New Theoretical and Numerical Methods for Wave-Motion Modeling and Optimization

Marcus Michael Noack

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

The Faculty of Mathematics and Natural Sciences

University of Oslo

May 2017

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 1861

ISSN 1501-7710

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Preface

As a child, my daily school routine was not distinguished by great excellence. Characterizing my achievements in school as average would be considered a compliment. In 2006, when I graduated from high school, I was told that I could pursue any career, but I should stay away from mathematics. If anything, this comment motivated me to become the physicist I am today.

In 2014, I received an industrial PhD scholarship with Kalkulo AS, a subsidiary of Simula Research Laboratory. My time at Simula Research Laboratory was full of interesting discussions which formed me as a researcher. I want to start by thanking Markus Mueller and Martin Krause. Both mentored me during my last two years at the University in Jena, with incredible patience and creativity, and motivated me to become the researcher I needed to be to successfully finish my PhD. Lengthy debates with both of them, long after work hours, raised my interest in theoretical physics and mathematics. A special thanks goes to my supervisors, Stuart Clark and Are Magnus Bruaset, who always gave me the freedom to follow my own ideas but were helpful when needed. I want to mention some colleagues who played a special role during my PhD work. First of all Kristin McLeod, who is responsible for strengthening my communication skills. She was a role model for resourcefulness, especially in the first months of the PhD. An important friend and colleague, also especially in the important first months of the PhD, was Tor Gillberg who had the patience to explain and answer every single question I had. I also want to thank my parents and friends who did not see a lot of me when work kept me busy, which was always the case. During the end of my PhD, the questions I faced became more difficult to answer. Some of them would have never been answered if it hadn't been for Simon Funke. He played a large role in my professional development. I can only give my best to asymptotically approach his expertise. Last but not least, I want to thank Kalyn Marie Hanna who had to cope with me working long hours and being moody when I was finally home. She is my rock and kept me sane during the last year. Thanks to her for also proofreading the thesis.

Thank you all. This thesis is dedicated to all of you.

Contents

Chapter 1

Introduction

As a devoted physicist, I believe that the universe we live in is governed by only a few fundamental principles. Without a doubt, optimization and wave propagation are two of them. The impact of wave propagation as a principle is difficult to comprehend. We encounter wave phenomena everywhere, and in all scales imaginable, ranging from the vibration of sub-atomic strings of the unverified string theory and the probability waves of quantum mechanics, all the way up to the recently discovered gravity waves predicted by Einstein as part of his general theory of relativity. The fundamentality of wave propagation is only superseded by optimization. In fact, wave propagation is a consequence of nature trying to optimize energy flow. Virtually everything in physics can be traced back to optimization. Both principles cannot only give us a tremendous amount of information about the very basic fabric of our universe, they can also shape our everyday lives in a fundamental way. This thesis attempts to outline and describe the importance of understanding how wave propagation and optimization play a role in our everyday lives and ways in which we can utilize these concepts to improve upon them.

In 1909, a Croatian scientist observed two distinct signals from a regional earthquake. He noticed a travel-time difference in the signals and discovered a discontinuity in the elastic properties between the Earth's crust and mantle, today commonly referred to as Mohoviĉić discontinuity. This was one of the first times wave phenomena were used to obtain information about the interior structure of a body [37]. Since this first use of a wave signal to image the sub-surface, a vast number of methods have been developed to image interiors of bodies. Commonly referred to as wave imaging, this field comprises a multiplicity of different methods and requires a high degree of knowledge and expertise. Wave imaging is composed of two main steps: the forward modeling of a wave and the inverse or optimization step. Different components of the wave field, such as travel times, amplitudes or wave-form spectra can be used for wave imaging depending on required computing times and resolution of the output image. In the forward step, the chosen components are modeled with respect to a deliberately or randomly defined parameter model of the interior of the body. The parameters of the model are adapted during the inverse step to minimize the misfit between observed and calculated data. The two main steps of wave imaging, wave modeling and inversion, play important roles in a vast array of applications.

Wave-motion modeling is the basis for earthquake modeling. In fact, some of the fastest and most sophisticated wave-motion simulation tools arose from this field [14]. Wave-motion modeling had a large impact on acoustics [9], where the short wavelengths of waves result in large computational domains. The medical field is one of the biggest beneficiaries of wave-motion modeling. Electrophysiological modeling

[41] and ultrasound [40] are only a small subset of applications. In short, every field that uses waves, such as electro-magnetic (communication), water (tsunami), air (acoustics, atmospheric), elastic (earthquake, exploration, material), gravity or any other kind of wave can benefit from advancements achieved in this field.

The number of applications of methods for wave-motion computations, though vast, is dwarfed by the number of applications of methods for function optimization. In fact, wave propagation itself is an application of optimization since a wave travels the path of minimal time. But, wave propagation is no exception. All physical phenomena can be formulated as an optimization problem, which is the motivation of Hamiltonian mechanics, which in turn, builds the basis for many principles in quantum mechanics. In the theory of relativity, masses move in space-time along shortest paths, the so-called "geodesics". Besides being woven into the deep fabric of our universe, optimization has many technical applications. Optimization is indispensable in engineering, finance, the medical field and power plant design. Every plane and car we use on a daily basis has parts designed with the help of optimization methods.

We have seen the vast importance of wave-motion computations and optimization. However, both fields pose challenging problems. Modeling of wave motion can be a complex and time-consuming process. The simulation can involve the computation of more than $10e^{12}$ nodes. To build efficient solvers, analytical and numerical methods must be tailored to take full advantage of novel parallel computer architectures. In the inverse step, we are facing high-dimensional parameter spaces and frequent occurrences of local optima. Given the number of dimensions, the employment of purely stochastic optimization procedures is unfeasible. Local optimization procedures, on the other hand, struggle to find the global optimum.

The presented work demonstrates how newly developed theoretical and numerical methods help save computing time and resources when performing wave-motion modeling and optimization. I will show how the saved computing power can be combined with novel optimization methods to increase the quality of wave-data inversions. The perfect cooperation of methods for wave modeling and inversion leads to a more efficient imaging of the interior of objects and sub-surfaces, and can have other far reaching implications. For the reader to fully understand this thesis, it is necessary to introduce some rudimentary knowledge about wave-motion modeling and optimization, which will be taken care of in the following sections.

1.1 Motivation of the Work

As seen in the last section, the list of fields that use wave-motion computation is long and includes radiology, archaeology, biology, atmospheric science, geophysics, oceanography, plasma physics, materials science, astrophysics and quantum physics, along with many others. Therefore, improving the methods used in wave imaging would potentially have a great impact on many fields. In cardiac modeling, a wave can be propagated through the heart tissue to model an electrical signal [50]. Water-wave simulations are used to model tsunamis for hazard assessment [20]. Sound waves are modeled to assess the perfect geometry for an entertainment center, opera house or concert hall [32]. Other applications include earthquake hazard assessment [35], oil and gas exploration, and vibration modeling in architechtural or machine elements [1]. In short, efficient and accurate wave-motion modeling can save lives and help supply energy to Earth's growing population.

The inversion or optimization step is a basic step for wave imaging [7, 17], but is also indispensable in many other applications in engineering and economics. In fact, optimization is one of the most

fundamental principles of our universe, and a list of examples and applications in influential fields could fill an entire library.

As outlined above, wave imaging is only one application of wave-motion modeling and optimization, among many others. The work was funded by Kalkulo AS and the Research Council of Norway as part of the Industrial PhD scheme. The work conducted at Kalkulo AS aims to improve seismic imaging for oil and gas exploration, but also the numerous other applications and the tremendous positive impact on many other fields motivated this work.

1.2 Modeling Wave Motion

A large part of the thesis will focus on the wave equation

$$\begin{aligned} &\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} - c(\mathbf{x})^2 \nabla^2 u(\mathbf{x},t) = f(\mathbf{x},t) \\ &u(\mathbf{x},0) = 0 \\ &\frac{\partial u(\mathbf{x},0)}{\partial t} = 0 \end{aligned} \tag{1.1}$$

and its variations and approximations. Here $u(\mathbf{x},t)$ is commonly interpreted as pressure or amplitude, $c(\mathbf{x})$ is the spatially dependent wave speed, $f(\mathbf{x},t)$ is the source function and ∇ is the nabla symbol. Plane waves are the simplest solution of the wave equation [47]. By substituting the plane wave solution for high frequencies into the wave equation, we can derive a frequency-dependent form of the acoustic wave equation

$$-\omega^2 A(\mathbf{x}) \left(|\nabla T(\mathbf{x})| - \frac{1}{c(\mathbf{x})} \right) + i\omega \left(2\langle \nabla A(\mathbf{x}), \nabla T(\mathbf{x}) \rangle + A(\mathbf{x}) \nabla^2 T(\mathbf{x}) \right) + \nabla^2 A(\mathbf{x}) = 0, \tag{1.2}$$

where $A(\mathbf{x})$ is an amplitude field, $T(\mathbf{x})$ is a time field and ω is the wave frequency. Equation (1.2) must be satisfied for any frequency ω [47]. Therefore, all sub-expressions in equation (1.2) must vanish independently, which results in three equations. The equation

$$|\nabla T(\mathbf{x})| = \frac{1}{c(\mathbf{x})} \tag{1.3}$$

is called the "eikonal equation". The solution of equation (1.3) gives first-arrival travel times of a propagating wave and equals the solution of the wave equation for infinitely high frequencies. The eikonal equation has a large variety of applications and will be discussed in the following. The equation

$$2\langle \nabla A(\mathbf{x}), \nabla T(\mathbf{x}) \rangle + A(\mathbf{x}) \nabla^2 T(\mathbf{x}) = 0, \tag{1.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, is called the "transport equation" and describes the amplitude of a wave. The equation

$$\nabla^2 A(\mathbf{x}) = 0 \tag{1.5}$$

is commonly neglected if the aim is to model high frequencies of the propagating wave.

Another approach to obtain an approximation to the wave equation is to ignore its time-dependent part. The result is the "Helmholtz equation". The Helmholtz equation in the frequency domain

$$A(\mathbf{x},\omega)(\nabla^2 + \frac{\omega}{c(\mathbf{x})^2}) = 0 \tag{1.6}$$

describes the solution of the wave equation for a certain frequency and is obtained by applying separation of variables to the wave equation. For a point source we can define the Helmholtz Green's function as the solution of

$$\nabla^2 G(\mathbf{x}, \omega) + \frac{\omega^2}{c(\mathbf{x})^2} G(\mathbf{x}, \omega) = -\delta(\mathbf{x} - \mathbf{x_0}), \tag{1.7}$$

[31], where $\delta(\mathbf{x} - \mathbf{x_0})$ is the Dirac delta function at the position $\mathbf{x_0}$. The solution in three dimensions is given by

$$G(\mathbf{x}, \omega) = A(\mathbf{x}) e^{i\omega T(\mathbf{x})}, \tag{1.8}$$

where $A(\mathbf{x})$ is the amplitude field and $T(\mathbf{x})$ is the travel-time field [27]. Given a travel-time field and an amplitude field, we can assemble the Helmholtz Green's function.

The eikonal equation plays an important role in seismic imaging since it is the backbone of travel-time inversion, which is still one of the most popular techniques to image the sub-surface. Furthermore, travel-time inversions can be performed prior to wave-form inversions to yield an accurate initial model. An advanced initial model for the wave-form inversion can render the parameter space of the inversion quasi-linear, which increases the probability to converge into the global optimum significantly. The result of the travel-time inversion can be improved by including amplitude information in the computation. The eikonal equation influences many fields. In fact, whenever a front has to be tracked based on an underlying velocity model, the eikonal equation has to be solved. This problem is common in computer vision, navigation, path optimization and many more applications. A large number of eikonal solvers have been developed over the last decades. The expanding-square or expanding-box methods [48, 49] compute the solution in a box shape around the source. The method suffers in accuracy for complex velocity models. The fast-marching method [42, 37] represents an improvement but is sequential by nature. The fast-sweeping method [52] computes the solution of the eikonal equation iteratively and can be parallelized to some extent. A very efficient and highly parallelized way to compute travel times and amplitudes is presented in Research Paper 1: "Fast Computation of Eikonal and Transport Equations on GPU Computer Architectures". The computed amplitudes and travel times can directly be used to assemble the Helmholtz Green's function (1.8). Despite the success of the eikonal equation, its solution is a harsh approximation and valid only for specific assumptions.

To avoid the drawbacks of the solution of the eikonal equation, the wave equation (1.1) can be solved directly. Wave propagation is subject to causality. Therefore, active regions of wave propagation are traveling through space and are not randomly appearing. This fundamental physical characteristic gives rise to a computational method which separates active from inactive regions of the wave propagation. Active in this context means that the wave exhibits amplitudes greater than a certain threshold. Only active regions need to be computed, which can save computing resources and time. The Research Paper 2, "A Two-Scale Method using a List of Active Sub-Domains for a Fully Parallelized Solution of Wave Equations", proposes an algorithm that carefully selects active sub-domains and assigns processing units accordingly. The method was patented in the USA.

Including anisotropy was proven to be a crucial step of wave-motion modeling in the 1960s [29]. One interesting way to start the theoretical treatment of anisotropy is to look at velocity surfaces and the corresponding dispersion relations. The dispersion relations can be transformed to different metric spaces to derive different wave, eikonal and transport equations governing wave motion in different kinds of anisotropic media. A description of this exclusively analytical method is given in the Research Paper 3, "An Acoustic Wave Equation in a Transformed Metric Space for Various Types of Anisotropy".

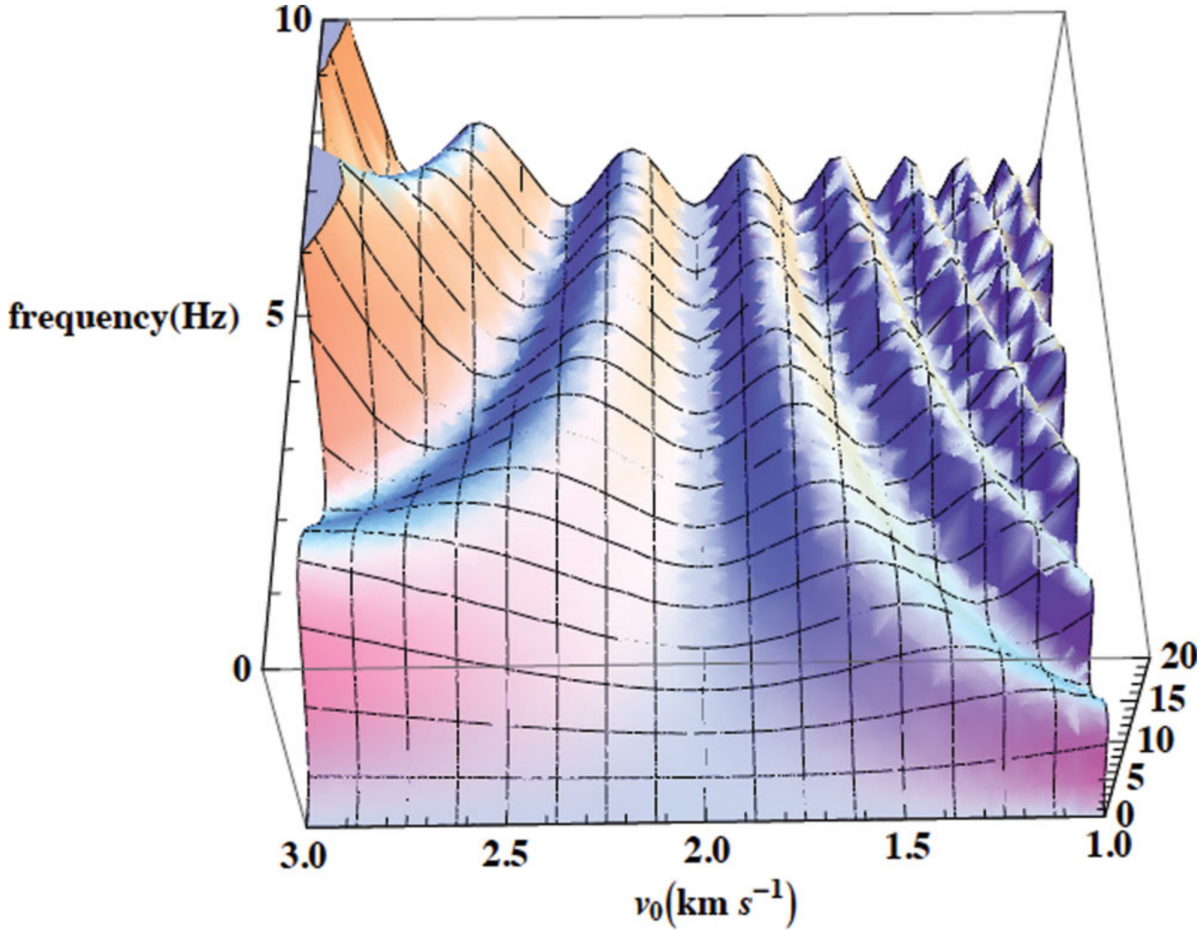


Figure 1.1: An example of a misfit functional. The optimization is a challenging task because of the plurality of local optima. Image courtesy of Alkhalifah and Choi [5].

1.3 Function Optimization and Selected Applications

Inversion is the adaption of a model to be able to explain some observation. It is a main building block of partial-differential-equation(PDE)-constrained optimization. During the inverse step, the model parameters need to be adapted in order to optimize the misfit or objective functional

$$\Psi(\mathbf{m}) = ||\mathbf{L}(\mathbf{m}) - \mathbf{d_0}||, \tag{1.9}$$

where $\mathbf{m}$ represents the model parameters, $\mathbf{d_0}$ is the data set, $\mathbf{L}$ is the modeling operator and $||\cdot||$ is a norm. The operator $\mathbf{L}$ can, for instance, represent a partial differential equation (PDE), such as the wave equation. The optimization of the objective functional defined in equation (1.9) can be extremely challenging because of the high degree of non-linearity (Figure 1.1). In case the operator $\mathbf{L}$ represents the wave equation, there are two main reasons for the non-linearity of the objective functional: firstly, the oscillatory nature of the wave field, and secondly, the possible complex reflectivity of the medium. In the past, astonishing progress has been made in the field of optimization [7, 17, 3]; however, complex misfit functions still pose complex challenges. The new forward modeling techniques mentioned in the last section are tailored to model the wave field in an efficient way. The saved computing time and resources can be used to employ a more involved inversion scheme. The idea is to use benefits of global and local

optimization methods to explore the search space and to increase the probability of finding the global optimum. The Research Paper 3, “Hybrid Genetic Deflated Newton Method for Global Optimisation”, gives insight into how to find the global optimum in complex and high-dimensional search spaces. The method was patented in the United States.

An important application of optimization schemes is the investigation of source parameters of a wave source. To accomplish that, a data misfit between measured and computed wave-motion data is minimized. Discrete applications include earthquake source inversion, acoustic and atmospheric sciences, and electro-dynamics. Wave-source optimization is an extremely challenging task because of the potential of many parameters and non-convex, non-linear misfit functions. The Research Paper 5, “Hybrid Genetic Deflated Newton Method for Distributed Wave-Source Optimization”, investigates the possibility to employ the new optimization scheme to invert for parameters of wave sources.

The methods described, thus far, open up new possibilities for a more efficient imaging of the sub-surface. The proposed solver for eikonal and transport equations offers an efficient travel-time inversion considering amplitude information. The result can serve as an initial model for wave-form inversion. At first, the acoustic wave field can be modeled to achieve a more accurate initial model for the full-elastic wave-form inversion. The three mentioned kinds of inversion use the proposed hybrid inversion scheme. The Research Paper 6, “Combining new Methods for Wave-Motion Modeling and Function Optimization to Improve upon Exiting Wave-Imaging Methods”, discusses how to use the developed methods to make the most of the data and obtain accurate information about the sub-surface in minimal time.

1.4 An Overview of the Research Papers

As outlined above, the full wave-form inversion suffers from several problems, mainly connected to computational costs and accuracy of the output image. All research papers try to address and solve contemporary problems in wave-motion modeling, optimization, wave imaging and in related fields. Partly, this is achieved by parallelization as presented in Research Papers 1 and 2. Research Paper 3 shows how a purely theoretical reformulation of a problem can lead to advancement of wave-motion modeling. In Research Paper 4, parallelization works in cooperation with a new optimization method to achieve a fast convergence into the global optimum of a function. Research Paper 5 investigates the ability of the new optimization method to invert for parameters of a distributed wave source. Research Paper 6 shows how the developed methods, presented in this thesis, can be combined to build an efficient tool for wave-form imaging. All developed numerical and analytical methods are tailored to improve upon wave-form imaging, not only in a seismic application, but in all fields where wave propagation and/or inversion poses complex challenges.

Research Paper 1: Fast Computation of Eikonal and Transport Equations on GPU Computer Architectures

Eikonal models have been widely criticized for the low resolution and accuracy of the output image. However, they are still often favored because of the relatively simple implementation and the associated low computational costs. In Research Paper 1, a method is introduced to compute solutions of eikonal and transport equations simultaneously, and highly parallelized, on GPU computer architectures. The high level of parallelization and also accuracy is due to the employment of novel pyramid-shaped stencils,

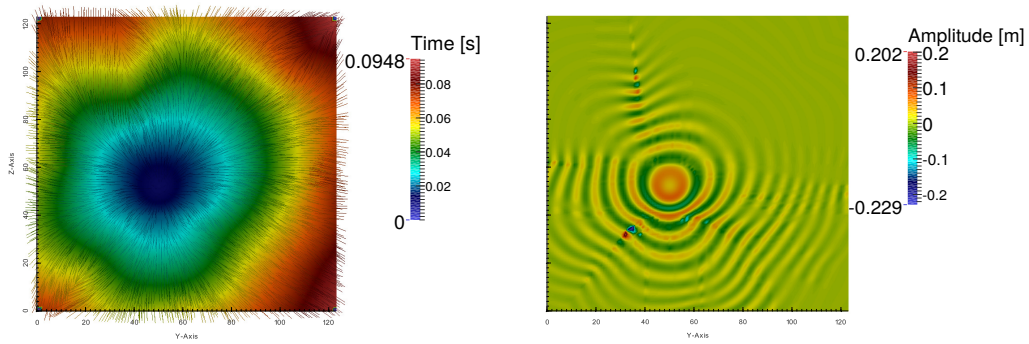


Figure 1.2: The time field (left) and the Helmholtz Green's function (right) of a wave. Ray directions are indicated as lines in the time field. Time and amplitude fields are computed simultaneously to yield the Helmholtz Green's function.

which take diagonal nodes into account. The computed amplitudes can be used to assemble the Helmholtz Green's function. The proposed method also computes all ray directions in the computational domain (see Figure 1.2). The ray directions can be used, among others, for illumination studies. This novel approach can work as a feasible alternative to full wave-form inversion or as the first step to obtain accurate initial models for more advanced computations.

