac{kg}{m^3}.\end{aligned} \tag{4.6}$$

The barycenter z-component of such an element is located at $\frac{1}{4}$ or $\frac{3}{4}$ of its height, depending on its orientation. In both cases, if the element is located between interfaces i and $i+1$, the density sampled for this element by the base implementation is calculated as

$$\rho_{b,ref} = \rho(2000i + 500) = \rho(2000i + 1500) = 4035.53\,\frac{kg}{m^3} \tag{4.7}$$

which is higher than the averaged result from Equation 4.6.
Again, this is a contrived example to show the offset of barycentered compared to homogenized density for a regular tetrahedron in a single-layered element structure. In the real, more irregular mesh, varying barycenter locations contribute to a better averaging out of the density. Moreover, homogenized averaging relies on numerical quadrature and does not compute exact results, too. Therefore, the difference in density should be less pronounced in

the actual simulation. Indeed, the median ratio of homogenized ρ to barycenter ρ for all elements in this simulation run is measured at 0.9567 compared to the predicted ratio of $\frac{\bar{\rho}_{ref}}{\rho_{b,ref}} = 0.8275$.



(a) Velocity in x-direction at rec. 2.

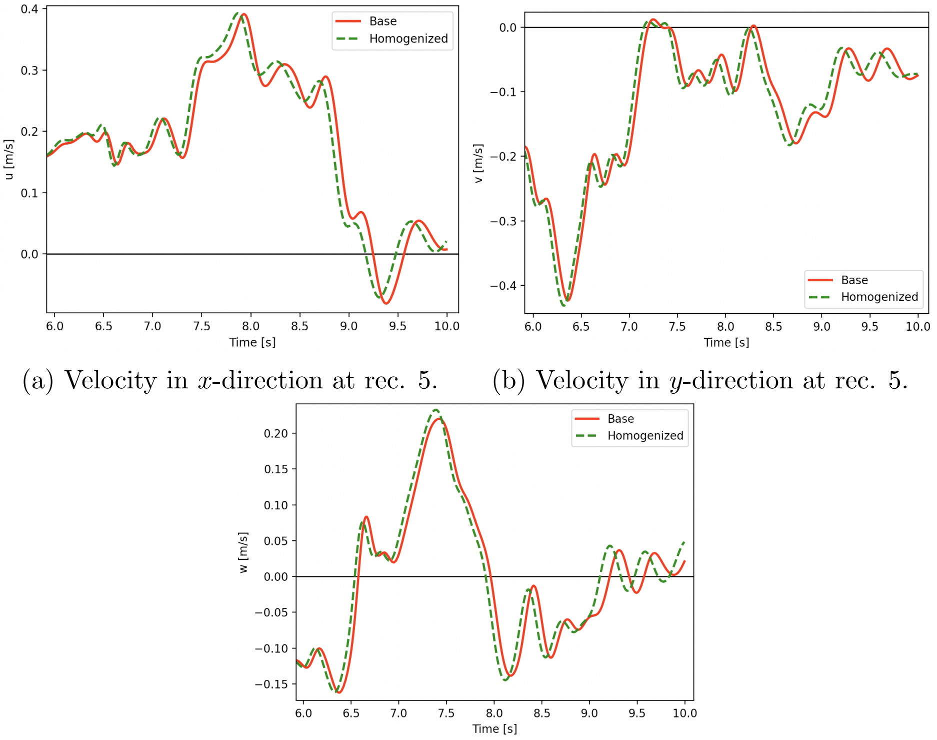
(b) Velocity in y-direction at rec. 2.

(c) Velocity in z-direction at rec. 2.

Figure 4.14: TPV5 velocity output at receiver 2 for periodic density using $|\sin(z)|$.

Figure 4.14 contains the last four seconds of velocity output at receiver 2. As expected, there is a minor yet ever-increasing lateral offset between both curves. The wave moves slightly faster when using the homogenized implementation owing to the difference in material density explained above. As predicted by Equation 4.1, a wave propagates faster through a less dense medium. Other than that, the overall congruence of both curves is remarkable as well. This may be an indication of similar material boundaries between elements resulting from both implementations. Most likely only few neighboring elements are assigned substantially different density values in both versions.

Particle velocities at receiver 5 can be seen in Figure 4.15. These are consistent with the above findings from the analysis of Figure 4.14. In fact, all receivers show a lateral shift between both curves for all recorded parameters with the homogenized wave propagating faster. At receiver 5, the offset of up to around 0.25 s seems to be even larger than at receiver 2. This suggests that the former is located further away from the fault.