Research Paper 2: A Two-Scale Method using a List of Active Sub-Domains for a Fully Parallelized Solution of Wave Equations

Wave-motion modeling is perfectly suited for parallel computer architectures. The computation of the four-dimensional wave field at a certain time step only uses values already computed from the last two time steps. Therefore, in some sense, the same principle as in Research Paper 1 is commonly used to solve the wave equation on parallel computer architectures.

The grid sizes when modeling wave motion are commonly large. They are imposed by velocities in the physical domain and frequency requirements of the solution. The size of the grid greatly exceeds the global memory of the commonly used processing units. The solution is to use a plurality of processing units and distribute the workload and data between them. This is normally done by dividing the domain into many sub-domains and allocating each sub-domain to one processing unit in a static way; static in the sense that the allocation is stable during the whole computing time.

Research Paper 2 introduces a method to dynamically allocate processing units (especially GPUs) to active sub-domains as shown in Figure 1.3. In each time step, processing units are re-assigned to active sub-domains, where active means that the sub-domain contains waves with an amplitude greater than a given threshold. A host processor decides which sub-domains need to be active in the initialization process. During the computation, the processing units decide by themselves if the assigned sub-domain contains amplitudes greater than the given threshold and consequently needs to be active. If not, the sub-domain is deactivated and the processing unit is ready to receive the next task from the host processor. The result is a method that greatly reduces the required computing resources and computing time. The method was evaluated and found to be worth protecting by a US patent.

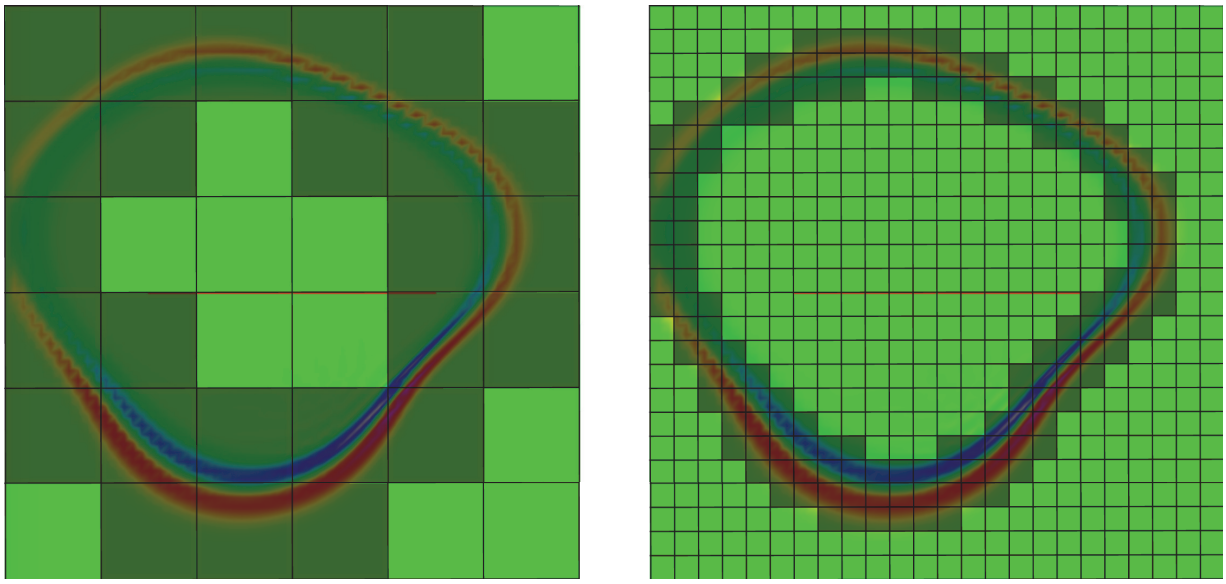


Figure 1.3: Effective problem size compared to actual problem size for two different examples. The domain is divided into 36 (labeled dark) sub-domains on the left side. The ratio of sub-domains to active sub-domains is 1.44. The right side shows the same problem with a division in 576 sub-domains. The ratio of sub-domains to active sub-domains in this case is 3.81. The example shows that the computational costs benefit from more sub-domains since active regions can be separated from non-active regions more accurately. This principle has a limit: the sub-domains need to be large enough to fully utilize the chosen processing units. Therefore, in the case of an abundant number of processing units, the number of nodes in a sub-domain, and hence, the resolution can and should be enlarged to optimally utilize all available processing units.

Research Paper 3: An Acoustic Wave Equation in a Transformed Metric Space for Various Types of Anisotropy

Acoustic wave propagation does not describe a physical phenomenon in natural anisotropic media [4]. Nevertheless, approximations of wave fields can be computed by using the acoustic assumption. For instance, to model an electric signal through heart tissue, the acoustic wave equation can be solved. The fibers in the heart show a preferred direction, thus the medium is anisotropic. The acoustic assumption is also used to approximate the elastic wave equation since the computation of the acoustic wave equation is computationally cheaper. To model the physically impossible acoustic wave propagating through anisotropic media, we have to apply some advanced strategies. When approximating elastic wave propagation, we can set the shear wave velocity to zero to model only p-waves [4, 51]. However, this method produces artifacts and only works if we are actually dealing with elastic wave propagation. In other cases, we can make use of another method.

Instead of actually using different wave speeds in different directions, we can assume that space is stretched or contracted in certain directions. Hence, we can define a dispersion relation and transform it into another metric space. From the new dispersion relation, new wave equations follow directly to model various types of anisotropy. Research Paper 3 shows how to apply the theory to a variety of anisotropies (see Figure 1.4), which can directly be used for electrical signal propagation in the heart tissue, seismic waves and many other applications.

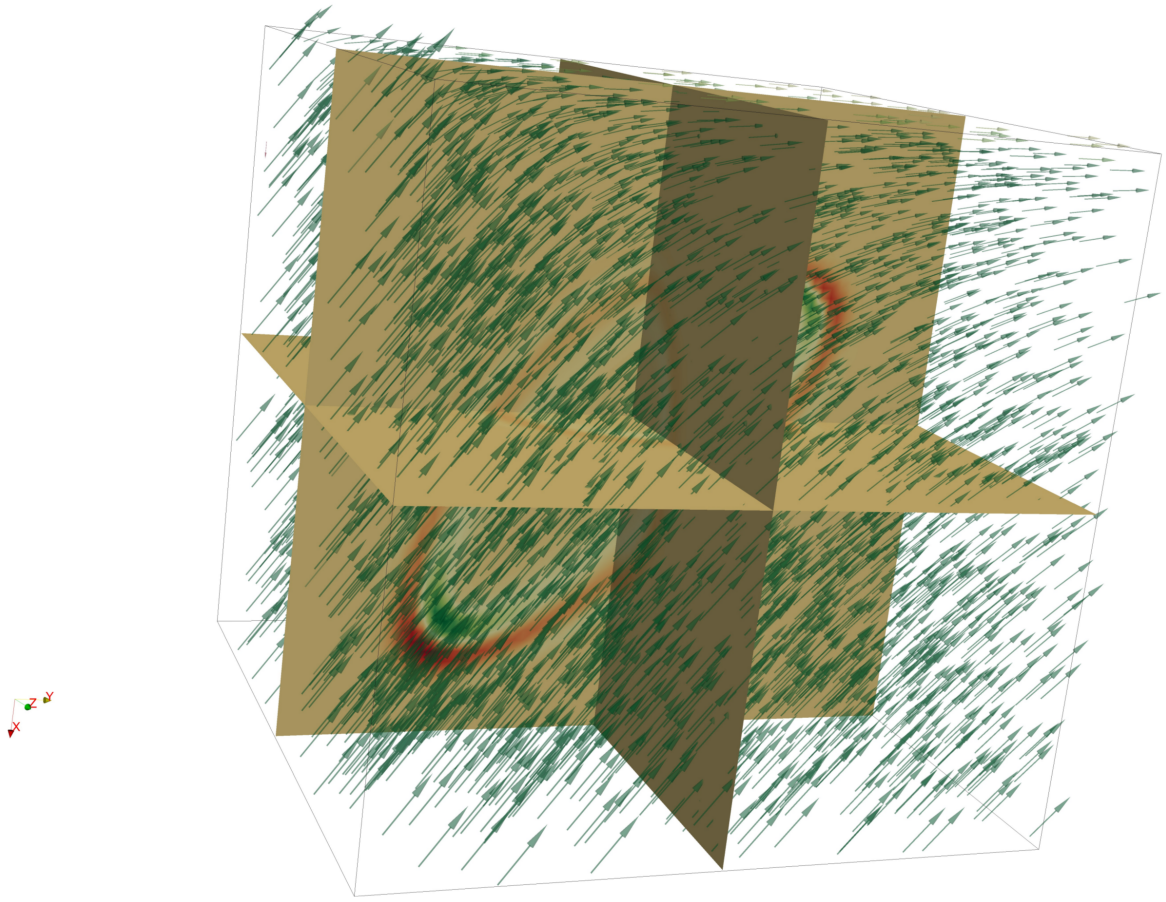


Figure 1.4: An acoustic wave propagating through an anisotropic medium. The wave is following a spatially-changing, preferred direction which is indicated by green arrows.

Research Paper 4: Hybrid Genetic Deflated Newton Method for Global Optimisation

The highly non-linear parameter spaces (see Figure 1.1) linked to wave-form inversion pose a, so far, unsolved problem. Due to many local optima, finding the global optimum turns out to be a complex challenge. The costly forward modeling of a wave makes the employment of a purely global search algorithm unfeasible since it needs many function evaluations to converge. Local optimization methods, on the other hand, struggle to find the global optimum.

The previously proposed methods to model wave phenomena are able to speed up wave-motion modeling significantly and save computing resources. The saved computing time and resources can be used to employ more sophisticated inversion methods. Global and local optimization procedures can be combined to make the most of the method's benefits.

Research Paper 4 introduces a tailored method for optimization in non-convex, non-linear parameter spaces. The main idea is to use a hybrid of global and local optimization methods combined with deflation. In the first step, several individuals are placed in the parameter space. The notion of using individuals is inherited from the genetic algorithm, where individuals are entities that are subject to rules motivated by actual biological processes. In this case, an individual is a living organism that adapts and procreates. The number of used individuals depends on the demand for accuracy and available computing resources

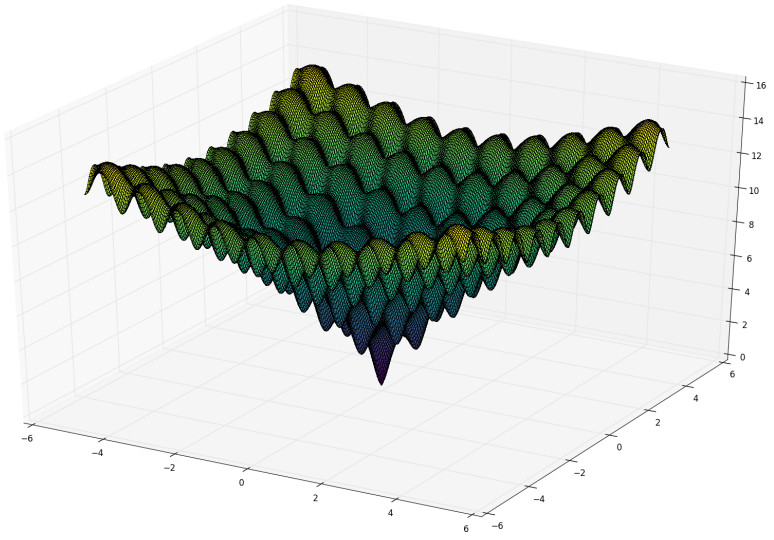


Figure 1.5: The figure shows a typical objective function as encountered in many applications. Especially in the field of seismic imaging, the objective function can be non-linear, non-convex and periodic due to the oscillatory nature of the wave field and the complex reflectivity of the medium.

and time. The placed individuals all perform a local search algorithm in parallel. Subsequently, the locations of the identified optima are stored, and the local search algorithm starts again from the initial position. To avoid that individuals propagate into the same optimum, deflations are added to the objective function where local optima were found. When the individuals struggle to find a new optimum, they can procreate. The new generation can then search for more optima in other regions of the search space. The proposed method (referred to as HGDN) makes it possible to get a notion of the complexity and shape of a function. It is therefore possible to get an understanding of the accuracy and resolution of the solution of the inversion. The proposed method was successfully tested on functions like the one shown in Figure 1.5. The HGDN method was found to be worth protecting by a US patent.

Research Paper 5: Hybrid Genetic Deflated Newton Method for Distributed Wave-Source Optimization

The hybrid genetic deflated Newton method (HGDN) has shown to be designed to optimize complex functions efficiently. A problem that needs optimization of very complex misfit functionals is wave-source optimization. Research Paper 5 investigates wave-source optimization for the acoustic wave equation. Since the HGDN method has a local component, the adjoint equations had to be derived to obtain first and second derivatives. As commonly known, the first derivatives equal the adjoint field evaluated backwards in time as illustrated in Figure 1.6. The results showed that the proposed method was able to find many optima, which almost equally well explained the measured data. Therefore, it is shown that a common Newton scheme for wave-form inversions cannot lead to success. The proposed method can

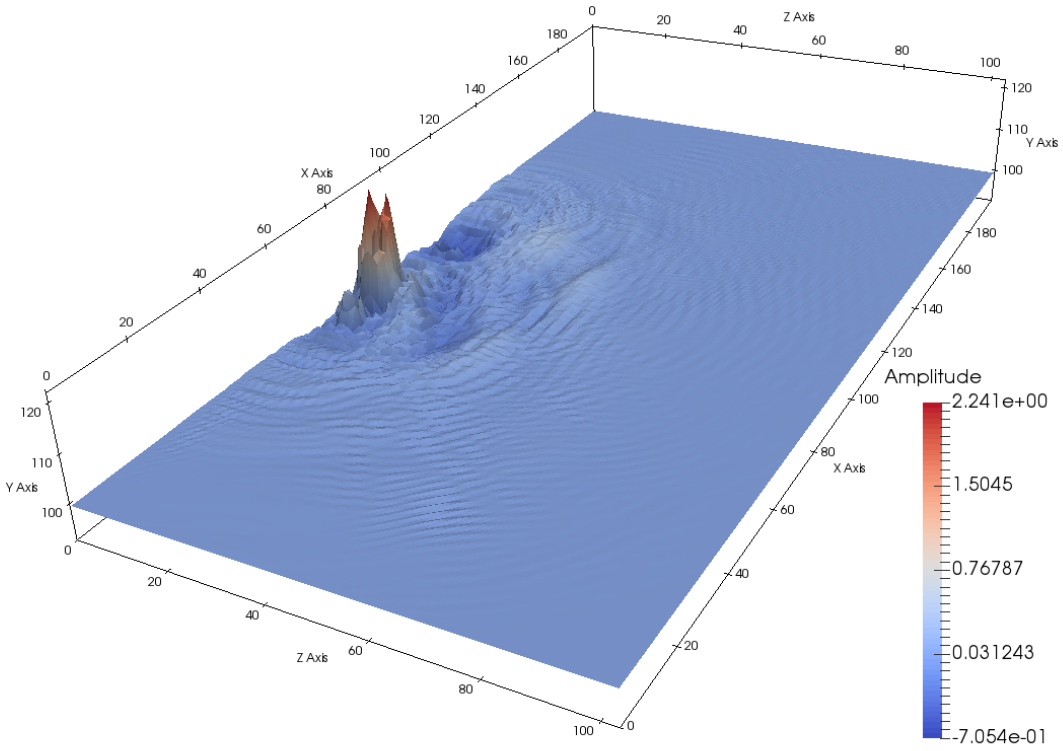


Figure 1.6: The adjoint field along a two-dimensional distributed wave source. The figure clearly shows where the magnitude of the initial source was over or under estimated. Large function values of the adjoint field mean large gradients for the respective region.

be game-changing in acoustic or atmospheric sciences and can be extended for elastic wave propagation and dynamic faults. The results can lead to a comprehensive understanding of geological fault lines as earthquake sources.

Research Paper 6: Combining new Methods for Wave-Motion Modeling and Function Optimization to Improve upon Exiting Wave-Imaging Methods

Research Paper 6 presents an attempt to combine all methods introduced in the previous research papers to build a new tool for efficient wave-imaging in a seismic context. Firstly, a travel-time inversion is proposed using the method introduced in Research Paper 1 to limit the search space and to obtain an accurate initial model for the wave-form imaging step. The method of Research Paper 3 can be used to include anisotropy. The method of Research Paper 4 (HGDN) is proposed to perform the inversion step. Next, the new algorithms for wave propagation introduced in Research Paper 2 are used together with the HGDN method to perform an acoustic and/or elastic wave-form inversion to further limit the parameter space or find the final solution.

The proposed approach could lead to more accurate images of the sub-surface. An improvement in the field of wave imaging could have great benefits in research and industry. Especially in the medical field and in seismic exploration, the gained information that stems from using an advanced optimization scheme

could potentially result in influential knowledge. It is, for instance, imaginable that a misfit function in the medical field does not depend on the exact location, in one or more directions, of the detected object. Or, several images explain the data equally well. The same holds for oil and gas exploration, where it is not unlikely that many models explain the measured data. It is, without a doubt, an advantage to obtain comprehensive knowledge of the misfit function before spending millions of dollars for drilling or performing surgery on a high-risk patient.

Organization of the Thesis

The research papers are discussed in logical order, which roughly coincides with the chronological order of my projects. Every chapter contains an introduction explaining the transition between research papers. In order to improve the quality of the thesis, some minor typographical errors, present in the published versions of the papers, were corrected.

1.5 Other Contributions

In addition to the research papers, several other contributions have been made to the current state of research based on the expertise acquired during the PhD period. To limit the extent of this thesis, the documents are not included here, but are referred to in this section.

1.5.1 Talks

- EAGE Conference and Exhibition, 2014, Amsterdam: M. Noack, and S. Clark. "Parallel and simultaneous computation of eikonal and transport equations by taking full advantage of GPU computer architecture"

- Workshop Programming of Heterogeneous Systems in Physics 2014, 2014, Jena: M. Noack, and S. Clark "Fast and accurate solutions of the eikonal and transport equations"

1.5.2 Posters

- Cardiac Physiome Workshop, 2015, Auckland: K. S. McLeod, S. Wall and M. Noack "Fast Sweeping vs. Fast Marching for Eikonal Methods of Electrophysiology – A Potential for Significantly More Efficient Computation?"

- SCEC Meeting, 2016, Palms Springs: M. Noack and S. Day "Hybrid Genetic Deflated Newton Method for Distributed-Source Optimization"

1.5.3 Book Chapter

- K. McLeod , M. Noack, J.Saberniak, K. Haugaa, 2015, "Structural Abnormality Detection of ARVC Patients via Localised Distance-to-Average Mapping" in
  "Statistical Atlases and Computational Models of the Heart - Imaging and Modelling Challenges"

1.5.4 Patents

- Marcus M. Noack, “A Two-Scale Method using a List of Active Sub-Domains for a Fully Parallelized Solution of Wave Equations”, Number: 471229US124, Year: 2016
- Marcus M. Noack and Simon W. Funke, “Apparatus and Method for Global Optimization”, Number: 473385US124, Year: 2016

Chapter 2

Eikonal and Transport Equations

Article published in *Geophysics*, September 2015
DOI: 10.1190/geo2014-0556.1

The solution of the wave equation for planar waves can be inserted into the wave equation to obtain an approximation by three terms. Because of the exponents of the frequency terms, all three terms have to equal zero for high frequencies. From this approach, the eikonal equation, the transport equation and a rest term, which is commonly neglected, can be derived. The eikonal equation should not only be seen as an approximation of the wave equation, but as a construct, which connects two fundamental principles, namely the Fermat's and the Huygen's principles. In 1678, Huygens proposed that each point a wave reaches serves as a secondary source; a principle that would later be improved upon by Fresnel to incorporate diffraction effects. The Fermat's principle, discovered by the French mathematician Pierre de Fermat, states that a ray of a wave will choose the path of minimal time; a formulation, which casts light on the impact of optimization on natural processes. The eikonal equation can be used to describe many processes in nature that resemble a moving front. If this front carries some sort of energy or quantity, the transport equation can be used to model and predict the amount of this energy or quantity at a certain point. The eikonal and transport equations have a wide variety of applications like exploration seismology [21, 22], electrophysiology [41], computer vision [10], wildfire modeling [30] and particle physics [11]. It is therefore essential to develop efficient solvers for eikonal and transport equations.

Both equations are well suited for highly parallelized finite-difference computations as shown in the next research paper.

Fast Computation of Eikonal and Transport Equations on GPU Computer Architectures

Marcus M. Noack[1,2,3] and Tor Gillberg[1,2,4]

[1]Kalkulo AS, P.O.Box 134, 1325 Lysaker, Norway
[2]Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
[3]Department of Informatics, University of Oslo, Gaustadalleen 23 B, 0373 Oslo, Norway
[4]Bank of America Merrill Lynch, 2 King Edward Street, London, United Kingdom

Abstract

Eikonal models have been widely used for travel-time computations in the field of seismic imaging, but are often criticized for having low accuracy and poor resolution of the output image. Including amplitude information can provide higher model resolution and accuracy of the images. A new approach for computing eikonal travel times and amplitudes is presented, and implemented, for multi-core CPU and GPU computer architectures. Travel times and amplitudes are computed simultaneously in iterations of the three-dimensional velocity model. This is achieved by using upwind travel-time information in a recently introduced fast-sweeping method, and computing amplitudes directly after the travel times. By performing the extra computations simultaneously with the travel times, the additional cost for the amplitude and ray paths is low. The proposed method was tested on synthetic 3D data sets to compute travel times, amplitudes and ray paths, from which the Helmholtz Green's function was assembled. Using a grid of 124^3 nodes, the computations were performed in less than one second. The proposed method could work as a feasible alternative to full wave-form modeling in seismic applications, which suffer from demanding computations, since it requires several order of magnitudes shorter computing times.

1 Introduction

Wave propagation phenomena play a central role in many fields of physics, environmental research and medical imaging. Therefore, the fast and accurate numerical modeling of these phenomena is an important task. In the field of geophysics, the ability of an accurate and fast wave modeling method can result in higher resolved images of the subsurface in a shorter computing time. Therefore, allowing for lower costs and higher confidence for oil and gas exploration, and new possibilities in geological research. Furthermore, a faster and more accurate wave modeling method can improve risk assessment methodologies for earthquakes and volcanoes.

Computing the direct solution of the wave equation is a difficult and time consuming process [12]. Non-reflecting boundary conditions and imposed time stepping further increase the required computational time.

Taking the Helmholtz equation as the static part of the wave equation is a frequently used simplification [6]. However, using the Helmholtz equation does not sufficiently decrease the computational time and the solution process remains complicated [6]. A widely used approach to avoid these drawbacks is to use the Helmholtz Green's function for a point source [11]. First arrival travel times and their amplitudes are needed in order to estimate the Helmholtz Green's function. These travel times and amplitudes are computed as solutions of the eikonal and transport equations, respectively. In this paper, we introduce methods that solve both equations fast and accurately.

2 Background

Several finite difference schemes have been introduced in the past years to solve the eikonal equation. Vidale [19] proposed the expanding square method in two dimensions and extended it later for applications in three dimensions with the expanding box method [20]. In this method, travel times are updated on the surface of a box around the source. When all nodes in this structure are updated the adjacent nodes are used to build a larger box. The expanding box method is still widely used but sometimes suffers in accuracy for complex velocity models due to the fact that the box shape does not in general resemble a wave front [14].

The fast marching method [16] tracks a general wave front as it moves through the domain. The fast marching method is more stable, accurate and faster than the expanding box method. To track a general wave shape, nodes are divided into three groups: alive, close and far [14]. All nodes in the wave front, called close nodes, are sorted depending on their travel times. The close node with the smallest travel time is re-labeled as alive, and thereby evolving the position of the front. Travel times of nodes in the vicinity of the re-labeled node that are not alive are recomputed and labeled as close. The procedure continues until all nodes are alive. The method handles sharp velocity contrasts well and the accuracy is sufficient for most applications [16]. The computational costs are of order $N \log N$, where the $\log N$ term is due to the ordering. Therefore, the feasibility of on-the-fly modeling decreases with data sets of increasing size. Even though there have been some attempts of parallelization [7], the method remains sequential by nature since the front passes only one node at a time. This is a drawback because processing units are becoming cheaper rather than faster [17].

The fast-sweeping method [24] sweeps through the three dimensional domain in eight alternating directions. When sweeping through the grid in one direction, travel times are computed for wave fronts traveling through the grid in that direction. The fast-sweeping method is an iterative method that sweeps through the domain until the solution converges. The method has a computational cost of order $O(N)$ and can be parallelized to some degree, which makes it appropriate for large data sets [25]. Using the fast-sweeping method for the amplitudes amounts to sweeping separately for the travel times and amplitudes until convergence [11].

A recently developed method, called the three dimensional parallel marching method (3DPMM) [4] is a parallel sweeping method, extending the parallel marching method to three dimensions [22]. In 3DPMM, a stencil shaped like a tetrahedron is used instead of the standard Godunov "box" stencil. The sweeps in 3DPMM are in different directions compared with traditional FSM sweeps. The result is a significant gain in parallelization opportunity compared with the traditional fast-sweeping method. On a grid with N^3 nodes, N^2 nodes can be computed in parallel. If the parallel computing opportunities are employed, the computational time can be decreased by several orders of magnitude. By making use of diagonal stencils,

the accuracy increases compared to the "box-stencils" [3], independent of the used algorithm. Therefore, the 3DPMM method offers a very fast and accurate solution of the eikonal equation.

Some methods try to combine aspects of both, sweeping and front tracking algorithms. This marriage of algorithms creates very efficient, but also more complicated algorithms. These algorithms allow for parallel implementations. The fast iterative method by Jeong and Whitaker [8] computes nodes in a list in parallel. The SOLAS method by Gillberg et al. [5] uses the 3DPMM method on selected parts of the full domain. Both methods are very efficient solvers, but the implementations are also more involved. We expect the methodology introduced in this paper to be applicable to algorithms other than pure 3DPMM.

3 A New Method for Travel-Time and Amplitude Computations

We propose a solver for the Helmholtz Green's function that takes full advantage of parallel computer architectures by extending the 3DPMM method. The main idea is to solve the transport equation simultaneously with the eikonal equation. Our key observation is that for any given node, the upwind amplitude data is in the vicinity of the upwind travel-time data. The data used to compute new travel times can be re-used immediately to compute amplitudes. In our novel approach, when the travel time of one node is updated we immediately estimate the amplitude for the same node. Therefore, the amplitude and the time planes sweep simultaneously through the domain. Since we only use data from the last update planes, the computations of all nodes in the current update plane are independent. In this way, entire planes can be updated in parallel. In order to estimate new travel times, the entry location of the ray yielding the new travel time is computed directly. The result is a method that calculates first-arrival travel times, amplitudes and all rays simultaneously and efficiently on multiprocessor architectures.

The remainder of the paper is organized as follows. The theory section gives an overview of the basic methods and the main principles of the algorithm, including a summary of the 3DPMM method and a section detailing the computation of ray paths and entrance locations. This is followed by a section describing two stencils for the amplitude estimation. The proposed method was applied to three synthetic examples to show the functionality of the novel solver for the Helmholtz Green's function as described in the results section.

4 Theory

The three-dimensional parallel marching method (3DPMM) was proposed by Gillberg et al. [4] and used to model geological folds. A folded layer is modeled as iso-surface of the solution $T(\mathbf{x})$ to the following static Hamilton-Jacobi equation

$$\begin{aligned} F\|\nabla T(\mathbf{x})\| + \psi\langle \mathbf{a}, \nabla T(\mathbf{x})\rangle &= 1 \\ T(\mathbf{x}) &= 0 \text{ on } \Gamma. \end{aligned} \tag{2.1}$$

In the above equation Γ is a reference horizon, $\|\cdot\|$ is the Euclidean norm, $\langle\cdot,\cdot\rangle$ is the inner product of the Euclidean space, F and ψ are scalars and $\mathbf{a}$ is a unit vector pointing in the direction of the symmetry line of the fold. The eikonal equation, commonly used to estimate first-arrival travel times of seismic waves [1] is, formulated as

$$\begin{aligned} \|\nabla T(\mathbf{x})\| &= s(\mathbf{x}), \\ T(\mathbf{x}) &= \Psi(\mathbf{x}) \; \forall \; \mathbf{x} \in \Omega, \end{aligned} \tag{2.2}$$

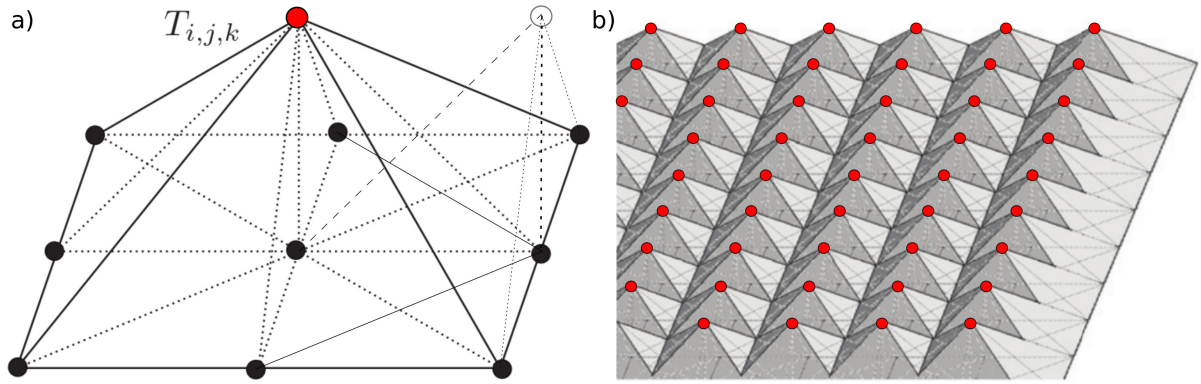


Figure 2.1: a) The pyramid shaped stencil used in 3DPMM. Notice the inclusion of the diagonal nodes. The node represented as an empty circle shows the position of a neighbor of the update node and is not used for the computation of T_{ijk}. b) A layer of independent, pyramid shaped stencils. The entire layer of stencils can be computed in parallel. Image modified from Gillberg et al. [4].

where $s(\mathbf{x})$ is the slowness field and $\Psi(\mathbf{x})$ are initial values given on the set Ω. The isotropic eikonal equation is a special case of the static Hamilton-Jacobi equation. The 3DPMM method can therefore be used unmodified to estimate solutions to the eikonal equation. For the description of 3DPMM, consider a regular box grid with nodal values T_{ijk} being approximations of first-arrival travel times. We assume that some nodes are given initially, and all other nodes are set to infinity. Since we are solving for first arrivals, we solve for the minimum obtained value of T. Nodes are updated with a "pyramid" shaped stencil [4], as illustrated in Figure 2.1a, in which the top node T_{ijk} is the one being updated. The stencil contains eight tetrahedron three-point stencils, 16 two-node stencils, and nine one-node stencils, as further detailed in Gillberg [2]. The smallest estimation for the update node of these 33 stencils is used according to Huygen's principle (for an explanation of the systematic application of Huygen's principle in the finite difference approximation see [13]). Appropriate upwind conditions can be used to decrease this number and increase performance. Entire planar surfaces of nodes can be computed in parallel since nodes on the same surface do not depend on each other, as shown in Figure 2.1b. The 3DPMM sweeps through the domain in axial directions, by shifting the planar surface of nodes in the direction of the "pyramid" tops. If a new estimate t_{new} is smaller than the previous estimate t_{old}, the node will receive a new estimate. In our novel method, amplitude estimates are computed for a node after the travel times only if $t_{new} \leq t_{old}$. This condition ensures that amplitude and time data in the upwind direction are used for the amplitude computation. The transport equation for computing the amplitude $A(\mathbf{x})$ is as follows [18]

$$\langle \nabla T(\mathbf{x}), \nabla A(\mathbf{x}) \rangle + \frac{1}{2} A(\mathbf{x}) \nabla^2 T(\mathbf{x}) = 0. \tag{2.3}$$

The gradient of the wave front, $\nabla T(\mathbf{x})$, is estimated during the travel-time computation. Given the gradient, the entrance location $\mathbf{x}_E = (x_E, y_E)$ of the ray into the stencil bottom layer can be calculated. Knowing the location where the ray enters the stencil, we can interpolate for the travel-time [23] and the amplitude values at this point. Let A_E, T_E be the amplitude and travel-time values at the location where the ray enters the stencil element, and A_N, T_N the corresponding new values of the node being updated. Using the transformed transport equation we can approximate A_N by

$$A_N = \frac{A_E(T_N - T_E)}{T_N - T_E + \frac{1}{2}||\mathbf{x}||^2 \nabla^2 T}, \tag{2.4}$$

where $||\mathbf{x}||$ is the Euclidean distance between node N and the location E. The additional work required to compute these values is small since most travel-time data has recently been used in computations and the gradient of the wave front is estimated in the travel-time computation. The accuracy of the method highly depends on the accuracy of the upwind travel-time field. It is necessary to avoid inaccurate travel times around the source by using special, very accurate techniques such as refined meshes, wave front construction methods [21, 15] or ray tracing methods [9, 15] to compute the travel times around the source. The estimation of the Laplacian of the travel-time field is the crucial and most difficult part of the calculation. Using a traditional three-point stencil for the second derivatives is often not possible since not all needed values are necessarily already assigned values. Two different ways to solve this problem are described in the following sections after characteristic curves for the eikonal equation are discussed. After obtaining the travel-time and the amplitude field we want to estimate the Green's function $G(\mathbf{x}, \omega)$ for the Helmholtz equation. The Helmholtz Green's function as presented in [11], is formulated as

$$\nabla^2 G(\mathbf{x}, \omega) + \frac{\omega^2}{v(\mathbf{x})^2} G(\mathbf{x}, \omega) = -\delta(\mathbf{x} - \mathbf{x_0}), \tag{2.5}$$

where $\delta(\mathbf{x} - \mathbf{x_0})$ is the Dirac delta function at the position $\mathbf{x_0}$, ω is the frequency and $v(\mathbf{x})$ is the wave velocity at $\mathbf{x}$. The solution in three dimensions is given by [10]

$$G(\mathbf{x}, \omega) = A(\mathbf{x}) e^{i\omega T(\mathbf{x})}, \tag{2.6}$$

where $A(\mathbf{x})$ is the amplitude field and $T(\mathbf{x})$ is the travel-time field. The Helmholtz Green's function can be used directly for estimating the amplitude field for different frequencies.

5 Computing Ray Paths and Entrance Locations

The characteristic curves of the eikonal equation are often referred to as rays in seismology. Characteristic curves of the isotropic eikonal equation show the fastest path between two points given the velocity [18]. Characteristic curves are often defined in terms of the Hamiltonian, H, of the equation. For the eikonal equation (2.2), the Hamiltonian is defined as $H(\mathbf{x}, \mathbf{p}) = \frac{1}{s(\mathbf{x})}|\mathbf{p}| - 1$, where $\mathbf{p} = \nabla T(\mathbf{x})$. Let $\mathbf{x}(s)$ be a parameterization of a characteristic curve, $\mathbf{x}(s) = (x(s), y(s), z(s))^T$, where s is the parameterization parameter. If the Hamiltonian H is convex in $\mathbf{p}$, the ray path can be found from the following relation,

$$\frac{d\mathbf{x}}{ds} = \nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}), \tag{2.7}$$

as defined in [18]. After differentiating H with respect to $\mathbf{p}$ and integration with respect to s we obtain the characteristic curve for the eikonal equation as

$$\mathbf{x}(s) = s\, v(\mathbf{x}) \frac{\nabla T(\mathbf{x})}{||\nabla T(\mathbf{x})||}, \tag{2.8}$$

$$\mathbf{x}(0) = \mathbf{x_0}, \tag{2.9}$$

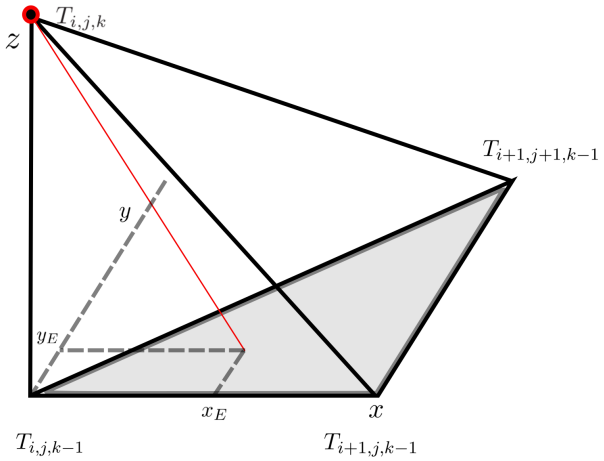


Figure 2.2: Illustration of the entrance location (x_E, y_E) at the bottom of the tetrahedron stencil.

where $||\nabla T(\mathbf{x})|| = \sqrt{\left(\frac{\partial T(\mathbf{x})}{\partial x}\right)^2 + \left(\frac{\partial T(\mathbf{x})}{\partial y}\right)^2 + \left(\frac{\partial T(\mathbf{x})}{\partial z}\right)^2}$, and $\mathbf{x_0}$ is the starting point of the curve. Since the pyramid base is a distance dz from the node being updated, the entrance location of the ray follows as

$$x_E = -dz \frac{\frac{\partial T(\mathbf{x})}{\partial x}}{\frac{\partial T(\mathbf{x})}{\partial z}} \tag{2.10}$$

$$y_E = -dz \frac{\frac{\partial T(\mathbf{x})}{\partial y}}{\frac{\partial T(\mathbf{x})}{\partial z}}. \tag{2.11}$$

The entrance location (see Figure 2.2) ensures that only rays traveling through the bottom of the stencil are used to estimate a new time value. If the entrance locations are outside the stencil boundaries, the shortest path along the boundary is used to compute the new amplitude estimate. In order to compute the time and amplitude values at the entrance point of the ray into the stencil, we use linear interpolation in case of a two-node stencil and bilinear interpolation in case of a three-node stencil.

5.1 Estimating the Laplacian for Slowly Changing Velocity Fields

The first method of estimating the Laplacian uses a wide area upwind of the stencil and is therefore applicable only to slowly changing velocity fields. Only one assigned node in the stencil configuration in Figure 2.1a is sufficient for creating a new time estimation. The amplitude calculation needs at least seven nodes with travel-time values to estimate the Laplacian. We therefore propose to use two more stencil planes in the upwind direction, as shown in Figure 2.3a. Already estimated nodes in a 147-point cuboid behind the stencil are used to approximate the necessary second derivatives for the Laplacian. The pseudo code for the algorithm using this stencil extension for the estimation of the Laplacian can be found in Algorithm 1. Provided a sufficiently large source with respect to the spatial dimensions, this stencil extension manages to create an amplitude estimate even during the first sub sweep, when most of the nodes within the stencil do not yet have a non-infinity time estimate (see Figure 2.3b). The method lacks in accuracy since the estimation of the Laplacian is not sufficiently local when a large set of nodes is used. The fraction of nodes not being upwind of the update node can be high, especially for fast varying velocity fields, which leads to poor accuracy.

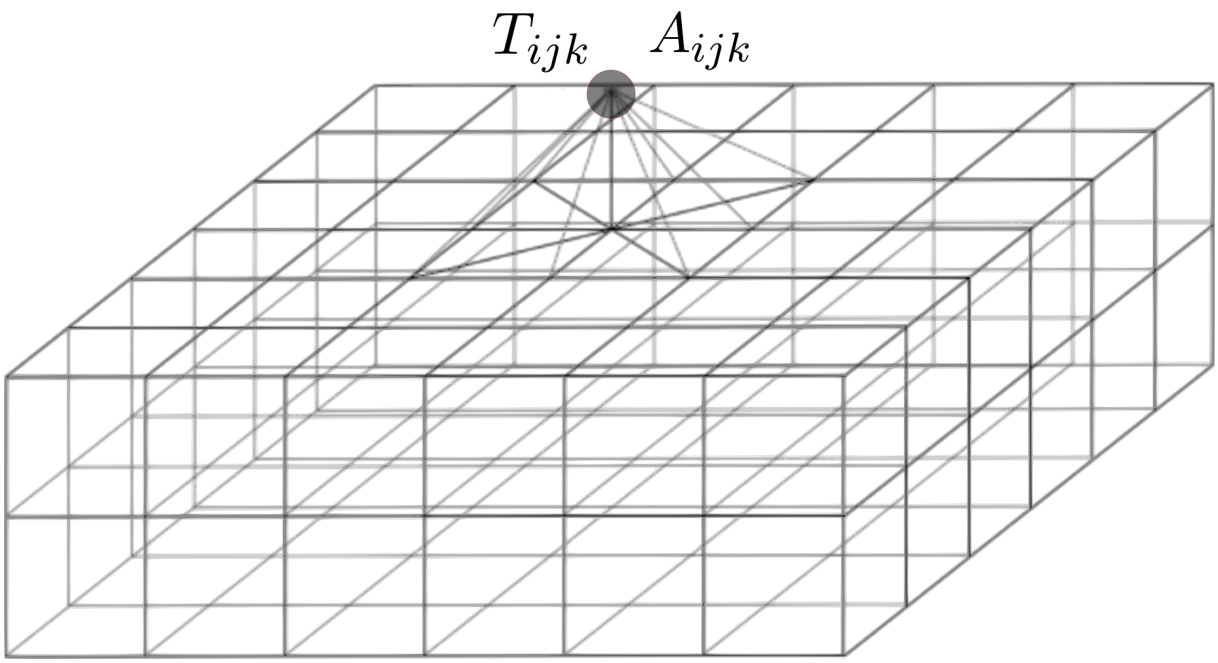


Figure 2.3: The nine-point pyramid stencil from Figure 2.1a and the 147 points used to estimate the second derivatives of the time field. T_{ijk} and A_{ijk} are the values of the node being updated.