(a) Velocity in x-direction at rec. 5.

(b) Velocity in y-direction at rec. 5.

(c) Velocity in z-direction at rec. 5.

Figure 4.15: TPV5 velocity output at receiver 5 for periodic density using $|\sin(z)|$.

4.3.2 Cosine Density Distribution

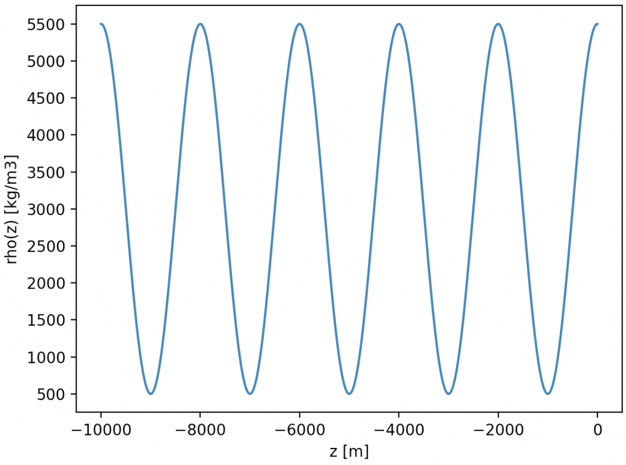


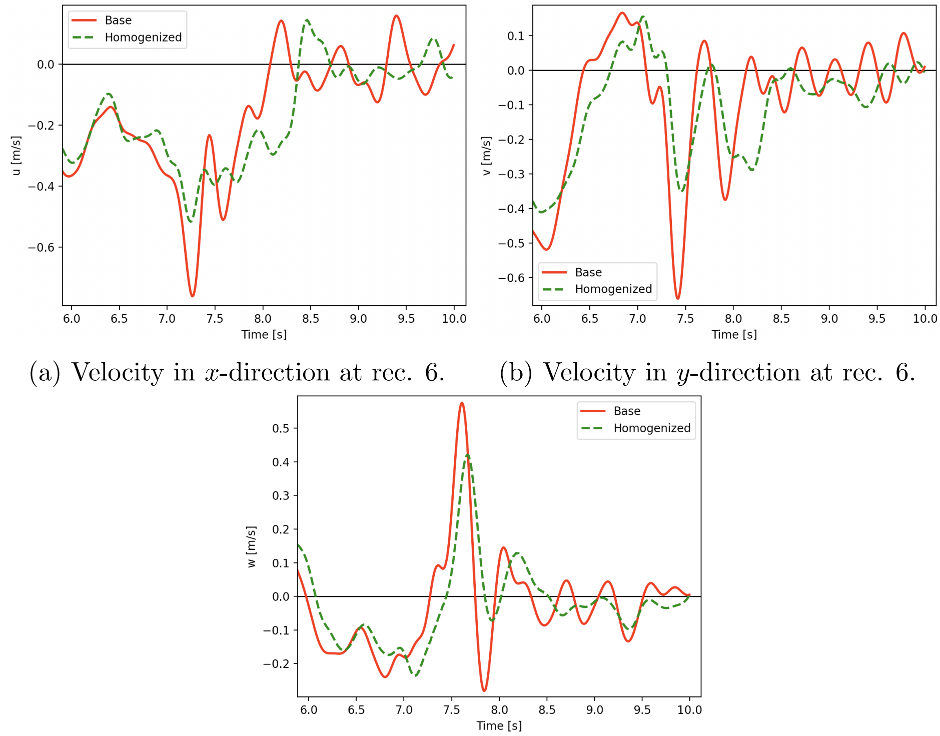
Figure 4.16: Density distribution for the first 10 km of depth using a $\cos(z)$ function on a layered TPV5 scenario.

The experiment from the previous subsection is repeated using

$$\rho(z) = 2500\cos\left(\frac{\pi}{1000}z\right) + 3000 \tag{4.8}$$

as a density function. The mesh remains unchanged and the easi setup is analogous to Listing 4.2. This time, the density distribution uses no absolute value and a full cosine period length coincides with a mesh layer of 2000 m thickness as illustrated in Figure 4.16. Similar assumptions hold about the expected results compared to Subsection 4.3.1. Because of the reduced period length and roughly inverse density curve compared to Figure 4.13, the offset between the resulting wave outputs should be inverted, too.

As presumed, the median ratio of homogenized ρ to barycenter ρ for all mesh elements is 1.1268. This is higher than the inverse of the same ratio observed in Subsection 4.3.1 which may be caused by the more compressed shape of a full cosine period instead of a half absolute sine period for density between two interfaces.