Algorithm 1: Pseudo code for a sub sweep in the positive x-direction (index i) for a slowly changing velocity field. The indexing is adapted to the size of the stencil and to ensure that all nodes in the stencil are inside the domain.

```
for i=4,...,n_x do
    for j=4,...,n_y-3 do
        for k=4,...,n_z-3 do
            for all of the eight tetrahedron stencils do
                Compute ∇T(x);
                Estimate entrance location x_E = (x_E, y_E);
                Compute t_new using values on the tetrahedron base;
            T_ijk = min(8 time estimates);
            if T_ijk <= t_old then
                T_ijk = t_new;
                estimate ∇²T(x) using
                  T_{i-a,j±b,k}, a ∈ {0,1,2,3}, b ∈ {0,1,2,3}, c ∈ {0,1,2,3};
                Create new amplitude estimate, a_new;
                A_ijk = a_new;
            Select angle of incidence;
            Update the previous estimates;
```

5.2 A Local Approach for the Estimation of the Laplacian

Provided a sufficiently large source, the method introduced in the previous section results in a new amplitude estimate for all numerical experiments we tested. This reduces the number of iterations needed for convergence, but the computational cost of performing an update is high. Two key points must be addressed: firstly, the number of sweeps can only be reduced by one with the stencil described in the previous section. Since the fast-sweeping approach is an iterative method and therefore needs many

sweeps to converge, a reduction of the needed sweeps by one is only a minor improvement. Secondly, the computational cost of finding the Laplacian in a 147-point cuboid is high considering that it has to be done for every node in the grid.

With these considerations in mind, we design a method that uses data from a smaller area behind the stencil. The Laplacian is estimated faster and more accurately at the expense of an increased number of sweeps needed for convergence. The idea is to create a method that does not always create an amplitude estimate to nodes where the time value has been updated. Rather, the algorithm assigns an amplitude value if it is possible to calculate a Laplacian using local data. Different ways of estimating the Laplacian are possible. It may be tempting to directly use the seven-point stencil (see Figure 2.4) behind the pyramid stencil for the calculation of the Laplacian; however the accuracy is poor in this case. The key to higher accuracy is to vary the position of three-point stencils for the second derivatives, depending on the entry of the ray into the pyramid base. The three-point stencils for the second derivatives that are closest to the ray path are used for the estimation of the Laplacian. In cases where the rays are outside the stencils, the shortest path along the stencil boundary is used (see Figure 2.4). Amplitudes and travel times have to converge independently, therefore at least three sweeps are necessary. This is a drawback that is rendered obsolete for non-homogeneous velocity fields. The algorithm using the local estimation of the Laplacian is presented in Algorithm 2.

Due to the special stencil shape, entire 2-D planes can be updated in parallel. The simultaneous computation of amplitudes and travel times and the abundant parallelization make the method very efficient.

Algorithm 2: Pseudo code for a sub sweep in the positive x-direction (index i) for the local approach. The indexing is adapted to the size of the stencil and to ensure that all nodes in the stencil are inside the domain.

```
for i=3,...,n_x do
    for j=3,...,n_y-2 do
        for k=3,...,n_z-2 do
            for all of the eight tetrahedron stencils do
                calculate ∇T(x);
                compute T_ijk using T_{i-1,j±a,k±b}, a ∈ {0,1}, b ∈ {0,1};
            select t_new = min(8 time estimates), update T_ijk;
            calculate corresponding entrance location x_E = (x_E, y_E);
            calculate ∇²T(x) using T_{i-a,j±b,k±c}, a ∈ {0,1,2}, b ∈ {0,1,2}, c ∈ {0,1,2};
            compute A_ijk using A_{i-1,j±a,k±b}, a ∈ {0,1}, b ∈ {0,1};
            if Amplitude computation possible and t_new <= t_old then
                select a_new;
            else if Amplitude computation possible and t_new > t_old and amplitude is unassigned
              then
                select a_new;
            select the corresponding angle of incidence angle, ∇T(x);
```

6 Implementation Details

The current implementation, using the local approach for the estimation of the Laplacian was written in C++ and was compiled with the CUDA code compiler nvcc. Because of the special stencil shape, some

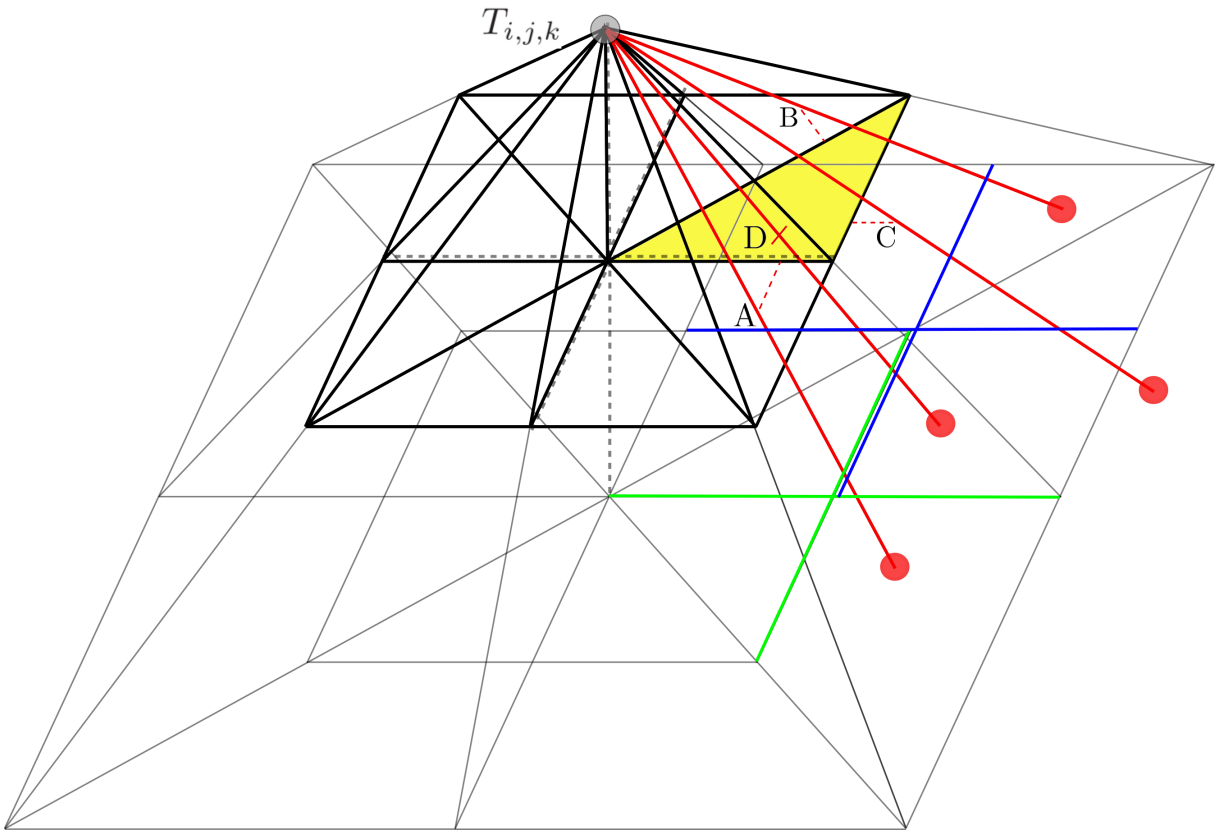


Figure 2.4: A local stencil for the simultaneous computation of travel times and amplitudes. The computation of the Laplacian is only possible when enough nodes are already assigned. The positions of the three-point stencils for the second derivatives depend on the ray direction (blue and green). This is more accurate than the use of the seven-point stencil behind the pyramid stencil (dashed). For rays A, B and C, the shortest path along the boundary of the stencil element, and linear interpolation is used to obtain the travel time and amplitude at the entrance location of the ray into the tetrahedron. For ray direction D, bi-linear interpolation is used to obtain the travel time and amplitude at the entrance location of the ray.

of the T-nodes were placed in the shared memory. Shared memory is a user-defined cache memory on GPUs. Since the computation of one node is quite demanding, the use of CUDA shared memory only leads to a speed up of around 10% compared to the use of global memory. Our code was also parallelized for multi-core CPU structures using OpenMP. The code was optimized with full optimization instructions using the -O3 flag. The numerical experiments were performed on two different machines.

The first machine is equipped with 32GB of DDR3 ram, a four core Intel Core i7-4800MQ CPU @ 2.70GHz CPU and a GeForce GTX 770M GPU. The second machine is a Linux server with 64 GB of DDR3 ram, an eight core dual Intel E5-2670 (2.6 GHz) with a total of 16 physical compute cores, and a Tesla K20x GPU. When computing times are presented, they are averages from 4 repeated measurements. However, the variation in computing time is negligible between different runs.

Grid	i7,1	i7,4	E5,1	E5,4	E5,8	E5,16	770M	K20x
124^3	11.18	3.84	14.39	8.40	7.20	5.91	0.90	0.71
248^3	94.49	30.83	147.42	75.44	57.09	53.13	6.21	4.49

Table 2.1: Computing time until convergence in seconds for different grid sizes and computer architectures for the first experiment. The type of processing unit and the number of used cores is indicated (processor, number of cores).

7 Results

In this section, we present computational times needed for three different velocity models. All results were computed with a spherical source of 1% of the volume of the domain, which was computed by a first order ray tracing. Different source sizes can be used depending on demanded accuracy and complexity of the velocity model. For every experiment computing times for all required sweeps until convergence are presented. Convergence was reached when no amplitude or travel-time estimate changed by more than 0.1% compared to the value after the previous sweep. For each experiment, CPU computing times for a sequential computation, as well as using multiple cores, are presented. The computational times needed when using the mentioned GPUs are also presented. Both, the CPU and GPU measurements use data stored with `float` accuracy.

The runs were performed on two different grid sizes, consisting of 124^3 or 248^3 nodes respectively. The grid size was chosen to satisfy the constraints of the specific GPU that was used for the experiments. Other grid sizes are possible as long as they do not exceed the GPU's bounded memory limitations.

7.1 Homogeneous Velocity Field

The simplest case, using a homogeneous velocity field is a special challenge for the method. This is due to straight rays traveling along the stencil boundaries. Our estimation of the Laplacian suffers slightly in accuracy in these regions, as seen in Figure 2.5. This error does not emerge when rays are bent. Convergence was reached after three iterations. Computing times are presented in Table 2.1.

7.2 Two Homogeneous Half Spaces

The method was also applied to two homogeneous half spaces with a smooth transition zone (see Figure 2.6). Once again, we see the inaccuracies in the 45 degree regions that are present due to the inaccuracy of the stencils along boundaries of the tetrahedrons. We can see a rise of the amplitude where rays collide (Figure 2.6); a natural behavior of linear systems known as superposition or, in the case of wave phenomena, interference. By reducing the width of the transition zone we can model interfaces in the domain. The algorithm also works in the case of non-continuous interfaces; however the accuracy is improved if treated as such by creating new wave fronts along an interface. Convergence was reached after three iterations. The computing times are presented in Table 2.2.

7.3 The Random Velocity Field

The third example is a random velocity field (see Figure 2.7) containing velocities in the range of 1000 to 1500 m/s. It was created by assigning uniformly distributed nodes random values in the range of 1000 to

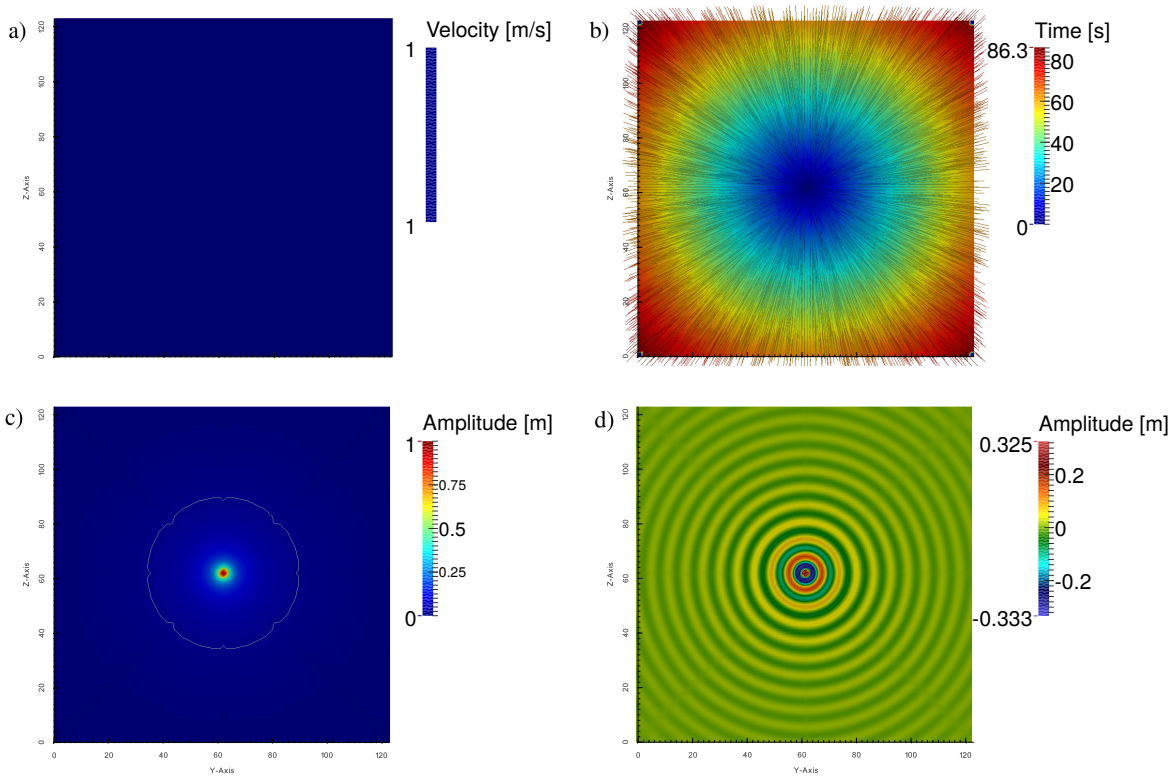


Figure 2.5: a) A homogeneous velocity model. b) The travel-time field, where isochrones are spheres and the rays are radial lines. c) The amplitude field, where amplitudes are decreasing with distance. Note that amplitudes are under-estimated in areas close to 45 degrees from the source. d) The corresponding Green's function.

Grid	i7,1	i7,4	E5,1	E5,4	E5,8	E5,16	770M	K20x
124^3	10.58	4.06	14.75	8.33	7.01	5.48	0.92	0.80
248^3	94.37	29.90	144.81	62.79	56.09	50.57	6.23	4.08

Table 2.2: Computing time until convergence in seconds for different grid sizes and computer architectures for the second experiment. The type of processing unit and the number of used cores is indicated (processor, number of cores).

1500 m/s and subsequent linear interpolation. The highest velocity gradients are 50 s^{-1}. Once again, we can see the superposition principle of waves as amplitudes rise where rays collide. In this example we see that rays avoid slow velocity areas. Notice that we have full ray coverage also in the slow velocity areas, and any "shadow zone" problems are avoided using the suggested method [18]. Convergence was reached after four iterations. The computing times are presented in Table 2.3.

7.4 Comments on Performance

The GPUs consistently outperforms the tested CPUs. This is expected since the 3DPMM allows for so many nodes[1] to be updated in parallel at any time. The CPU implementation scales well up to 4 cores, but

[1] A minimum of $124^2 = 15376$ nodes in the presented examples.

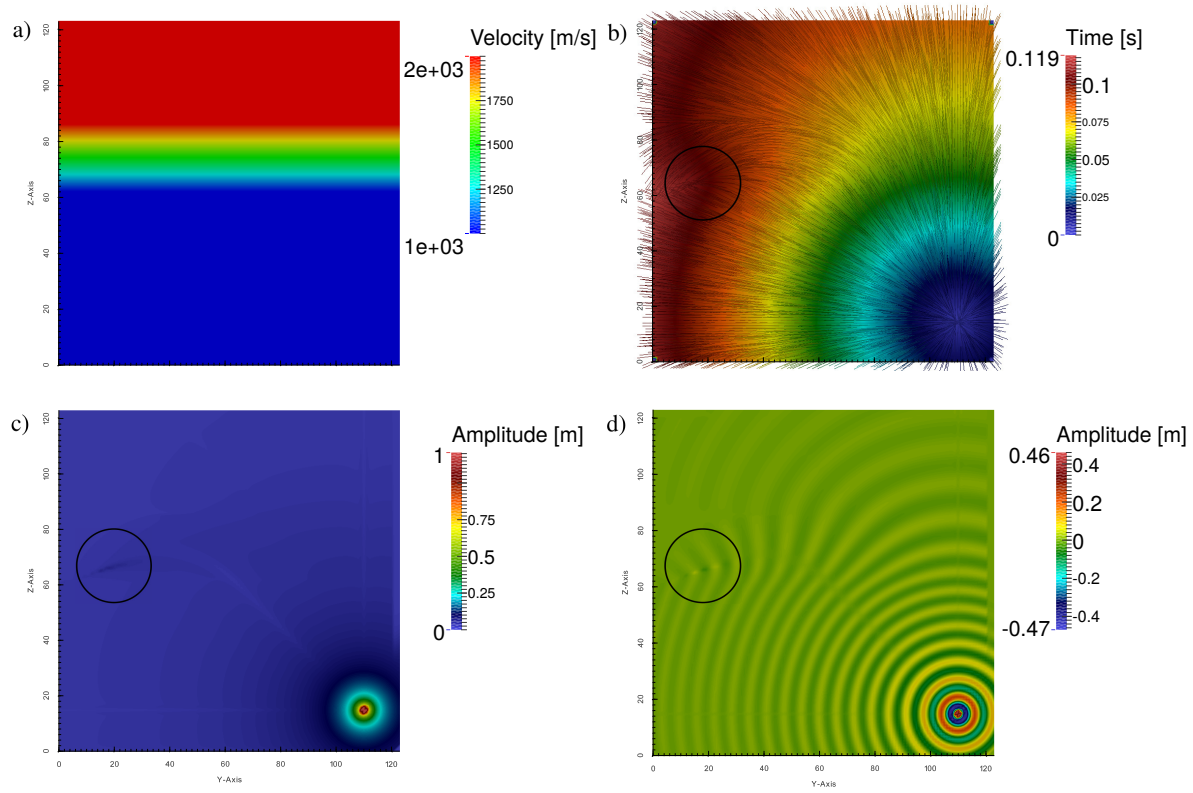


Figure 2.6: a) Two velocity half spaces with a smooth transition. b) The computed travel-time field. The wave front in the faster medium overtakes the wave front in the slower medium. The wave front builds a concave shape and eventually rays collide (black circle). c) Computed amplitude field. The area where rays collide shows an increasing amplitude as energy is rising in the same area indicated by the black circle. d) The corresponding Green's function.

Grid	i7,1	i7,4	E5,1	E5,4	E5,8	E5,16	770M	K20x
124^3	14.56	5.95	20.14	11.22	9.22	7.68	1.21	1.01
248^3	116.17	39.25	179.91	71.8	64.72	53.60	8.25	7.94

Table 2.3: Computing time until convergence in seconds for different grid sizes and computer architectures for the third experiment. The type of processing unit and the number of used cores is indicated (processor, number of cores).

flattens out thereafter. A significant amount of data is needed to update a node, which seem to decrease performance on the multi-core implementation.

8 Discussion and Perspectives

The introduced method is able to compute the Helmholtz Green's function in a stable, fast and accurate manner; however, it suffers from some restrictions. As mentioned earlier, the current implementation is built for acoustic waves in isotropic media. The adjustment for elastic waves is straightforward, though requires some modifications. Since the entrance location of the ray into the stencil is already used, the algorithm exhibits a natural ability for anisotropy. Adapting the algorithm to anisotropic problems is

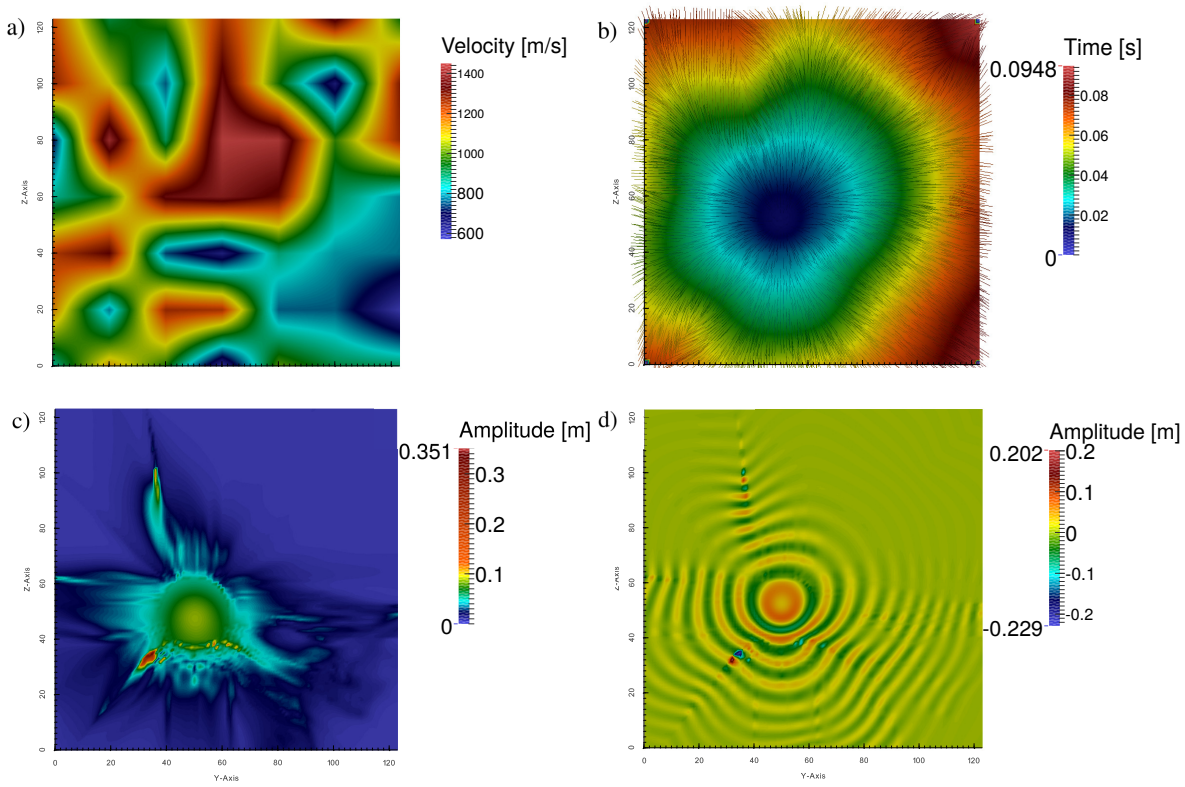


Figure 2.7: a) A random velocity field created by linear interpolation. b) In the travel-time field, rays avoid slow velocity zones and try to reach high velocity zones. c) The amplitude field shows where the linear interpolation causes roofs or valleys, and amplitudes are accordingly higher or lower. Sudden appearance of a slow velocity zone forces the wave front to a concave bending which leads to a superposition of amplitudes. d) The corresponding Green's function.

possible but not trivial since the gradient of the wave front no longer coincides with the ray path. However, ray paths can be found in the time estimation process [4] and our approach of estimating the Laplacian of the time field close to the ray origin can be used. The associated computational costs are expected to be higher in such an extension. In our current implementation, we have used first-order time stencils. An extension to second order stencils is straight-forward. In a second-order framework, the estimation of the Laplacian would be more natural. Future work will include a special treatment of interfaces in the sub-domain and a multi-GPU implementation for larger velocity models. When solving for travel times, the direction of the ray (angle of incidence) and the ray entrance into the stencil element is implicitly computed. Given the entrance location, rays can be computed by additionally solving for the shootout angle that is constant along rays. Rays allow for illumination studies and an analytical inverse step during tomography.