(a) Velocity in x-direction at rec. 6.

(b) Velocity in y-direction at rec. 6.

(c) Velocity in z-direction at rec. 6.

Figure 4.17: TPV5 velocity output at receiver 6 for periodic density using $\cos(z)$.

Looking at Figure 4.17, the last four seconds of particle velocity recorded at receiver 6 can be analyzed. A less clear result emerges in relation to the output from the previous subsection. The base and homogenized curves are less coherent overall which could be associated with reflections at different material boundaries in both setups.

Nonetheless, a lateral shift between the curves of both implementations is evident most of the time. For example at around 8.0 s the solid curve has a spike in velocity in x-direction which is shifted about half a second to the right for the homogenized velocity. The following parts of both u-curves are very similar and keeping the offset. The v- and w-curves display some minor yet regular phase shift between both implementations, too. This trend supports the results from the previous experiment. In this case, though, the base version's systematic error for single layers of elements between two mesh interfaces seems to be overlaid by some other unknown effects.

This section has attempted to provide a performance comparison of the homogenized implementation against the barycenter approach for continuous periodic functions used to model material density. A density distribution based on a $|\sin(z)|$ function in Subsection 4.3.1 has shown a lateral offset in wave velocity between both implementations. It has been identified as an effect related to single layers of mesh elements as a result of the low mesh resolution. It stands to reason that averaging material parameters only at a single point per element leaves the larger part of a complex parameter distribution out of consideration. The experiment has been repeated using a different trigonometric function in Subsection 4.3.2. This second simulation was able to confirm the initial results only to some extent because of overlaying effects present when using a $\cos(z)$ function for density.

4.4 Variation of Lamé Parameters

Moving on now to also perform validation and testing of homogenized Lamé parameters. The simulations done for this purpose are very similar to those from Subsection 4.3.1. Again, the same layered low-resolution TPV5 scenario is used. Equation 4.4 is adapted to deliver values in the appropriate range for λ and μ, which results in

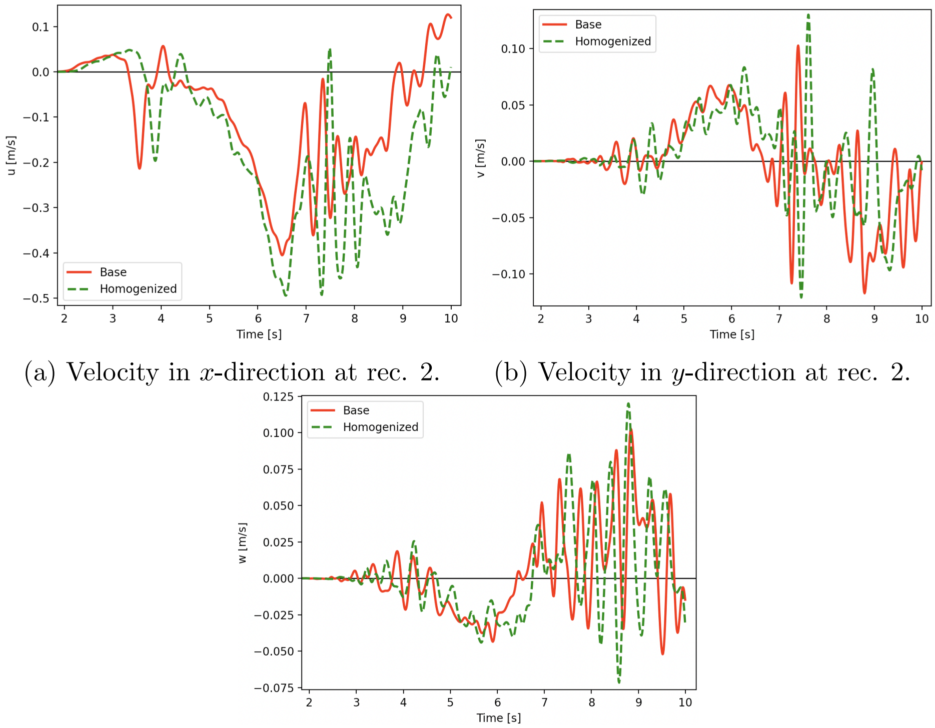
$$\lambda(z) = \mu(z) = 5 \cdot 10^{10} \left| sin\left(\frac{\pi}{2000} z\right)\right| + 5 \cdot 10^{9}. \tag{4.9}$$

In the easi file shown in Listing 4.3, density remains constant while Equation 4.9 is used to model λ and μ.

```
!FunctionMap
map:
  rho:     return 2760;
  mu:      return 50000000000*abs(sin(0.001570796326795*z)) + 5000000000;
  lambda:  return 50000000000*abs(sin(0.001570796326795*z)) + 5000000000;
```

Listing 4.3: easi configuration for periodic Lamé parameters using $|\sin(z)|$.

In Subsection 4.3.1, a systematic overestimation of density by the base implementation has been observed. A corresponding error can also be expected for the Lamé parameters. Unlike density which is considered in the denominator of the wave speed Equations 4.1, λ and μ are in its enumerator. This should lead to a reverse displacement of particle velocity curves compared to Figure 4.14 and Figure 4.15 where the homogenized wave propagates faster.
In contrast to density, a harmonic average is used for the homogenization of μ. Averaging λ is even more complex. Thus, it will be particularly interesting to compare the coherence of waves from both implementations apart from their expected lateral offset.
The median ratio of homogenized μ to barycenter μ and λ, respectively, is 0.5889 which is the highest discrepancy in all simulations so far.



(a) Velocity in x-direction at rec. 2.

(b) Velocity in y-direction at rec. 2.

(c) Velocity in z-direction at rec. 2.

Figure 4.18: TPV5 velocity output at receiver 2 for periodic Lamé parameters using $|\sin(z)|$.

The curves in Figure 4.18 illustrating particle velocity at receiver 2 are less congruent than those from previous experiments. As expected, a phase shift is present at some points in time. Between 3 - 5 s and from 7 s the homogenized wave lags behind the curve from the base implementation in x-direction. Similarly, between 6 - 8 s in y-direction the wave output using the barycenter implementation propagates faster. In z-direction, no clear trend is discernible.

In fact, the results are too cluttered in all spatial directions for the majority of time to make out any systematic difference between both curves. The more complex ways of averaging the Lamé parameters might contribute to more fundamental differences in the results generated by both approaches. This is most apparent at 8 - 10 s in x-direction where there is some major difference between the absolute values of both curves.

4.5 Variation of Viscoelastic Parameters

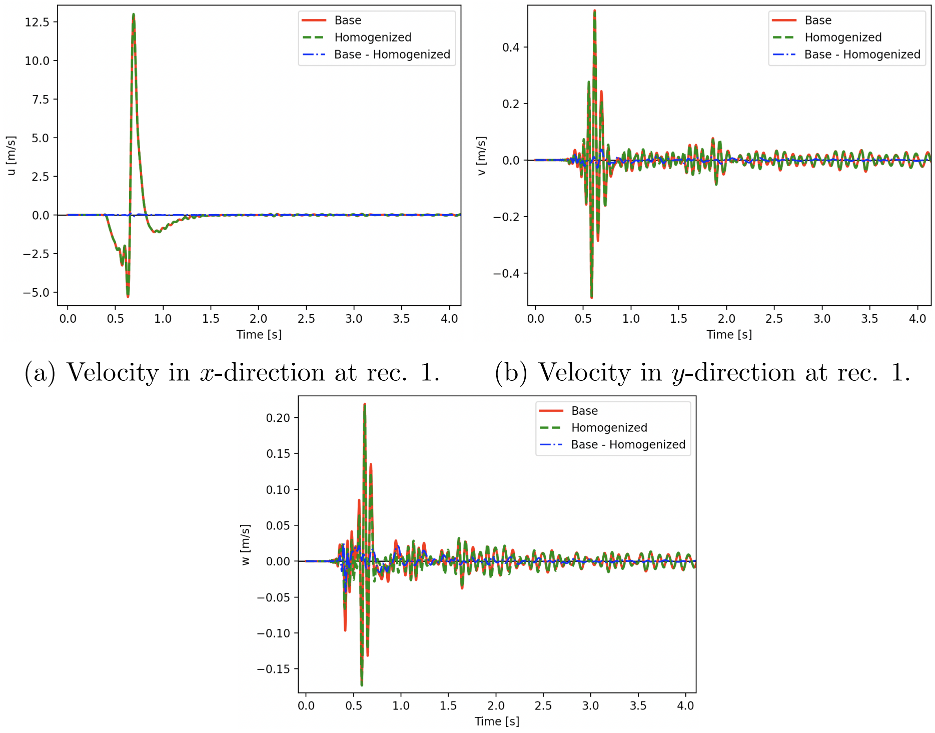
In this section, the effect of homogenization on the averaging of seismic quality factors in viscoelastic media will be investigated.