9 Conclusion

We introduced a methodology to compute first-arrival travel times and amplitudes simultaneously by extending the 3DPMM method. The method was implemented for multi-core CPUs and for GPUs, resulting in a fast and efficient solver for the Green's functions of the Helmholtz equation. When solving for travel times, the direction of the ray is implicitly computed giving us the angle of incidence and

ray paths. The proposed method was presented for first arrivals of acoustic waves, but extensions to anisotropic velocity fields are possible. The numerical experiments showed that the accuracy of the proposed method is highly dependent on the initialization step, and the estimation of the Laplacian of the time field. The method may potentially perform optimally in conjunction with accurate ray tracing methods in the vicinity of the source. The presented algorithm takes only first arrivals into account and is therefore a high-frequency approximation. As shown in our paper, it is possible to estimate several wave phenomena components in conjunction with first-arrival travel-time solvers. In fact, computing the amplitudes together with travel times increases the accuracy of the computed amplitude field when compared with post processing of the time field. Accurate amplitudes give rise to an additional source of information when imaging the subsurface. The methodology is well suited for parallel implementations, resulting in fast solvers for the Helmholtz Green's functions. The proposed method can be applied to different components of a wave and presents the first step to a faster tool to model wave motion. An accurate and fast tool for forward modeling of wave phenomena could work as a viable alternative to full wave-form modeling in several seismic applications. Our methodology is especially appealing when taken into consideration that the computational time is in the order of seconds.

10 Acknowledgements

The presented work was funded by Kalkulo AS and the Research Council of Norway under grant 238346. The work has been conducted at Kalkulo AS, a subsidiary of Simula Research Laboratory. We would like to thank Stuart Clark, Are Magnus Bruaset and Christian Tarrou for helpful comments and support.

Bibliography

[1] M.-G. Crandall and P.-L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of American Mathematical Society*, 277(1):1–42, 1983.

[2] T. Gillberg. *Fast and accurate front propagation for simulation of geological folds*. PhD thesis, Falculty of mathematics and natural sciences, University of Oslo, 2013.

[3] T. Gillberg, Ø. Hjelle, and A.-M. Bruaset. Accuracy and efficiency of stencils for the eikonal equation in earth modelling. *Computational Geosciences*, 16(4):933–952, 2011.

[4] T. Gillberg, M. Sourouri, and X. Cai. A new parallel 3–D front propagation algorithm for fast simulation of geological folds. *Procedia Computer Science*, 9:947–955, 2012.

[5] Tor Gillberg, Are M Bruaset, Øyvind Hjelle, and Mohammed Sourouri. Parallel solutions of static hamilton-jacobi equations for simulations of geological folds. *Journal of Mathematics in Industry*, 4 (1):10, 2014.

[6] E. Haber and S. MacLachlan. A fast method for the solution of the Helmholtz equation. *Journal of Computational Physics*, 230(12), 2011.

[7] M. Herrmann. A domain decomposition parallelization of the fast marching method. Technical report, German research foundation, 2003.

[8] Won-Ki Jeong and Ross T. Whitaker. A fast iterative method for eikonal equations. *SIAM Journal on Scientific Computing*, 30(5):2512–2534, 2008.

[9] B Julian and D Gubbins. Three-dimensional seismic ray tracing. *J. geophys*, 43(1):95–114, 1977.

[10] S. Leung, J. Qian, and R. Burridge. Eulerian Gaussian beams for high-frequency wave propagation. *Geophysics*, 72(5):SM61–SM76, 2007.

[11] S. Luo, J. Qian, and H. Zhao. Higher-order schemes for 3–D first-arrival traveltimes and amplitudes. *Geophysics*, 77(2):T47–T56, 2012.

[12] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C Lin, and Dinesh Manocha. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[13] P. Podvin and I. Lecomte. Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools. *Geophysical Journal International*, 105(1), 1991.

[14] N. Rawlinson and M. Sambridge. Seismic traveltime tomography of the crust and lithosphere. *Advances in Geophysics*, 46:81–197, 2005.

[15] N. Rawlinson, J. Hauser, and Sambridge M. Seismic ray tracing and wavefront tracking in laterally heterogeneous media. Technical report, Research School of Earth Science, 2008.

[16] J.-A. Sethian and A.-M. Popovici. 3-D traveltime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.

[17] Herb Sutter. The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb's journal*, 30(3):202–210, 2005.

[18] V. Červený. *Seismic ray theory*. Cambridge University Press, 2005.

[19] J. Vidale. Finite-difference calculation of travel times. *Bulletin of the Seismological Society of America*, 78(6):2062–2076, 1988.

[20] J. Vidale. Finite-difference calculations of traveltimes in three dimensions. *Geophysics*, 55(5):521–526, 1990.

[21] V. Vinje, E. Iversen, and H. Gjøystdal. Traveltime and amplitude estimation using wavefront construction. *Geophysics*, 58(8):1157–1166, 1993.

[22] Ofir Weber, Yohai S Devir, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Transactions on Graphics (TOG)*, 27(4):104, 2008.

[23] J. Zhang, Y. Huang, L. P. Song, and Q. H. Liu. Fast and accurate 3–D ray tracing using bilinear traveltime interpolation and the wave front group marching. *Geophysical Journal International*, 184(3):1327–1340, 2011.

[24] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250): 603–627, 2004.

[25] Hongkai Zhao. Parallel implementations of the fast sweeping method. *Journal of Computational Mathematics-International Edition*, 25(4):421, 2007.

Chapter 3

Wave-Motion Modeling on Parallel Computer Architectures

Article published in Elsevier's *Journal of Computational Science*, November 2015
DOI: 10.1016/j.jocs.2015.10.008

Research Paper 1 showed that the solution to the eikonal and transport equations can be obtained efficiently. Both equations are results of the high-frequency approximation of the wave equation as outlined in Chapter 1. Both discussed equations represent a broader principle of nature than pure wave propagation. Any front that moves depending on some kind of underlying velocity structure, can be described by the eikonal equation. The corresponding energy can be described by the transport equation. That makes a good solution strategy for these equations so abundantly applicable. However, in the context of wave imaging the strategy runs into problems. As described, eikonal and transport equations are results of a high-frequency approximation in which the signal at a point only depends on parameters along a one-dimensional curve, the so-called ray path. Since we are solving for first arrivals, rays of later arrivals are automatically lost, and with them, valuable information. Caustics in the wave front are areas where the time field is not differentiable and therefore the amplitude computation is, strictly speaking, not possible. Also, dispersion of a wave cannot be accounted for in the ray approximation, and interference will always be constructive since only wave fronts are considered.

The described drawbacks are commonly tolerated because of the inexpensive solution process of eikonal and transport equations in comparison to full wave-form modeling which does not suffer from these issues. However, the combination of the newest parallel computer architectures and the nature of wave propagation gives rise to an interesting phenomenon. The nature of wave propagation will always have the sequential time component. Therefore, neither the solution process of the eikonal nor the one of the wave equation can be parallelized entirely. However, all space dimensions can be computed in parallel. Despite the fact that the eikonal solution does not have a time dimension, physically we are propagating a wave front in time, which is represented by the sweeps in axial directions in the fast-sweeping method and the one-node-at-a-time restriction in a front-tracking approach. Regardless of the solution method, we cannot overcome the natural, sequential flow of time. The solution process for eikonal and transport equations as described in Chapter 2 takes advantage of two parallel dimensions, which is the maximum number considering one time dimension in the three-dimensional solution. The four-dimensional solution

of the wave equation, on the other hand, exhibits three spatially parallel dimensions and the sequential time dimension.Therefore, considering a perfect parallelization process, the first arrival computation using the eikonal or the wave equation is equally expensive (or rather inexpensive). Furthermore, the solution of the wave equation is not a high-frequency approximation and contains much more information.

These findings motivate the shift in focus from eikonal and transport equations to the investigation of the wave equation. Many different methods have been proposed in the past to solve the wave equation; some of them achieved great success [3, 35, 14]. When observing a wave in nature, we cannot help but notice that a wave is subject to causality. There are active and inactive regions, and activity has to be transported and cannot occur randomly.

These observations give rise to a new method for wave-motion modeling as presented in the next research paper. The method was found to be worth protecting by a US patent.

A Two-Scale Method Using a List of Active Sub-Domains for a Fully Parallelized Solution of Wave Equations

Marcus Noack[1,2,3]

[1]Kalkulo AS, P.O.Box 134, 1325 Lysaker, Norway
[2]Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
[3]Department of Informatics, University of Oslo, Gaustadalleen 23 B, 0373 Oslo, Norway

Abstract

Wave-form modeling is used in a vast number of applications. Therefore, different methods have been developed that exhibit different strengths and weaknesses in accuracy, stability and computational cost. The latter remains a problem for most applications. Parallel programming has had a large impact on wave-field modeling since the solution of the wave equation can be divided into independent steps. The finite-difference solution of the wave equation is particularly suitable for GPU acceleration; however, one problem is the rather limited global memory current GPUs are equipped with. For this reason, most large-scale applications require multiple GPUs to be employed. This paper proposes a method to distribute the workload on different GPUs by avoiding devices that are running idle. This distribution is done by using a list of active sub-domains so that a certain sub-domain is activated only if the amplitude inside the sub-domain exceeds a given threshold. During the computation, every GPU checks if the sub-domain needs to be active. If not, the GPU can be assigned to another sub-domain. The method was applied to synthetic examples to test the accuracy and the efficiency of the method. The results show that the method offers a more efficient utilization of multi-GPU computer architectures.

1 Introduction

Wave propagation plays a central role in many fields such as physics, environmental research and medical imaging to model acoustics, solid state physics, seismic imaging and cardiac modeling [14, 1, 17, 2, 11]. Different methods have been proposed for stable and accurate solutions of the wave equation, but the computational costs remain a problem for most applications [14].

The most commonly used methods to solve the wave equation can coarsely be divided into finite-element methods [13, 20], including spectral element methods [18], and explicit and implicit finite difference methods [10, 5]. The finite difference method is especially suitable for GPU acceleration because of the simple division into independent operations [15]. The solution in the current time step depends only on solutions of the previous time steps; hence, all nodes can be computed in parallel. The numerical solution of the wave equation is a memory demanding process since desired frequencies, model

sizes and wave velocities lead to a large number of wavelengths in the domain which imposes large grid sizes.

Two examples should be mentioned here. The first example is in the field of acoustics [14, 1], where the model size rarely exceeds 100 meters. Mehra et al. [14] presented the problem of a cathedral, where the sound velocity and the desire for a large range of frequencies requires a grid size of $22 \cdot 10^6$ nodes. Seismic imaging represents the second example, where the model dimensions are often in the order of a few hundred kilometers [9, 6, 12, 16] in lateral and vertical extension. For minimal wave velocities of $300\ m/s$ and frequencies of $10\ Hz$, the final grid size is around $16 \cdot 10^9$ nodes. For stability reasons it is not possible to choose the step size freely, which increases the computational cost further. Current GPUs have a maximum global memory of 24 gigabytes (K80 Tesla GPU); therefore, they can store around $6.4 \cdot 10^9$ single precision floating point numbers.

Since the resulting array is not the only data that has to be stored in the global memory of the GPU, the actual possible problem size is much smaller. Additionally, demands for accuracy and domain size are growing constantly and will always exceed the available resources. A solution to the problem is distributing the workload and data to different GPUs. The traditional approach is to assign one GPU to one specific sub-domain. For the entire computation, this assignment is static; therefore, most GPUs remain idle during the largest period of the computing time (see Figure 3.1) [15, 12, 16]. To address this issue, a list of active sub-domains can be used, as described in the following section.

The idea of considering exclusively the active part of a computation to save computing resources is not new. Di Gregorio et al. [4] employed the concept of active and inactive regions for wildfire susceptibility mapping (see also [3]). A rectangular bounding box distinguishes active from non-active regions and only active regions are computed. The bounding box method is also used by Zheng et al. [22] for flow simulation on GPU computer architectures. Teodoro et al. [19] proposed a method for an efficient wavefront tracking that only uses active elements which form the wavefront. The advancements in this case enable an efficient image processing. Zhao et al. [21] used local grid refinement to restrict the computation to active regions of interest.

2 A List of Active Sub-Domains

Gillberg et al. [8] introduced a list of active sub-domains for the simulation of geological folds by solving a static Hamilton-Jacobi equation. In the proposed method, the idea of Gillberg et al. [8] is adapted and used for the solution of the wave equation on multiple GPUs. The solution process for static Hamilton-Jacobi equations is very different from the solution process of the wave equation, and the application of the idea in Gillberg et al. [8] is therefore neither on domain nor on sub-domain level straightforward. The main differences are the dimensionality of the problem, the solution process on sub-domain level, e.g., the required stencil shapes, and the desired employment of multi-GPU computer architecture.

The solution of a static Hamilton-Jacobi equation, in Gillberg et al. [8], is found by a fast-sweeping method on sub-domain level which sweeps until convergence to find the viscosity solution. In order to parallelize the solution process, a pyramid-shaped stencil is used to compute nodes of an entire plane independently. Different stencil shapes require different ghost-node configurations and, therefore, different communication schemes. Since the solution of the wave equation is not an iterative process that needs to converge to a minimum, the activation patterns for sub-domains and the solution process on sub-domain level are very different in Gillberg et al. [8] from the method proposed herein. Furthermore, the method

by Gillberg et al. [8] is not developed to be used on a multi-GPU computer architecture; it is rather made to solve problems where strongly bent characteristic curves of the static Hamilton-Jacobi equation occur.

The adaption of the method in Gillberg et al. [8] includes, among others, the following: the establishment of an efficient communication between multiple GPUs, the adjustment of the activation pattern for sub-domains to the wave equation, implementing a different synchronization process, handling the fourth dimension and the employment of a different ghost-node configuration. However, the nomenclature is based on the one in Gillberg et al. [8] to simplify the comprehension for the reader.

The proposed method distributes the workload and data efficiently on different GPUs by activating sub-domains in which the wave exhibits amplitudes larger than a given threshold and adding these sub-domains to a list. Only the sub-domains on this list are distributed over available GPUs. During the computation on the sub-domain level, each GPU checks if the computed sub-domain needs to be active and, therefore, locks the domain for computation if the wave has traveled out of the domain boundaries. Therefore, the effective problem size can be decreased by orders of magnitude depending on the problem itself and the computing capacities.

The proposed approach is able to decrease the demands of computing resources for a given desired computational performance since it avoids idle GPUs. In the case of an abundant number of GPUs, the method allows increasing the number of sub-domains, and hence, improves upon the accuracy of the solution. More sub-domains also offer a more accurate isolation of active from inactive regions and, therefore, increase the performance (see Figure 3.2).

The method was implemented for the acoustic wave equation but can simply be adapted to more complicated scenarios. It should also be mentioned that the main scope of the proposed method is on multi-GPU computer architectures. However, every single GPU can be divided into independent parts to simulate a GPU cluster. This duality makes the method applicable on every parallel computer architecture and was used for all presented experiments. Furthermore, the method was developed for GPU computer architectures but the used principle leads to a speedup on all kinds of parallel computer architectures.

The remainder of the paper is organized as follows. The theory section gives an overview of the basic methods and the main principles of the algorithm, beginning with a summary of the mathematics and physics of the wave equation, followed by the description of the implementation. The method was applied to synthetic examples with different grid sizes.

3 Theory

The goal of the proposed method is to solve the wave equation, given by

$$\begin{aligned} \frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} &= v(\mathbf{x})^2 \nabla^2 u(\mathbf{x},t) \\ u(\mathbf{x},0) &= f(\mathbf{x}) \\ \frac{\partial u(\mathbf{x},0)}{\partial t} &= 0, \end{aligned} \tag{3.1}$$

on large grid sizes as efficiently as possible. Here, $u(\mathbf{x},t)$ is a scalar function, $v(\mathbf{x})$ is the wave velocity, ∇ is the nabla symbol and $f(\mathbf{x})$ is a scalar function describing the initial wave field. It has to be said that the proposed method is designed to solve all kinds of wave equations as efficient as possible. The acoustic wave equation is chosen here as an example for simplicity. To solve Equation (3.1) with the help of an

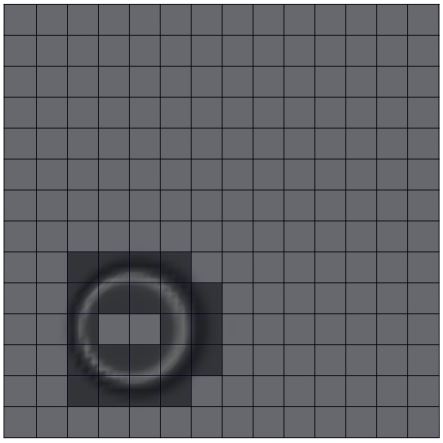


Figure 3.1: A snapshot of a propagating wave. The domain is divided into 196 sub-domains. Only 21 (labeled dark) of 196 sub-domains need to be active to compute the next time step. Therefore, in the traditional approach 89 percent of the GPU devices are running idle in the computation of the current time step.

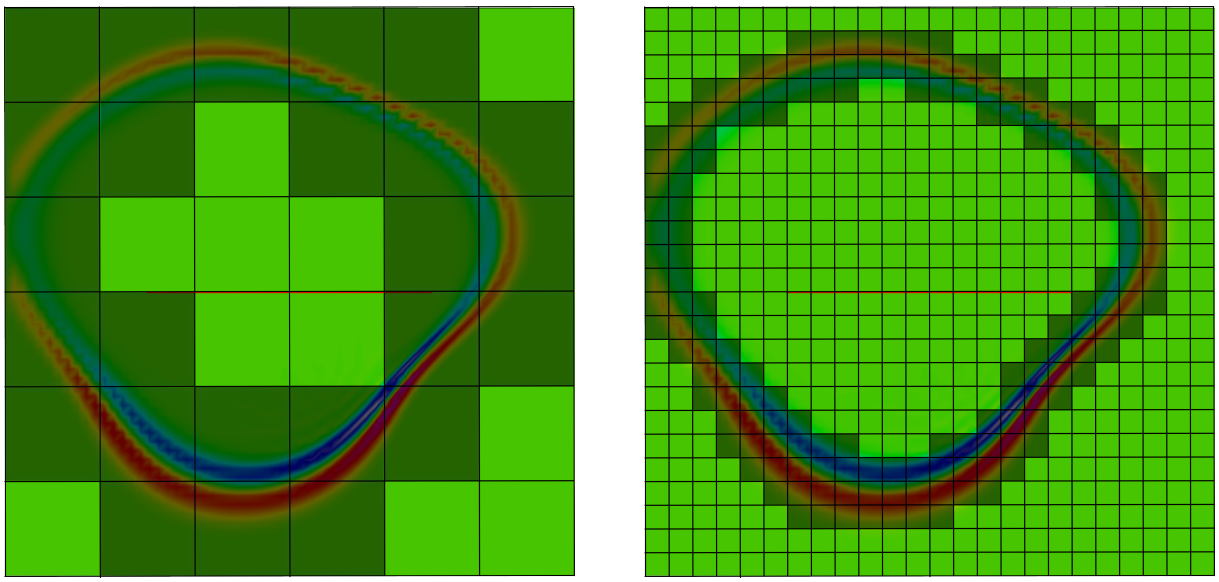


Figure 3.2: Effective problem size compared to actual problem size for two different examples. The domain is divided into 36 sub-domains on the left side. The ratio of sub-domains to active sub-domains (labeled dark) is 1.44. The right side shows the same problem with a division in 576 sub-domains. The ratio of sub-domains to active sub-domains in this case is 3.81. The example shows that the computational costs benefit from more sub-domains since active regions can be separated from non-active regions more accurately. This principle has a limit: the sub-domains need to be large enough to fully utilize the GPU device. Therefore, in the case of an abundant number of GPUs, the number of nodes in a sub-domain, and hence, the resolution can and should be enlarged to optimally utilize all available GPUs.

explicit finite difference scheme, it is mandatory to derive the finite difference approximation for the wave

Algorithm 1: Pseudo code for the top-level structure of the proposed algorithm. The first two time steps (0 and 1) must be given, therefore the loop starts with $i = 2$.

```
Initialization;
BuildSchedule(List,CL);
for i=2,...,TimeSteps do
    ComputeSchedule(CL,List,NumbSched);
    SyncSd(CL);
    BuildSchedule(List,CL);
```

equation, given by

$$u_{ijk}^{t+1} = v_{ijk}^2 dt^2 \nabla^2 u + 2u_{ijk}^t - u_{ijk}^{t-1}, \tag{3.2}$$

where $\nabla^2 u$ has not yet been discretized. Note that all nodes in the time step $t + 1$ are independent of all other nodes in the same time step. All values depend only on the values of past time steps, thus the solution process exhibits abundant parallelization. The computed wave field $u(\mathbf{x})^{t+1}$ in a certain time step will be the needed wave field $u(\mathbf{x})^t$ in the next time step and $u(\mathbf{x})^t$ will be the required $u(\mathbf{x})^{t-1}$ in the subsequent time step. Therefore, provided that the computation takes place only on one GPU, only data has to be copied to the device in the initialization step. This advantage is preserved in the case of multi-GPU computation. The algorithm checks if GPU devices and the data set on their global memory can be reused. If so, pointers are redirected one time step backward; therefore, no copying of new data is necessary as long as no new sub-domain is activated.

To guarantee the possibility for a correctly working communication between the sub-domains and to eliminate the need for communication during the computation, the incorporation of a sufficient amount of ghost-nodes around each sub-domain is necessary. Ghost-nodes are copies of nodes in adjacent domains (see Figure 3.4) [8, 7]. For accuracy reasons, in the proposed approach, a central finite difference scheme of fourth order was used for the second derivatives of the Laplacian

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i} \approx \frac{-u_{i+2} + 16u_{i+1} - 30u_i + 16u_{i-1} + u_{i-2}}{12\Delta x_i}. \tag{3.3}$$

Computing on the CPU or on one GPU, Equation (3.3) requires the domain setting illustrated in Figure 3.3. Two layers of nodes cannot be computed because of the spatial extent of the Laplacian. The communication between the sub-domains works with the same (sub-)domain setting. Therefore, the sub-domains for the multi-GPU computation are padded by two ghost-node layers at each side as illustrated in Figure 3.4 [16]. The use of different stencil shapes for the computation of the Laplacian requires the adjustment of the ghost-node configuration. Algorithm 1 shows the top-level structure of the implementation of the method. It consists of a loop over all time steps. In every time step, the algorithm computes all tasks of the current schedule, synchronizes the sub-domains and builds a new schedule. In the pseudo-code presented in this paper, the number of sub-domains is denoted by s_x,s_y and s_z, respectively. The size of a sub-domain is denoted by b_x, b_y and b_z, respectively. The wave field array and the velocity array are stored by sub-domain. Therefore, the velocity array is a four-dimensional array. The first three dimensions describe the sub-domain (ii, jj, kk), and the last dimension represents a flattened array that describes the position in the sub-domain $((i * (b_y + 4) * (b_z + 4)) + (j * (b_z + 4)) + (k))$, where the 4 originates from the ghost-node layers. The wave array is handled in a similar way with an additional time dimension.

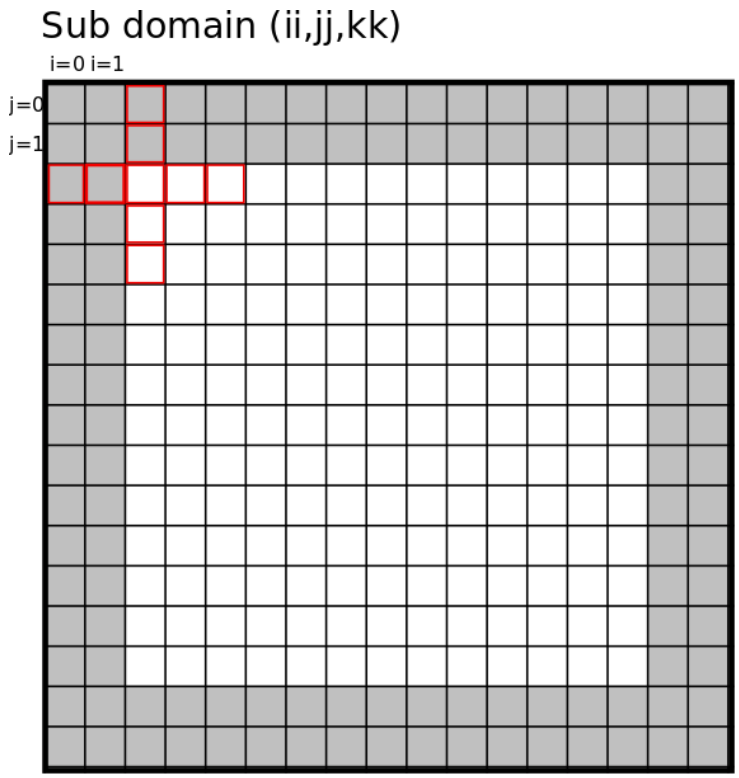


Figure 3.3: The domain for the computation on one GPU or the CPU. The extent of the stencil for the Laplacian is shown in red. Two layers of nodes on each side cannot be computed.

Thus, the wave array is five-dimensional $(u(timestep, ii, jj, kk, pos.\ in\ sub\ domain))$. Since the last dimension for the sub-domain array is flattened, the treatment with CUDA is very straightforward.

3.1 Building the List of Active Sub-Domains

In the initialization step, the wave field is defined for the first two time steps in accordance to Equation (3.1). If a node gets a value assigned larger than a given threshold, the corresponding sub-domain is activated. Activation means that the corresponding value in a boolean array (CL in the pseudo-code) gets the value "true" assigned. The coordinates of the sub-domains (denoted by ii, jj, kk) are written into a list. This list gives the method its name and can be seen as a schedule for the next computation. The sub-domains in the list are referred to as tasks. In each time step, the available GPUs are optimally assigned to the tasks in the schedule, considering the least necessary data transfer (for more explanation see Figure 3.5). Computing on the sub-domain level and synchronizing can change the activation of sub-domains; hence, it is important to build a new schedule after computation and synchronization.

3.2 Computation of the Schedule

After a list containing the schedule is built, every available GPU is assigned a task from the schedule, where one task equals one sub-domain. The corresponding sub-domains are copied to the different devices, where the next time step is computed in parallel. If a GPU is active a second time step in a row, data is not transferred again but reused to save computing time. During computation each GPU checks if at least one node in the sub-domain gets assigned an amplitude which is larger than a given threshold. If not, the

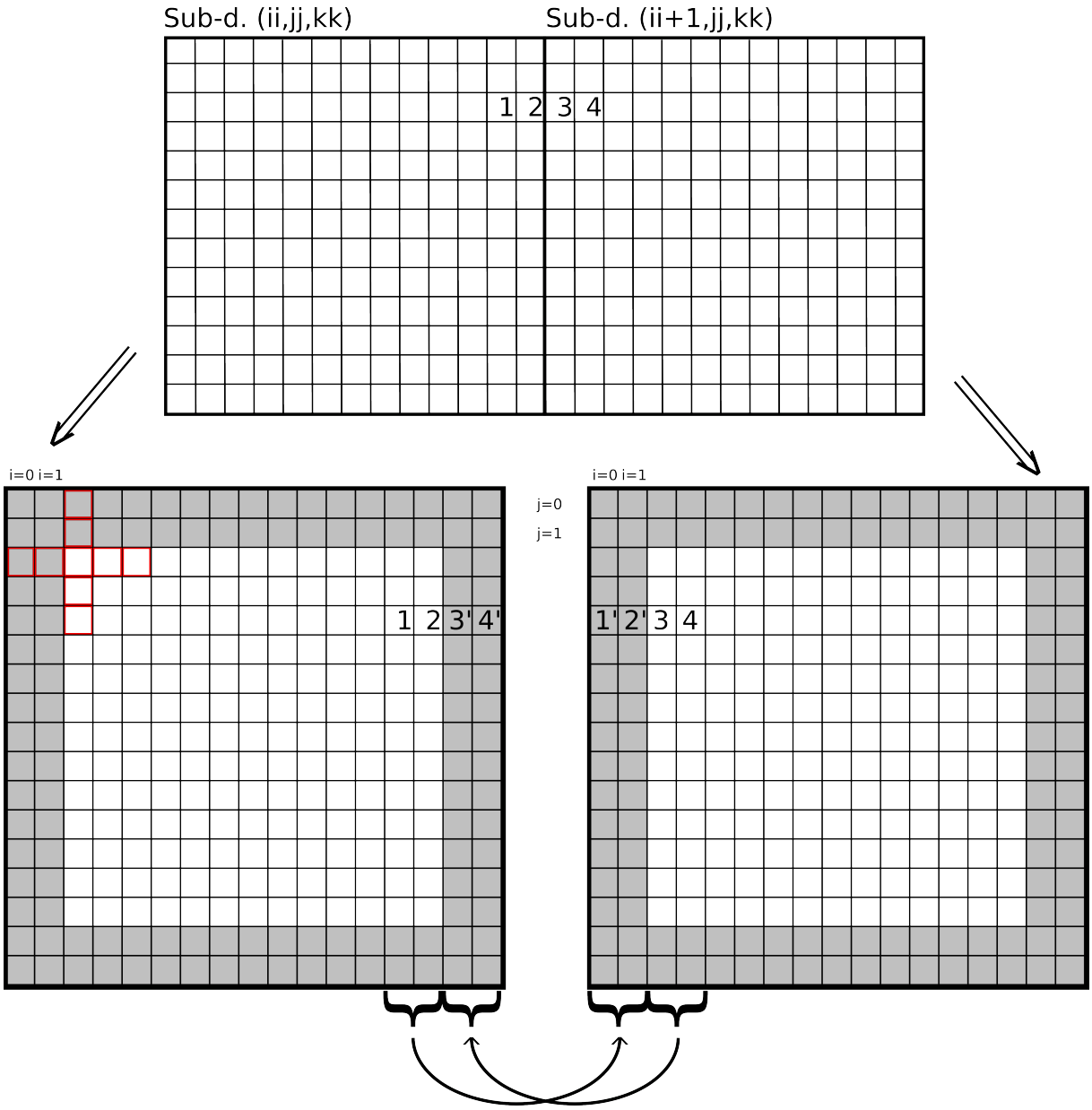


Figure 3.4: For a correctly working communication, the sub-domains (top) are surrounded by two layers of ghost-nodes in our computation in accordance with the extent of the stencil for the Laplacian. Ghost-nodes (X') are copies of the corresponding node (X). During synchronization, the values of the nodes 1 and 2 are first copied to the corresponding nodes in case their values are bigger than the given threshold. Afterwards, the values of the nodes 3 and 4 are copied to the corresponding location in case of sufficiently large amplitudes. The sub-domain is activated if at least one node in the sub-domain gets a new value assigned.

corresponding GPU tells the host that the sub-domain may be deactivated. Since several sub-domains are computed simultaneously and the computation on the sub-domain level is in parallel, the algorithm exhibits a two-level parallelization.

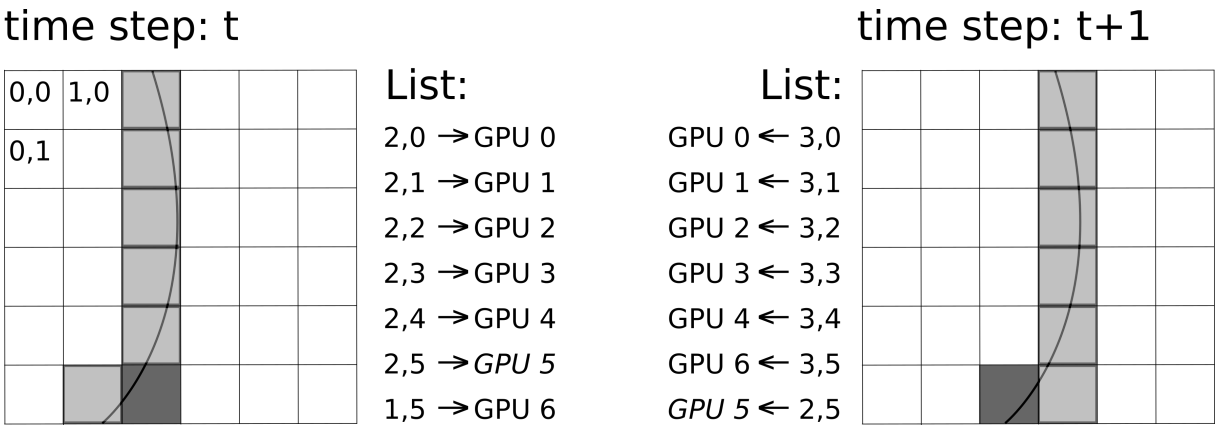


Figure 3.5: The figure shows two time steps of a propagating wave. For simplicity, the wave is represented as a wave front. Active sub-domains are written in the list. At time step t seven sub-domains are active, hence the work is distributed on seven GPUs. In the time step $(t + 1)$, the wave front moved out of some sub-domains into others. Hence, different sub-domains are active (in gray). Note that sub-domain (2,5) is active in both time steps (dark gray). Therefore, data can be reused and does not need to be copied. If the number of available GPUs is smaller than seven, in this example, the number of active sub-domains can be divided into groups of the number of GPUs.

Algorithm 2: BuildSchedule(LIST,CL)

```
NumbSched=0;
for ii = 0; ii < s_x; ++ii do
    for jj = 0; jj < s_y; ++jj do
        for kk = 0; kk < s_z; ++kk do
            if CL[ii][jj][kk]== "true" then
                NumbSched=NumbSched++;
                LIST[NumbSched][0]=ii;
                LIST[NumbSched][1]=jj;
                LIST[NumbSched][2]=kk;
        return (NumbSched);
```

Algorithm 3: ComputeSchedule(CL,List,NumbSched)

```
in parallel
for all tasks in schedule do
    select GPU device;
    Solve(CL,List,TaskInSchedule);
```

Algorithm 4: Solve(CL,List,TaskInSchedule)

```
allocation of memory;
cuda copy host to device;
DeviceFunction<<< blocks, threads >>>();
cuda copy device to host;
if device function discovers no amplitude > Threshold then
    CL[ii][jj][kk]="false"
```

Algorithm 5: SyncSd(CL)

```
/*Synchronization in positive x-direction*/
in parallel
for ii = 0; ii < sx − 1; ++ii do
    for jj = 0; jj < sy; ++jj do
        for kk = 0; kk < sz; ++kk do
            if CL[ii][jj][kk] == "true" then
                for i = bx; i < bx + 4; ++i do
                    for j = 0; j < by + 4; ++j do
                        for k = 0; k < bz + 4; ++k do
                            if |u[timestep][ii][jj][kk][i, j, k]| >= Threshold then
                                u[timestep][ii + 1][jj][kk][i − bx, j, k] =
                                  u[timestep][ii][jj][kk][i, j, k];
                                CL[ii+1][jj][kk] = "true";
```

3.3 Synchronization and Activation of Sub-Domains

After the computation of one time step, all sub-domains must be synchronized. For that, all ghost-nodes have to be copied to their corresponding position in the adjacent sub-domain. This process is taken care of by sweeps in positive and negative axial directions, one direction at a time, to avoid memory interference. A ghost-node is only copied to its corresponding position in the adjacent sub-domain if its value is larger than a given threshold. If a value of a node is copied to the adjacent sub-domain, this sub-domain is activated for the computation of the next time step. An if-condition makes sure that only sub-domains which were active in the last time step are synchronized to save computational costs.

4 Results

To prove the functionality of the proposed method, four key features were investigated. Firstly, to ensure that the accuracy of the traditional finite difference computation is preserved when applying the proposed method, resulting wave fields were compared. Secondly, computing times were measured to show that the list building step, which is additional work compared to the traditional method, only contributes a small amount to the overall computing time. Thirdly, overall computing times were compared. Finally, the ability of the new method to decrease the effective problem size is shown by means of a real-life situation. The first three key features were investigated on the basis of two different experiments that are introduced in the following sections. The fourth key feature was investigated on the basis of one experiment which was created to resemble a real-life seismological problem. The available computer architecture consists of two GeForce GTX 770 M GPUs. All experiments were designed to simulate a GPU cluster when necessary to obtain informative results by dividing one of the available GPUs into many processing units.

4.1 Experiment 1

Experiment 1 was designed to offer comprehensibility and clarity of the presented results. For Experiment 1 a domain of $248 \times 248 \times 248$ nodes was divided into $2 \times 2 \times 2$ sub-domains of $124 \times 124 \times 124$ nodes.

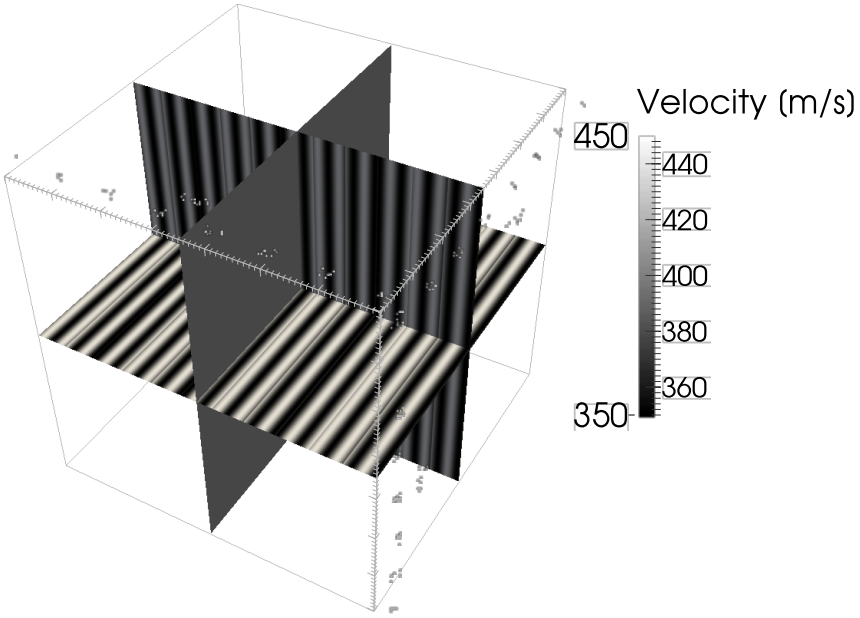


Figure 3.6: The velocity used to assess the accuracy of the proposed method.

The velocity was chosen to be homogeneous in the entire domain. Accounting for the ghost-nodes the resulting problem size was $256 \times 256 \times 256$ nodes. The initial condition was chosen to be a narrow Gaussian function. Due to the small problem size, it is possible to map the entire domain on one of the available GPUs.

4.2 Experiment 2

Experiment 2 was designed to investigate the performance of the method based on a real-life example. For Experiment 2, a domain of $308 \times 308 \times 308$ nodes was divided into $11 \times 11 \times 11$ sub-domains of $28 \times 28 \times 28$ nodes. The small sub-domain size makes it possible to simulate a computer architecture with 1331 GPUs on one of the available GPUs (not accounting for MPI communication). The velocity field was given by

$$v(\mathbf{x}) = 400 + (50 \times \sin(|\mathbf{x}| \times 38)) \tag{3.4}$$

and is illustrated in Figure 3.6. The chosen velocity field exhibits high frequencies and gradients of the velocity. It therefore represents a proper challenge for the proposed method. Accounting for the ghost-nodes, the resulting problem size was $352 \times 352 \times 352$ nodes. The initial condition was chosen to be a narrow Gaussian function.

4.3 Experiment 3

Experiment 3 was designed to prove the validity of the main essence of the proposed method: saving effective problem size. For the Experiment 3, a domain of $924 \times 924 \times 924$ nodes was divided into $33 \times 33 \times 33$ sub-domains of $28 \times 28 \times 28$ nodes. Accounting for the ghost-nodes the resulting problem size was $1056 \times 1056 \times 1056$ nodes. To make the result relevant for a real life application, the velocity field was chosen to represent a geological setting. The velocity model is shown in Figure 3.7.

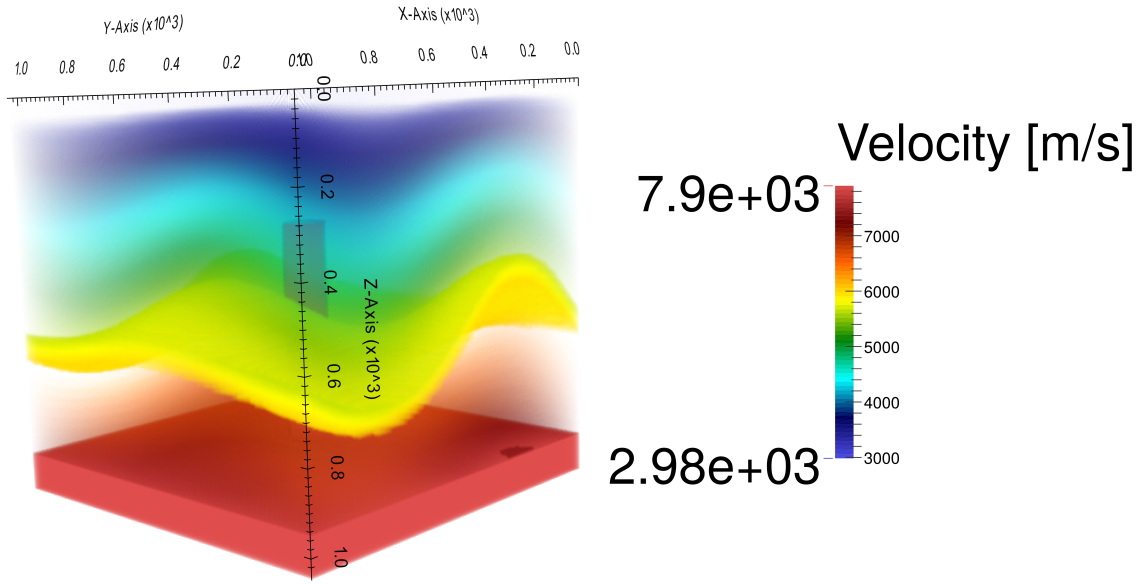


Figure 3.7: The velocity model chosen for Experiment 3. It is based on a real-life geological setting.