We reuse the unstructured WP2_LOH1 scenario described in Subsection 4.2.1 to run two simulations for the base and homogenized averaging of seismic Q. It features a viscoelastic material distribution with constant density and constant Lamé parameters. Q_p alternates between 80 and 120, Q_s between 240 and 320 every 50 m of depth. The easi configuration required for this setup can be seen in Listing 4.4.

```
!FunctionMap
map:
  rho:    return 3000;
  mu:     return 30000000000;
  lambda: return 20000000000;
  Qp:     return 80 + floor(sin(0.062831853071796 * z) + 1) * 40;
  Qs:     return 240 + floor(sin(0.062831853071796 * z) + 1) * 80;
```

Listing 4.4: easi configuration for alternating seismic Q on 50 m thick layers.

Seismic Q factors have no direct impact on a wave's velocity or reflection characteristics. As explained in Subsection 2.1.4, Q_p and Q_s determine the dissipation behavior of P and S waves. Accordingly, one would expect a high level of overall coherence between both simulation results. Since the physical material parameters ρ, λ and μ are set to be homogeneous, diffuse kinds of reflections at potential material boundaries should be rare. Instead, variations in wave amplitudes are likely to be the main differences between both versions.



(a) Velocity in x-direction at rec. 1.

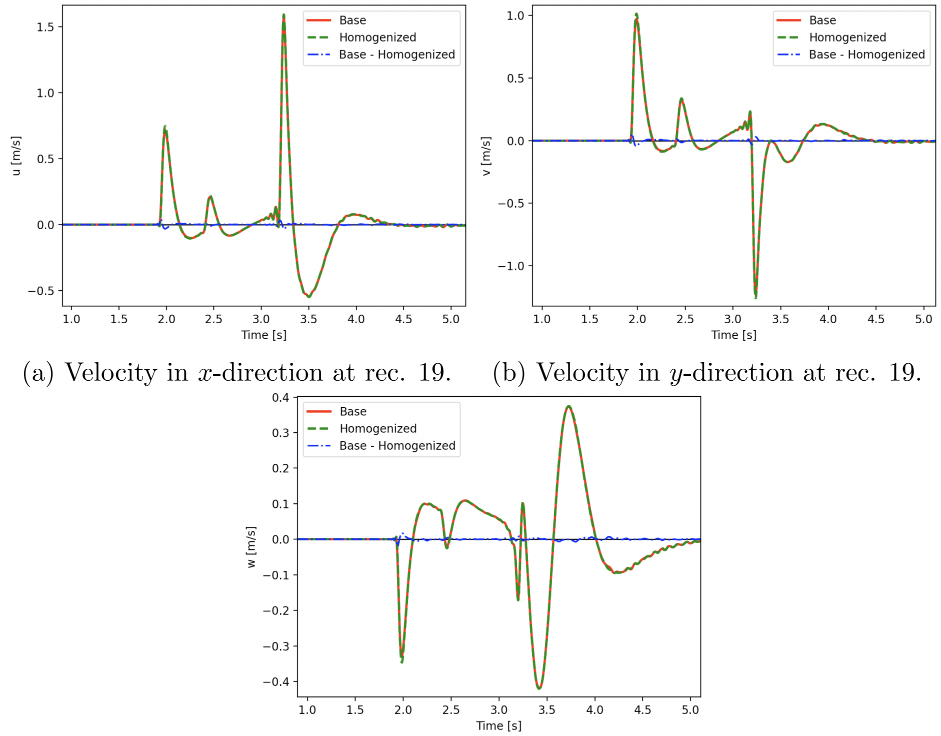
(b) Velocity in y-direction at rec. 1.

(c) Velocity in z-direction at rec. 1.

Figure 4.19: WP2_LOH1 velocity output at receiver 1 for alternating seismic Q on 50 m thick layers.

The resulting wave velocity outputs at receiver 1 can be seen in Figure 4.19. Both implementations deliver virtually identical curves in x-direction. As expected, some differences in wave amplitude are visible in the other spatial directions while the waves themselves are highly coherent. At first glance, it looks like the homogenized wave output is slightly more

attenuated, especially during the first $1.5\,s$ in z-direction. While this is true especially for some areas with isolated peaks like this, there are other points in time where the dashed curve overshoots the solid one, too.
Therefore, a definite statement on the impact of homogenization on the resulting wave output for varying viscoelastic parameters is not possible. It is worth mentioning, though, that just like for density in Subsection 4.2.2, the homogenized seismic Q values here represent more of an average of the discrete values alternating every $50\,m$. The base implementation always uses either of the two possible values for Q_p and Q_s, respectively. Consequently, the seismic Q factors sampled for an element by the barycenter implementation always deviate upwards or downwards from the average. This might be an important factor in the deviations in amplitude between both curves, even though no regularity can be found.