4.4 Comparison of Solutions

Since sub-domains are activated only if the amplitude of an approaching wave is larger than a certain threshold, one has to make sure that the lost information does not degrade the final solution. Therefore, the solution of the acoustic wave equation computed on the CPU using the traditional method was compared to the solution obtained with the new proposed method. For an elaborated analysis of the numerical accuracy, the L_1 and the L_2 norm, defined by

$$||u(\mathbf{x},t)||_{L_1} = \frac{\sum_{ijk} |u^t_{ijk} - \hat{u}^t_{ijk}|}{N} \tag{3.5}$$

and

$$||u(\mathbf{x},t)||_{L_2} = \frac{\sqrt{\sum_{ijk} (u^t_{ijk} - \hat{u}^t_{ijk})^2}}{N} \tag{3.6}$$

respectively, are presented. u^t_{ijk} in Equations (3.5) and (3.6) represents the solution of the proposed method and $\hat{u}^t_{ijk}$ represents the solution computed on the CPU without division into sub-domains. At first, the solution of Experiment 1 was compared with the solution on the CPU along a one-dimensional cross section (see Figures 3.8 and 3.9). For this example, the threshold was chosen to be 0.001% of the amplitude of the initial condition. The L_1 and L_2 error norms for different thresholds are presented in Figure 3.10. Next, Experiment 2 was conducted and compared to the corresponding computation on the CPU using the traditional method. The L_1 and L_2 error norms of the solution of Experiment 2 are presented for different thresholds in Figure 3.11. In order to determine the threshold, estimated amplitudes in the area of interest and numerical errors must be considered. For example, in a seismic scenario, the amplitude in the area of interest is important; there is no point in considering waves with an amplitude of $0.1\ mm$ if the computation is used to assess the risk of earthquake damage to buildings. However, for a computation of many time steps in a domain which is divided into many sub-domains, smaller thresholds should be considered for accuracy reasons.

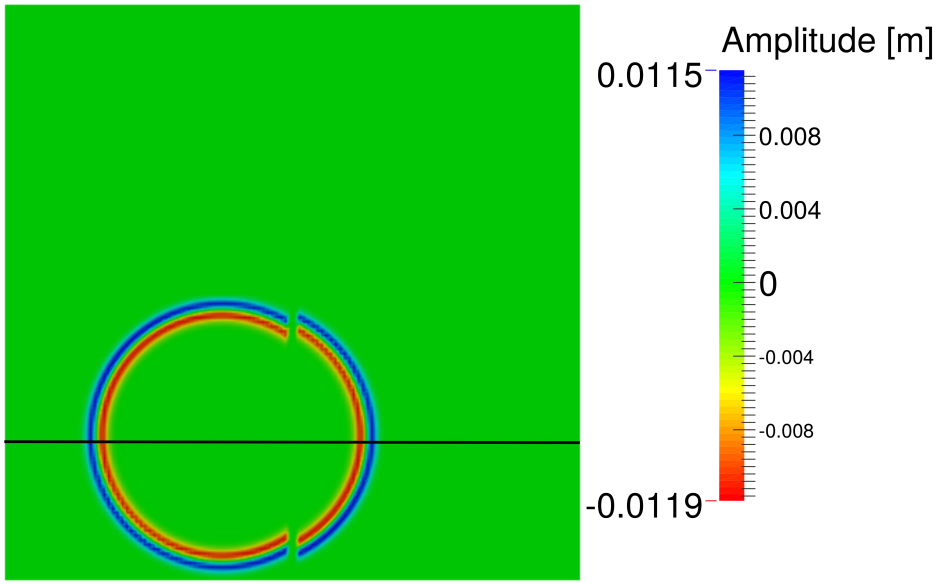


Figure 3.8: The figure shows the position and orientation of the cross section shown in Figure 3.9. The solution is shown in the domain that includes ghost-nodes, hence an offset is visible. This offset is not a numerical error and does not affect the final solution.

4.5 Time Measurement

In the current implementation, the computation of one time step consists of the solution of the acoustic wave equation, a synchronization of all active sub-domains and the building of a new schedule. To establish the proposed method as a standard way to solve the wave equation on multi-GPU computer architectures, it must be proven that the additional list building step does not take the majority of the overall computing time. In the synchronization step, the values of the ghost-nodes are copied to adjacent sub-domains and hence to other GPUs. The synchronization step is a necessary step in the traditional approach too and does therefore not need to be justified. However, in the current implementation, this step is not simultaneous to the solution process on the GPU. It is therefore included in the following measurements. For Experiment 1, the costs of synchronizing the sub-domains and building the new list amouts to 2% of the overall computational costs in the case of sequential synchronization. The synchronization in one direction can be a parallelized loop; thus, the synchronization and list building only takes about 0.5% of the overall computing time on a 4-core CPU machine (Intel Core i7-4800MQ CPU @ 2.70GHz). The percentage of the computational costs of the list building and synchronization step compared to the computation mainly depends on the ratio between the ghost-nodes and the overall number of nodes. The current implementation includes a condition to ensure that only active sub-domains are synchronized, which lowers the computational costs and represents an advantage compared to the traditional approach where all sub-domains, and hence all GPUs, have to communicate during the entire computing time, independent whether there is information to exchange or not. As a worst case scenario for the proposed method, Experiment 2 was conducted and the computing time of the list building and

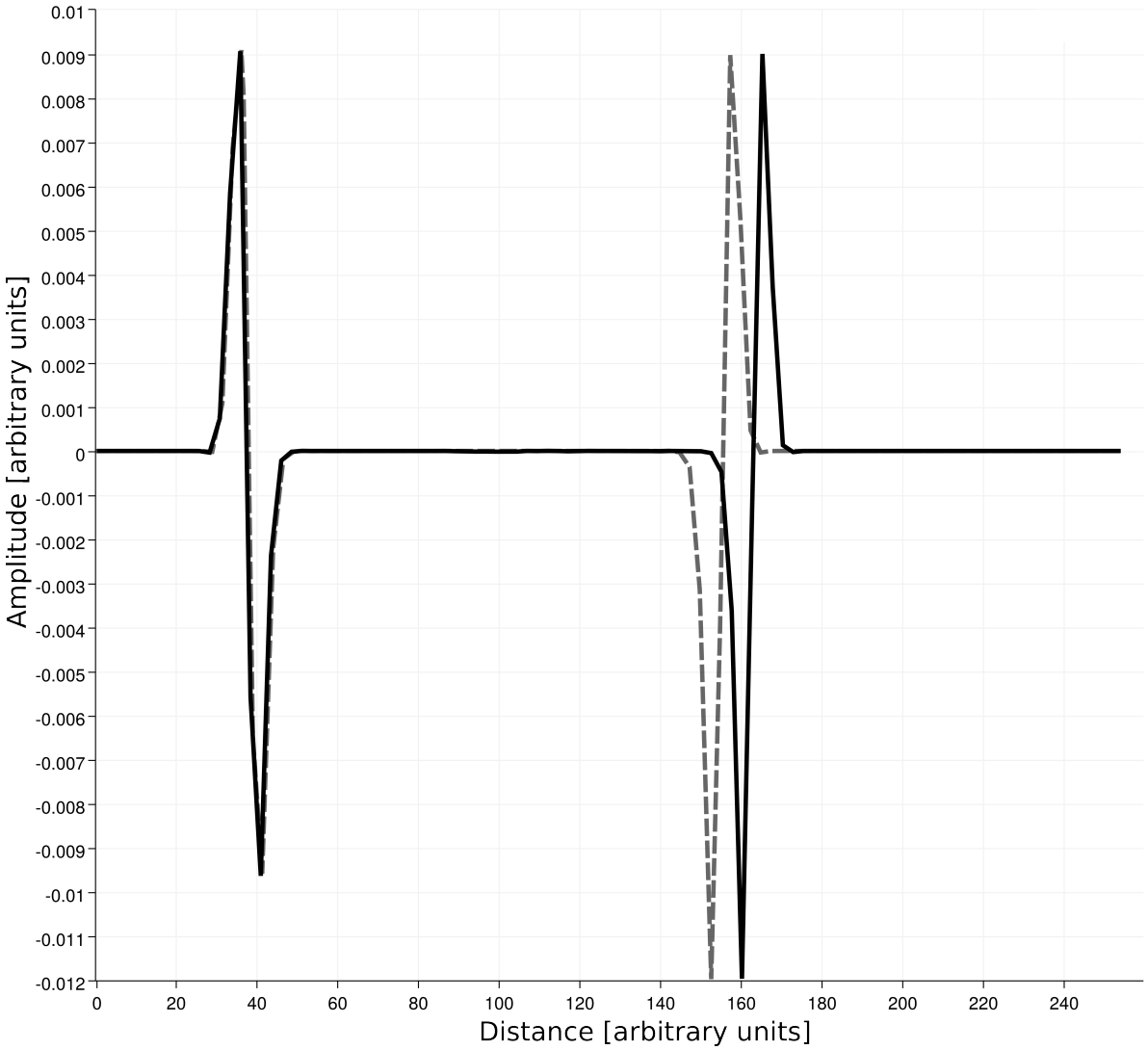


Figure 3.9: The illustration shows the comparison of the solution of the acoustic wave equation on one GPU (solid line) with the solution with the proposed method (dashed line) along a cross section (Figure 3.8). The wave form is similar. Note the occurrence of the phase shift because of the ghost-node layer which is purposely included. The phase shift is no numerical error and does not affect the final solution. The threshold for this example was chosen to be 0.001% of the amplitude of the initial condition.

synchronization steps was measured. The small sub-domains result in a low ratio of overall nodes to ghost-nodes which maximizes the synchronization time. For Experiment 2, the list building and synchronization steps needed 3.56% of the overall computing time using a sequential synchronization. In the case of a parallelized synchronization on a 4-core CPU machine, the list building and synchronization steps need below 1% of the overall computing time.

4.6 Computing Time

The new proposed method reaches full potential on multi-GPU clusters when the number of GPUs equals the maximum number of active sub-domains during the computation. Here, since the mentioned GPU cluster was not available, the problem size of Experiment 1 and 2 was chosen to simulate a GPU cluster

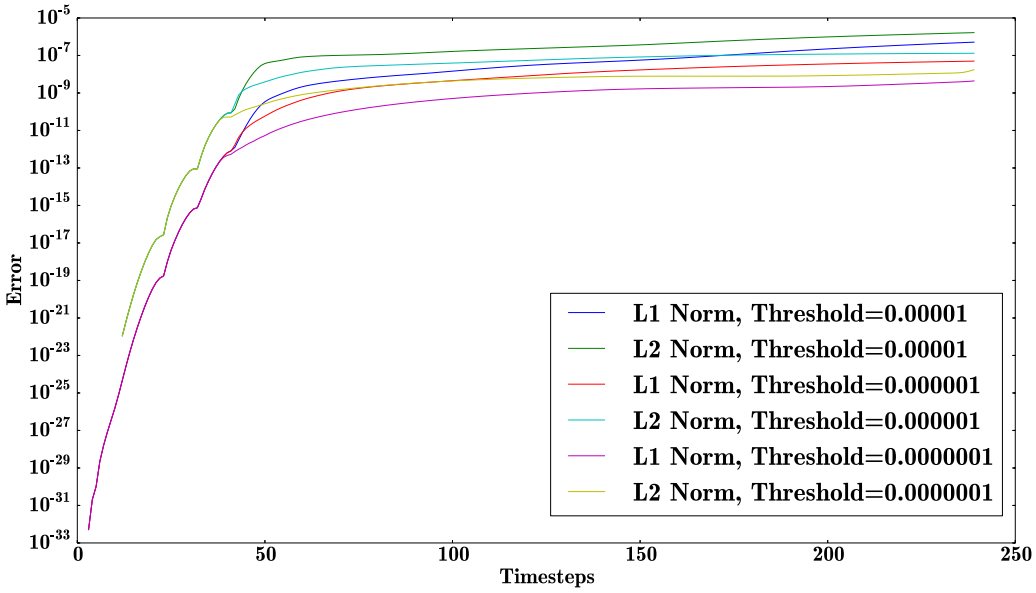


Figure 3.10: Representation of error norms L_1 and L_2 of the solution for the homogeneous velocity field for different time steps. The first deflection marks the first transition of the wave front into an adjacent sub-domain. Note that the error reacts strongly to the reduction of the threshold.

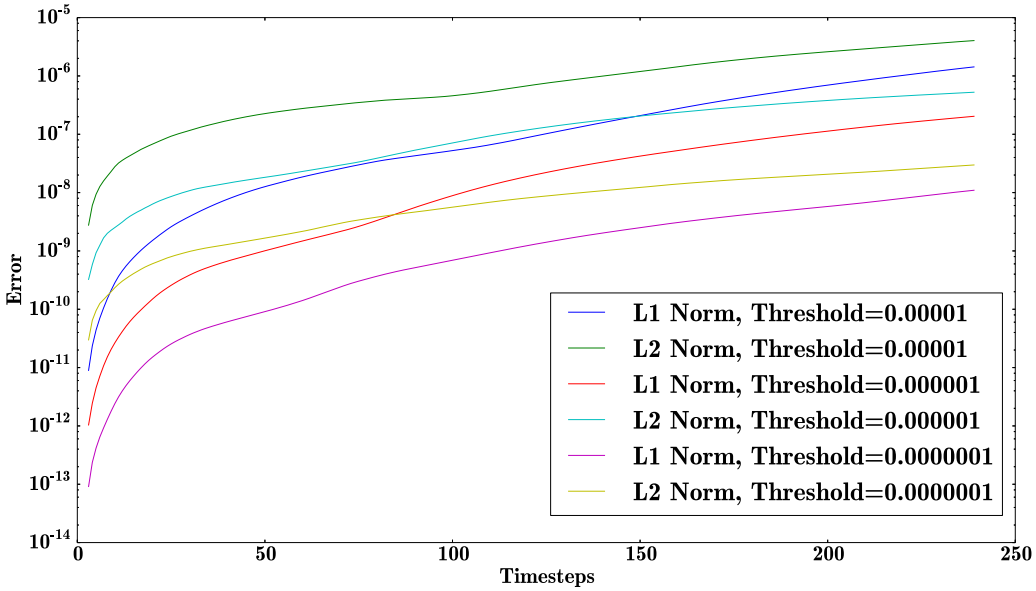


Figure 3.11: Representation of error norms L_1 and L_2 of the solution for the velocity field shown in Figure 3.6 for different time steps. Note once again that the error reacts strongly to the reduction of the threshold. The wave source in this experiment was located close to a sub-domain border, hence the first deflection is close to the origin.

which is able to communicate between GPUs in an instant.

Firstly, the computing time of Experiment 1 is presented. The computation was firstly performed in

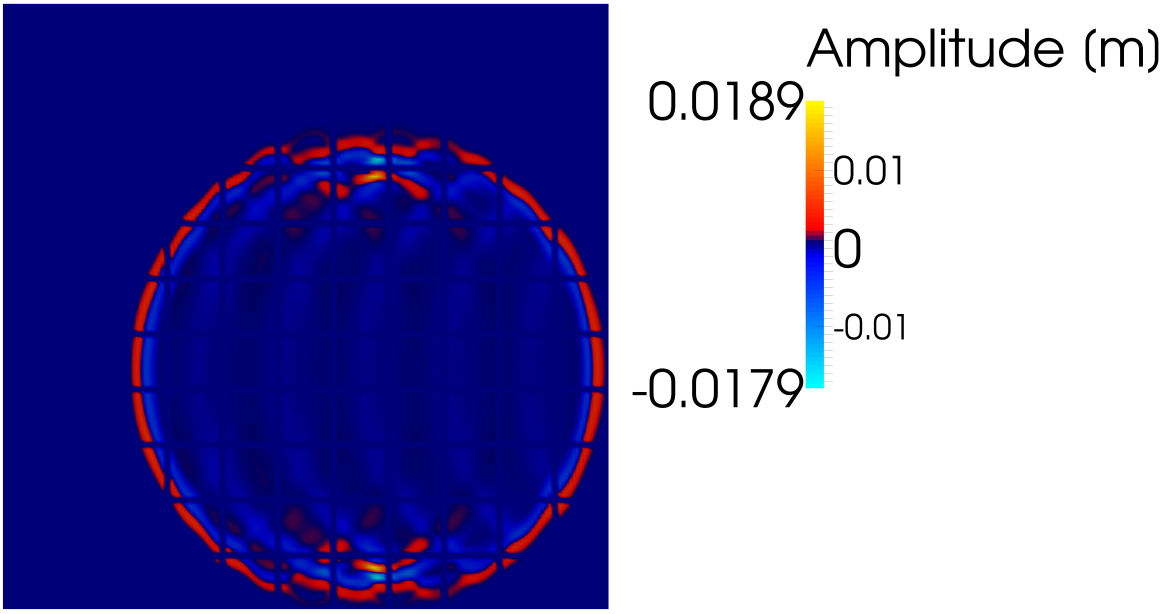


Figure 3.12: A slice of the resulting wave field computed with the new proposed method. Note that reflections make it impossible to deactivate sub-domains downwind of the wave front for the given configuration.

the traditional way, meaning that the entire domain was mapped on one GPU without using sub-domains. This result was compared to the same computation on one GPU and two GPUs using the proposed method. As soon as the number of active sub-domains exceeds the number of available GPUs, the computations becomes partly sequential. One GPU computed 100 time steps in 14.7 seconds using the traditional method. The new proposed method employed on one GPU only needed 4.84 seconds, which makes the computation 3.02 times faster. The proposed method needed 4.61 seconds for the same computation when two GPUs were used, resulting in a speedup of 3.19. The speedup in this example is due to the fact that the effective problem size was reduced to $124 \times 124 \times 124$ nodes for the first 80 time steps before the wave front propagated into adjacent sub-domains. Experiment 1 showed the functionality for small numbers of sub-domains. For a more elaborated investigation of the computing times, Experiment 2 was conducted and compared to the traditional method. For Experiment 2, one of the GPUs was divided into 1331 processing units to simulate a cluster of GPUs. To make the statement clear, the conditions for the traditional method were optimized. As described, since the traditional computation takes place on one GPU there is no communication step. Even for these optimized conditions for the traditional method, the speedup is significant compared to the proposed method. One GPU computed 150 time steps in 36.62 seconds on the mentioned grid. The new proposed method used only maximal 120 active sub-domains and needed 7.88 seconds, which makes the computation 4.64 times faster. For 300 time steps, the same computation takes 58 seconds using the proposed method and 73 seconds using the traditional approach. The speedup in this case amounts to 1.26 times. A slice of the wave field is shown in Figure 3.12. The computing times are summarized in Table 3.1.

Exp.	Trad. M.	No. Sub-d.	Architecture	Comp. Time	Speedup
1	14.7 s	8	1/2 GTX 770M	4.84s /4.61 s	3.02/3.19
2	36.52 s	11^3	1 GTX 770M	7.88 s	4.64

Table 3.1: Computing time in seconds for different experiments and computer architectures.

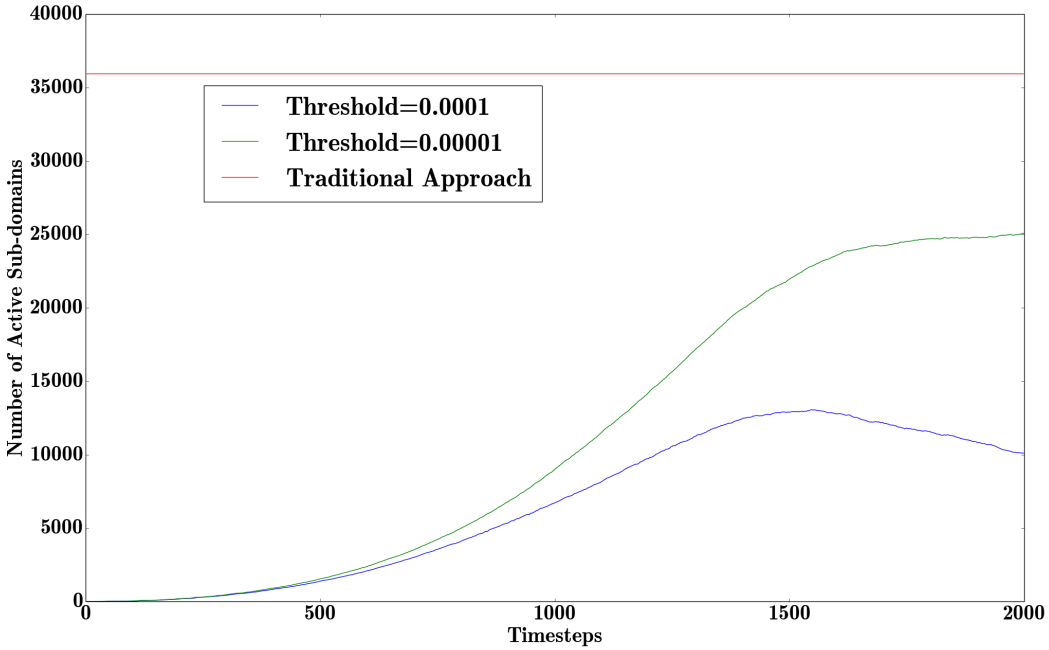


Figure 3.13: The number of active sub-domains as a function of the time steps is shown for two different thresholds and the traditional approach. The maximum number of active sub-domains is 13700 for a threshold of 10^{-4} and 25086 for a threshold of 10^{-5}. The traditional method employs 35937 sub-domains for the entire computing time. Note that the green graph approaches the maximum number of active sub-domains and the gradient approaches zero. Therefore, the number of new activated sub-domains per time step is close to the number of deactivated sub-domains in the same time step.