(a) Velocity in x-direction at rec. 19. (b) Velocity in y-direction at rec. 19.

(c) Velocity in z-direction at rec. 19.

Figure 4.20: WP2_LOH1 velocity output at receiver 19 for alternating seismic Q on $50\,m$ thick layers.

Figure 4.20 presents particle velocities for the same setup recorded at receiver 19. The results are nearly identical for all three spatial directions. However, it should be noted that oscillations at this receiver are rare compared to those in y- and z-directions at receiver 1. In summary, the effect of homogenization of seismic Q on the resulting wave output seems to be rather negligible. With a similar alternating material distribution for density, major errors in the base velocity output have been shown in Subsection 4.2.2. For Q_p and Q_s, however, the arithmetic averaging presented in Subsection 3.2.3 seems to be a reasonable trade-off between physical accuracy and implementation effort. Nonetheless, a potentially more precise homogenization of seismic Q will be briefly addressed in Chapter 5.

5 Conclusions

Within the scope of this thesis, a homogenization technique for material parameters in seismic simulations has been derived. It compares with the state of the art approach using barycenter averaging to approximate the values of material parameters per element.
Based on numerical quadrature, the homogenization approach has been successfully integrated into the SeisSol simulator. Its entire functionality fits into the existing code structure by means of a new QueryGenerator subclass. Material parameters are initialized after a mesh has been loaded and before the actual simulation starts. As a result, the more sophisticated computation of material parameters has virtually no impact on runtime performance. It scales only with the amount of mesh elements and the number of quadrature points.

In the first place, the homogenization approach for material parameters has proven to be correct. No major deviations from the respective expected or known exact results have been observed for all simulations conducted in the context of this work. Furthermore, the homogenization technique produces more accurate results in multiple cases where the barycenter approach is subject to systematic errors.
For thin material layers of alternating density in Subsection 4.2.2, the benefit from homogenization is most apparent. The homogenized approach generates results very close to the exact reference output with only minor fluctuations. However, strong oscillations in particle velocity output are visible when using the base implementation. They occur due to reflections at material boundaries between elements. Homogenization helps to better average out the parameter values sampled for elements stretching across multiple layers. It can be concluded that the homogenized approach is capable of producing results of equal or superior accuracy at a lower mesh resolution than the state of the art.
Using a discontinuous density profile with wide layers like in Subsection 4.2.1, other important insights can be gained. For both TPV5 and WP2_LOH1, the base implementation on a fine mesh yields the most inaccurate results. It frequently overshoots the exact curves and shows oscillations which are not present on the other outputs. Instead, the smaller element size creates more material boundaries between neighboring elements. Consequently, a higher resolution mesh not necessarily leads to more accurate and especially smoother output curves. Moreover, the homogenized approach best approximates the exact result in the WP2_LOH1 scenario. Particularly around points with spikes in the seismograms, homogenization sets itself apart from the barycenter approach and closely follows the output generated using a layered mesh.
For continuous material profiles like the ones used in Section 4.3, barycenter averaging shows to be incapable of resolving more complex parameter distributions below the size of an element. This gives rise to an ever-increasing phase shift in particle velocity output compared to the homogenized implementation that uses multiple sampling points per element. In Section 4.4, however, homogenization of the Lamé parameters leads to some more irregular differences from the results generated by the base implementation. It is difficult to determine

which of both outputs represents the more accurate result since we cannot identify any systematic deviation. Technically, homogenization computes better averages and should therefore be considered the superior solution. Under that assumption, this might be the most significant result because it reveals some fundamental errors induced by barycenter averaging which would require further investigation to fully understand them.
Unlike the elastic material parameters, seismic quality factors do not seem to benefit from homogenization. At least, the experiments outlined in Section 4.5 are not able to evoke an environment in which a noticeable error using barycenter averaging would occur. As expected, some differences in wave amplitude could be observed between both implementations. These have been too small, though, to have any significant effect associated with them. Anyway, the homogenized implementation generally samples more averaged out parameter values. In theory, this should yield a more exact result for elements extending over multiple layers.

With respect to the research question given in the beginning, it can be concluded that it is generally sensible to apply homogenization of material parameters in seismic simulations. Especially in scenarios involving layered material distributions, the homogenized approach allows for greatly improved averaging of material parameter values. Nonetheless, other scenarios and parameters have been found which do not obviously benefit from homogenization. Regardless of the accuracy gained in a particular seismic scenario, the effect of homogenization comes at virtually no added cost. Additionally, no scenario is conceivable in which the homogenized approach is subject to an error not committed by the existing implementation. Therefore, in practice, the accuracy of seismic simulations can only benefit from homogenized material parameters.

Future work in this area may focus on the implementation of homogenization for poroelastic and anisotropic materials. In Subsection 3.2.4, a homogenization approach for all poroelastic material parameters used in SeisSol has been derived. In most cases, simple arithmetic or harmonic averages have been identified as the correct means of averaging. No complex calculations like for the first Lamé parameter seem to be necessary. This thesis does not further pursue the investigation of poroelastic materials. This is because the need to conceive promising scenarios and evaluate results for all of the added parameters would go beyond the scope of this work. Similarly, the anisotropic case features yet more material parameters which have not been discussed so far. Homogenization of these parameters requires extensive additional work.
Furthermore, it has been indicated in Subsection 3.2.3 that the choice of an arithmetic average for seismic quality factors is just a first approximation. It serves to provide an initial assessment of the effect of homogenization on seismic Q. As discussed above, the homogenized particle velocity output did not substantially differ from the results obtained using the existing implementation. It is unlikely that a more accurate homogenization approach would yield very different results in these particular scenarios. Nonetheless, other environments might be conceived which are more suited to show systematic errors in the seismic output related to barycenter averaging of seismic Q. In that case, the work of Kristek and Moczo [KM03] is a suitable starting point for an extension of the homogenization approach to viscoelastic media. Additional work will be necessary to prepare and implement their findings about homogenization of seismic Q in SeisSol.

List of Figures

List of Tables

List of Listings

Bibliography

[AHH$^+$21] S. Ahlers, A. Henk, T. Hergert, K. Reiter, B. Müller, L. Röckel, O. Heidbach, S. Morawietz, M. Scheck-Wenderoth, and D. Anikiev. 3D Crustal Stress State of Germany According to a Data-Calibrated Geomechanical Model. *Solid Earth*, 12(8):1777–1799, 2021.

[BM06] Heiko Bauke and Stephan Mertens. *Cluster Computing*. X.systems.press. Springer-Verlag Berlin Heidelberg, 2006.

[Brz18] Dariusz W. Brzezinski. Computation of Gauss-Jacobi Quadrature Nodes and Weights with Arbitrary Precision. In *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems*, volume 15, pages 297–306, 2018.

[Che16] Alexander H.-D. Cheng. *Poroelasticity*. Theory and Applications of Transport in Porous Media. Springer Cham, 2016.

[DK06] Michael Dumbser and Martin Käser. An Arbitrary High-Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes — II. The Three-Dimensional Isotropic Case. *Geophysical Journal International*, 167(1):319–336, 2006.

[Fas22] Manuel Fasching. ImpalaJIT. `https://github.com/Manuel1605/ImpalaJIT`, 2022. Accessed: 2022-06-26.

[GL20] Paul W. J. Glover and Miao Luo. The Porosity and Permeability of Binary Grain Mixtures. *Transport in Porous Media*, 132:1–37, 2020.

[GN49] L. Grunberg and Alfred H. Nissan. Mixture Law for Viscosity. *Nature*, 164:799–800, 1949.

[GR09] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309 – 1331, 2009.

[HT13] Nicholas Hale and Alex Townsend. Fast and Accurate Computation of Gauss–Legendre and Gauss–Jacobi Quadrature Nodes and Weights. *SIAM Journal on Scientific Computing*, 35(2):A652–A674, 2013.

[INT17] INTERTWinE Consortium. Best Practice Guide to Hybrid MPI + OpenMP Programming. `http://www.intertwine-project.eu/sites/default/files/images/INTERTWinE_Best_Practice_Guide_MPI%2BOpenMP_1.1.pdf`, 2017. Accessed: 2022-07-06.

[KBH02] Philip Kearey, Michael Brooks, and Ian Hill. *An Introduction to Geophysical Exploration.* Wiley-Blackwell, 2002.

[KM03] Jozef Kristek and Peter Moczo. Seismic-Wave Propagation in Viscoelastic Media with Material Discontinuities: A 3D Fourth-Order Staggered-Grid Finite-Difference Modeling. *Bulletin of the Seismological Society of America*, 93(5):2273–2280, 2003.

[KP21] Martin Kronbichler and Per-Olof Persson. *Efficient High-Order Discretizations for Computational Fluid Dynamics.* CISM International Centre for Mechanical Sciences. Springer Cham, 2021.

[KUU$^+$21] Lukas Krenz, Carsten Uphoff, Thomas Ulrich, Alice-Agnes Gabriel, Lauren S. Abrahams, Eric M. Dunham, and Michael Bader. 3D Acoustic-Elastic Coupling with Gravity: The Dynamics of the 2018 Palu, Sulawesi Earthquake and Tsunami. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, 2021.

[LeV04] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2004.

[Mar21] Martin Čuma. Hybrid MPI/OpenMP Programming. `https://www.chpc.utah.edu/presentations/images-and-pdfs/mpi-openmp21f.pdf`, 2021. Accessed: 2022-07-06.

[MKV$^+$02] Peter Moczo, Jozef Kristek, Vaclav Vavrycuk, Ralph Archuleta, and L. Halada. 3D Heterogeneous Staggered-Grid Finite-Difference Modeling of Seismic Motion with Volume Harmonic and Arithmetic Averaging of Elastic Moduli and Densities. *Bulletin of the Seismological Society of America*, 92(8):3042–3066, 2002.

[Mor19] Igor B. Morozov. Mechanics of Seismic Q. *CSEG RECORDER*, 44(1):20–24, 2019.

[Sei06] Seismic Wave Propagation and Imaging in Complex Media: A European Network (SPICE). Problem WP2_LOH1. `http://www.sismowine.org/model/WP2_LOH1.pdf`, 2006. Accessed: 2022-06-05.

[Sou05a] Southern California Earthquake Center (SCEC). TPV5. `https://strike.scec.org/cvws/download/TPV5_forwebsite.pdf`, 2005. Accessed: 2022-06-05.

[Sou05b] Southern California Earthquake Center (SCEC). Views of TPV5 Overall and Fault Geometries. `https://strike.scec.org/cvws/download/faultstationsTPV5.pdf`, 2005. Accessed: 2022-06-05.

[Str71] Arthur H. Stroud. *Approximate Calculation of Multiple Integrals.* Prentice-Hall Series in Automatic Computation. Prentice-Hall, 1971.

[Str74] Arthur H. Stroud. *Numerical Quadrature and Solution of Ordinary Differential Equations*, volume 10 of *Applied Mathematical Sciences.* Springer, 1974.

[SW03] Seth Stein and Michael Wysession. *An Introduction to Seismology, Earthquakes, and Earth Structure.* Wiley-Blackwell, 2003.

[Sé20] Sébastien Brisard. What Is Homogenization? Part 3: Ensemble Averages vs. Volume Averages. `https://sbrisard.github.io/posts/20200421-what_is_homogenization-03.html`, 2020. Accessed: 2022-06-30.

[Tea22a] SeisSol Team. easi Examples. `https://github.com/SeisSol/easi/tree/master/examples`, 2022. Accessed: 2022-06-26.

[Tea22b] SeisSol Team. SeisSol – High Resolution Simulation of Seismic Wave Propagation in Realistic Media with Complex Geometry. `https://github.com/SeisSol/SeisSol`, 2022. Accessed: 2022-06-30.

[Tea22c] SeisSol Team. SeisSol Documentation. `https://seissol.readthedocs.io`, 2022. Accessed: 2022-06-26.

[Tea22d] SeisSol Team. SeisSol Homepage. `https://www.seissol.org`, 2022. Accessed: 2022-06-26.

[Uni22] United States Geological Survey (USGS). What Is a Fault and What Are The Different Types? `https://www.usgs.gov/faqs/what-fault-and-what-are-different-types`, 2022. Accessed: 2022-06-23.

[Uph22] Carsten Uphoff. easi Documentation. `https://easyinit.readthedocs.io`, 2022. Accessed: 2022-06-26.

[WGU$^+$22] Sebastian Wolf, Martin Galis, Carsten Uphoff, Alice-Agnes Gabriel, Peter Moczo, David Gregor, and Michael Bader. An Efficient ADER-DG Local Time Stepping Scheme for 3D HPC Simulation of Seismic Waves in Poroelastic Media. *Journal of Computational Physics*, 455:110886, 2022.

[ZBB$^+$20] Assem Zharbossyn, Zhazira Berkinova, Aidana Boribayeva, Assiya Yermukhambetova, and Boris Golman. Analysis of Tortuosity in Compacts of Ternary Mixtures of Spherical Particles. *Materials*, 13(20):4487, 2020.