4.7 Saving Computing Resources

Experiment 3 was conducted to show how efficient the algorithm is in saving computing resources in a real-life situation. 2000 time steps were computed enabling the wave front to travel through all sub-domains. The number of active sub-domains in each time step for two different thresholds is shown in Figure 3.13. The maximum number of active sub-domains was 13700 or 25086, and on average 6563 or 11232 sub-domains were active for the thresholds 10^{-4} or 10^{-5}, respectively. To obtain a meaningful measure to compare the efficiency of the traditional and the proposed method, the number of overall computed nodes can be considered. Using the traditional approach, $229.76 \cdot 10^{10}$ nodes were computed. Using the new proposed method, $58.88 \cdot 10^{10}$ nodes were computed for a threshold of 10^{-4} and $106.24 \cdot 10^{10}$ nodes for a threshold of 10^{-5}. These results indicate that, using the novel approach in experiment 3, 74.4% (for a threshold of 10^{-4}) or 53.7% (for a threshold of 10^{-5}) of computing resources can be saved. The results are summarized in Table 3.2.

Threshold	Nodes Trad. Method	Nodes Prop. Method	Saving
10^{-4}	$229 \cdot 10^{10}$	$58.88 \cdot 10^{10}$	74.3 %
10^{-5}	$229 \cdot 10^{10}$	$106.24 \cdot 10^{10}$	53.7 %

Table 3.2: Number of computed nodes for the traditional method and two different thresholds, and the percentage of saved resources.

5 Discussion and Perspective

The results section showed that the new proposed method computes the same result as the computation on one single GPU with a significant improvement of computational efficiency. Figure 3.9 shows that there is only a negligible difference in amplitude. The phase shift is included intentionally to show the effect of the ghost layers and can easily be removed. The L_1 and L_2 error norms show how the error introduced by ignoring small values developed over time. The error increases strongly in the beginning but reaches a stable value after some time. It was shown that the error strongly reacts to the reduction of the threshold size. Therefore, for large problems smaller thresholds should be chosen. In these scenarios, the proposed method maintains its superiority over the traditional method since more sub-domains mean a more accurate separation of inactive from active zones. Larger problem sizes also allow for a bigger ratio of inactive to active zones since commonly emerging wavelength are smaller compared to the problem size. In other words: the larger the model size compared to the emerging wave lengths, the higher the possibility for inactivating most of the model space, especially when using a very time limited source term. This fact allows smaller thresholds when computing larger problems without loss of benefit. The comparison of the error norms of Experiment 1 and Experiment 2 also showed that the error increases only slightly for complex problems.

The beneficial effect of the method is obvious: regions where the amplitude of the wave is smaller than a certain threshold are not part of the computation and do not waste computing resources. This principle leads to a significant speedup, even for an example that is not perfectly suited for the method. Instead of one GPU dealing with $256 \times 256 \times 256$ nodes the algorithm activates only one sub-domain in the beginning, leading to a much smaller effective problem size. In later time steps, the adjacent sub-domains are activated. Since the number of sub-domains exceeds the number of GPUs the computation is partly sequential; however, the speedups of 3.02 times using one GPU and 3.19 times using two GPUs are still promising and in the expected range. For this example, a bounding box method would have yielded the same speedup because of the limited problem size and sub-domain number, which make it impossible to deactivate sub-domains behind the traveling wave. For more complex problems, sub-domains are deactivated as soon as the wave has traveled outside and the proposed method outperforms the bounding box method. In Experiment 1, eight GPUs would not have been much faster since the activation of most sub-domains happens in the last 20 time steps. Hence, most of the time the GPUs would have been idle. Furthermore, the proposed method makes the division into eight sub-domains using one or two GPUs possible in the first place. The speedup is mainly due to the fact that the effective problem size is reduced by a factor of eight for a large part of the computation. The rest of the computation is subject to a partly sequential computation due to the chosen problem size and hardware. Therefore, the measured speedups are in a reasonable range.

Experiment 2 simulates a real-life example computed on a GPU cluster equipped with 1331 GPUs.

Each GPU can compute the solution of the wave equation on a grid of $28 \times 28 \times 28$ nodes. Since the problem size is manageable by one GPU, the simulated cluster does not need to communicate when performing the traditional approach, therefore giving it an unrealistic advantage. During the computation using the traditional method, most of the simulated 1331 GPUs are waiting most of the time for their turn. On the other hand, the proposed method checks for active sub-domains and reduces the efficient problem size significantly to a maximum of 120 sub-domains in the first 150 time steps. The result is a 4.64 times faster computation. It has to be said, that the conducted experiment shows the traditional method at its best, and the new proposed method at its worst, since the high ratio of ghost-nodes to overall nodes maximizes the time for synchronization and list building steps. Even in this worst case scenario, the computing time for the list building and synchronization steps are small because only active sub-domains are synchronized with their neighbors and the synchronization can be performed in parallel. The same experiment conducted for 300 time steps showed a 1.26 times faster computation using the proposed method. The smaller speedup for 300 time steps is due to the special character of the velocity field. The high-frequency, periodic velocity field causes many reflections which make it impossible to inactivate sub-domains when using the given setting (see Figure 3.12). In this case, a larger grid and more time steps would be beneficial since the amplitude of the reflected waves would decay below the threshold at some point. Experiment 3 proved the ability of the new method to save computing resources on the basis of a real-life application. Instead of 35947 active sub-domains used by the traditional method, the new algorithm only activated a maximum number of 13700 or 25086 sub-domains depending on the size of the threshold. On average, 6563 or 11232 sub-domains were active. The overall number of computed nodes showed that the saving of computer resources is significant for the chosen experiment for both thresholds.

The success of the method highly depends on problem specific parameters, like source definition, velocity model and problem size, and on the used computer architecture. However, all wave propagation algorithms can benefit from the proposed algorithm in the beginning of the wave propagation. When the active wave field is only small, all GPUs can be used for a higher resolution, and hence, a higher accuracy of finite different approximations around the source. The proposed algorithm loses all its benefits as soon as a wave is active in all sub-domains. In this case the consumption of computing resources is the same as with the traditional method excluding the list building step. However, this scenario is rare in practice.

In the future, the sub-domains could be irregularly shaped and thus, better at isolating active from inactive zones. Furthermore, automatic tools that define sub-domains depending on wave activity and the number of available GPU devices could be very beneficial. Such a tool could divide the active regions into as many sub-domains as possible, resulting in higher resolution and/or computational performance. The goal is to optimally distribute computing resources only on active regions and not wasting them on regions in the domain where the wave exhibits negligible amplitudes.

6 Conclusion

This paper has proposed a method for distributing the workload of solving the wave equation on a multi-GPU computer architecture. The proposed algorithm can save computing resources by deactivating areas where the amplitude of the wave undergoes a defined threshold. The available computing resources are entirely utilized for regions where the wave is active; hence, no GPUs are running idle. Therefore, smaller clusters can perform equally well as larger ones. Using the proposed algorithm, one can divide the domain in more sub-domains than available GPU devices and still obtain a good performance. In cases

when enough GPUs are available, increasing the number of nodes, and thus the resolution of the solution, without losing computing time is possible. The proposed algorithm offers more efficient and accurate wave form modeling by optimizing the workload distribution on GPU clusters and therefore, has a large potential impact on industry and research.

7 Acknowledgements

The presented work was funded by Kalkulo AS and the Research Council of Norway under grant 238346. The work has been conducted at Kalkulo AS, a subsidiary of Simula Research Laboratory. I would like to thank Stuart Clark, Are Magnus Bruaset, Christian Tarrou and Xing Cai for beneficial comments and support.

Bibliography

[1] Dick Botteldooren. Finite-difference time-domain simulation of low-frequency room acoustic problems. *The Journal of the Acoustical Society of America*, 98(6):3302–3308, 1995.

[2] Jon F Clœrbout. *Imaging the earth's interior*. 1982.

[3] Donato D'Ambrosio, Salvatore Di Gregorio, Giuseppe Filippone, Rocco Rongo, William Spataro, and Giuseppe A Trunfio. Fast assessment of wildfire spatial hazard with gpgpu. In *SIMULTECH*, pages 260–269, 2012.

[4] Salvatore Di Gregorio, Giuseppe Filippone, William Spataro, and Giuseppe A Trunfio. Accelerating wildfire susceptibility mapping through gpgpu. *Journal of Parallel and Distributed Computing*, 73 (8):1183–1194, 2013.

[5] A. Fichtner. *Full seismic waveform modellng and inversion*. Springer, 2011.

[6] Scott French, Vedran Lekic, and Barbara Romanowicz. Waveform tomography reveals channeled flow at the base of the oceanic asthenosphere. *Science*, 342(6155):227–230, 2013.

[7] T. Gillberg. *Fast and accurate front propagation for simulation of geological folds*. PhD thesis, Falculty of mathematics and natural sciences, University of Oslo, 2013.

[8] Tor Gillberg, Are M Bruaset, Øyvind Hjelle, and Mohammed Sourouri. Parallel solutions of static hamilton-jacobi equations for simulations of geological folds. *Journal of Mathematics in Industry*, 4 (1):10, 2014.

[9] Stephen P Grand, Rob D van der Hilst, and Sri Widiyantoro. Global seismic tomography: A snapshot of convection in the earth. *GSA today*, 7(4):1–7, 1997.

[10] Heiner Igel, Peter Mora, and Bruno Riollet. Anisotropic wave propagation through finite-difference grids. *Geophysics*, 60(4):1203–1216, 1995.

[11] C. Kittel and P. McEuen. *Introduction to solid state physics*. Wiley new york, 1976.

[12] Dimitri Komatitsch, Gordon Erlebacher, Dominik Göddeke, and David Michéa. High-order finite-element seismic wave propagation modeling with mpi on a large gpu cluster. *Journal of Computational Physics*, 229(20):7692–7714, 2010.

[13] John Lysmer and Lawrence A Drake. A finite element method for seismology. *Methods in computational physics*, 11:181–216, 1972.

[14] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C Lin, and Dinesh Manocha. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[15] Paulius Micikevicius. 3d finite difference computation on gpus using cuda. In *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, pages 79–84. ACM, 2009.

[16] Kim B Olsen, Ralph J Archuleta, and Joseph R Matarese. Three-dimensional simulation of a sagnitude 7.75 earthquake. *Science*, 270:8, 1995.

[17] Armen P Sarvazyan, Oleg V Rudenko, Scott D Swanson, J Brian Fowlkes, and Stanislav Y Emelianov. Shear wave elasticity imaging: a new ultrasonic technology of medical diagnostics. *Ultrasound in medicine & biology*, 24(9):1419–1435, 1998.

[18] Géza Seriani and Enrico Priolo. Spectral element method for acoustic wave simulation in heterogeneous media. *Finite elements in analysis and design*, 16(3):337–348, 1994.

[19] George Teodoro, Tony Pan, Tahsin M Kurc, Jun Kong, Lee AD Cooper, and Joel H Saltz. Efficient irregular wavefront propagation algorithms on hybrid cpu–gpu machines. *Parallel computing*, 39(4): 189–211, 2013.

[20] Takumi Toshinawa and Tatsuo Ohmachi. Love-wave propagation in a three-dimensional sedimentary basin. *Bulletin of the Seismological Society of America*, 82(4):1661–1677, 1992.

[21] Ye Zhao, Feng Qiu, Zhe Fan, and Arie Kaufman. Flow simulation with locally-refined lbm. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 181–188. ACM, 2007.

[22] Jingwei Zheng, Xuehui An, and Miansong Huang. Gpu-based parallel algorithm for particle contact detection and its application in self-compacting concrete flow simulations. *Computers & Structures*, 112:193–204, 2012.

Chapter 4

The Duality of Anisotropy and Metric Space

Article published in Elsevier's *Heliyon* Journal, March 2017
DOI: 10.1016/j.heliyon.2017.e00260

Over the last decades, the focus of wave-motion modeling has broadened beyond geophysics and optics. More and more fields in research and industry adopt the description and modeling of wave motion, and particularly of wave motion in anisotropic media. A famous example is cardiac modeling, where an electrical wave excites the heart-muscle fibers to contract [41, 50]. In fact, most fields that deal with wave propagation need to consider anisotropy as soon as the underlying velocity cannot be described by a simple scalar [32, 9, 40, 13, 25, 24, 44, 39, 8, 23, 28, 6, 36, 33].

In this case, wave and eikonal equations have to be derived to fit certain conditions, like tensor valued velocity fields. It can be found, that many fields struggle to find the correct formulations of the governing equations. This motivates the work on a unified theory that allows for a simple and straight-forward derivations of equations describing wave propagation through certain kind of anisotropic materials. The theory exploits the duality of a transformed space and velocity. Research Paper 3[1] presents the results of the corresponding work.

[1] Research Paper 3 was written in British English due to author preference

Acoustic Wave and Eikonal Equations in a Transformed Metric Space for Various Types of Anisotropy

Marcus M. Noack[1,2,3] and Stuart Clark[1,2]

[1]Kalkulo AS, P.O.Box 134, 1325 Lysaker, Norway
[2]Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
[3]Department of Informatics, University of Oslo, Gaustadalleen 23 B, 0373 Oslo, Norway

Abstract

Acoustic waves propagating in anisotropic media are important for various applications. Even though these wave phenomena do not generally occur in nature, they can be used to approximate wave motion in various physical settings. We propose a method to derive wave equations for anisotropic wave propagation by adjusting the dispersion relation according to a selected type of anisotropy and transforming it into another metric space. The proposed method allows for the derivation of acoustic wave and eikonal equations for various types of anisotropy, and generalizes anisotropy by interpreting it as a change of the metric instead of a change of velocity with direction. The presented method reduces the scope of acoustic anisotropy to a selection of a velocity or slowness surface and a tensor that describes the transformation into a new metric space. Experiments are shown for spatially dependent ellipsoidal anisotropy in homogeneous and inhomogeneous media and sandstone, which shows vertical transverse isotropy. The results demonstrate the stability and simplicity of the solution process for certain types of anisotropy and the equivalency of the solutions.

1 Introduction

Anisotropic wave propagation has a variety of different applications because, compared with isotropic wave propagation, it is a more general representation of wave propagation and is valid in a wider range of materials. For instance, recent scientific attention has been focused on the development of devices that can appear to cloak objects. Prototypes of such devices have been constructed that bend certain wavelengths of light using metamaterials [30, 25]. In the case of acoustic waves, a device for preferential propagation of the sound through one of two possible and symmetrically aligned channels has been constructed, leading to effective anisotropic propagation of the wave [12]. Other metamaterials enhance the amplitude of sound waves before entering a microphone to enhance sensors [9]. These metamaterials are composed of sub-wavelength meta-atoms that lead to wavelength-scale effects such as non-reciprocal propagation of waves, negative refraction and wave isolation [23, 13]. Current cloaking technology is based on the concept of bending the wave around an object such that properties of the wave do not change, rendering

the object in between the source and the sensor undetectable. As such, coordinate transformations that maintained the invariance of the wave properties have been defined, for electromagnetic waves [27], elasto-dynamic waves [24] and for acoustic waves in 2D [10] and 3D [8]. In addition, there exists a mathematical similarity between acoustic waves (longitudinal, compressional or P-waves) and transverse or s-waves through mapping the material properties [22]. Therefore, the modeling of anisotropic acoustic waves can be used to study several types of wave propagation. In this paper, we present a new and generic approach to model anisotropic acoustic wave propagation based on metric space transformations.

Metric space transformations have been proposed to handle anisotropy before. Dellinger [11] proposed to stretch a circle to model elliptical anisotropy, while Joets and Ribotta [19] applied the idea for the propagation of light in anisotropic media where the ray trajectories are the geodesics of an anisotropic metric space. Borovskikh [4] used the duality of anisotropy and metric space for a theoretical treatment of various eikonal equations for anisotropic media.

The first advancements in anisotropic wave propagation were made by physicists in the 19th century to investigate the propagation of light; however the major advances in anisotropic wave propagation were in the field of seismology [17]. The full anisotropic behavior is defined by a fourth-order tensor c_{ijkl} to relate stress and strain. Due to the inherent complexity of this tensor, Voigt [35] noticed symmetries that allowed the $3 \times 3 \times 3 \times 3$ tensor to be reduced to a 6×6 symmetric matrix $C_{\alpha\beta}$. For particular anisotropic materials, the number of potential coefficients is further reduced; to 5, for example, for transversely isotropic materials, or to 9 for orthorhombic media [34]. For three-dimensional metric space transformations, 9 coefficients must be defined, as we will show below. In the next section, a method is presented that generalizes the procedure of deriving wave and eikonal equations for different kinds of anisotropy.

In nature, acoustic media does not physically admit body waves with anisotropic behavior; only a wave traveling along a curved manifold or through a moving medium can exhibit anisotropic behavior. However, it is possible to construct wave and eikonal equations for acoustic, anisotropic wave propagation by using the dispersion relation of the wave equation [1, 38]. The idea is to design a metric space for which the media properties are isotropic. [4]. Hence, using the duality of anisotropic media and metric space, the anisotropic case can be treated like the isotropic one. Therefore, the benefits of the solution of the acoustic wave equation in isotropic media, like the simplicity of the mathematical treatment and the stability of the numerical solution, are inherited by the resulting equations for anisotropy. For elastic media, the acoustic wave equation is used as an approximation of the plane wave (P-wave) motion [1], while ignoring shear waves [38, 1]. Other situations in which anisotropic, acoustic wave propagation can be encountered are electric waves in muscle tissue [37, 31] and acoustic waves in moving media.

A metric space is in general defined by a set for which distances between all elements are defined. The definition of the distance between elements of the set is called a metric. The metric induces a topology on the set which leads to our definition of anisotropy. The duality of a metric space and anisotropy shall in this paper be used to generalize various types of anisotropy into one theory. The proposed unified theory can be used to derive simple, stable and efficient numerical solvers for wave and eikonal equations, and for a better theoretical understanding of anisotropy. A form of stretching the elliptical anisotropy iteratively to obtain a solution to the transversely isotropic eikonal equation can be found in bin Waheed et al. [3].

In homogeneous media, a sphere describes the velocity surface for the isotropic case. For ellipsoidal anisotropy, the sphere can directly be transformed into the new metric space by using the corresponding

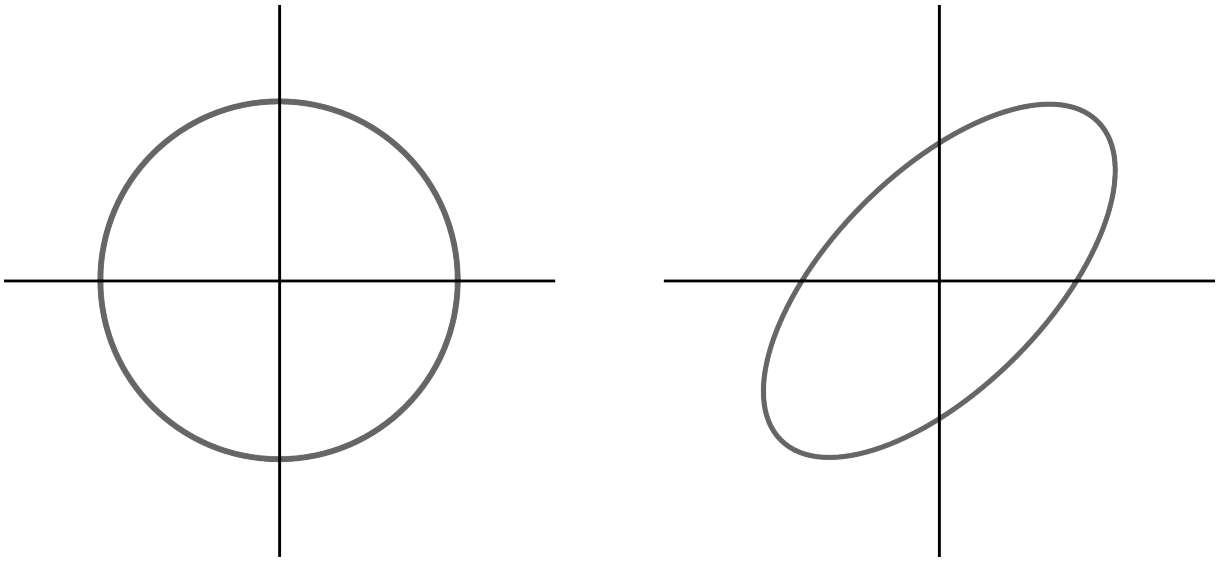


Figure 4.1: A comparison of a circle in different metric spaces. The circle on both sides of the figure is the set of all points satisfying $x^2 + y^2 = 1$. The underlying metric on the left side is the euclidean metric. The underlying metric on the right side is obtained by compressing and rotating the corresponding basis by 45 degrees with respect to the standard basis.

basis (Figure 4.1). Another version of this transformation was also used in Stovas and Alkhalifah [32]. This principle can be adapted for other types of anisotropy by using different L^p norms for the computation of distances. At first, a surface must be chosen to describe the wave front, which can then be stretched and tilted by transforming it into another basis for the velocity. From the resulting surface, a dispersion relation can be derived which leads to the corresponding wave equation. From the wave equation, the corresponding eikonal equation can be obtained. The eikonal approximation provides information about first arrivals [29], does not account for caustics [20] and requires a sufficiently well defined source [26] if amplitudes are of interest. Even so, eikonal models are widely used in many fields as approximation due to their simplicity [16, 15]. Additional to the approximation of wave propagation, the solution of the eikonal equation has many other applications in a large variety of fields [5, 26, 6, 4, 14, 36, 21, 31].

The proposed theory can be derived by using a new basis for the slowness or the velocity; however, the chosen basis for the velocity in this work is more illustrative. The proposed method generalizes various types of anisotropy and offers a simple derivation, implementation and application since the given tensor field at each model point is illustrative, and dealing with angles between semi-principal axes and coordinate system axes [38] can be avoided. There is a natural limitation of the method due to the definition of a metric space. All elements of the set have to have a well (uniquely) defined distance between them. Therefore, triplications can not be accounted for. However, triplications only occur in very rare cases under certain circumstances and are therefore seldom considered in practice [33].

The remainder of the paper is organized as follows. Firstly, the theory section gives an overview of the idea and the physical background. Starting with the dispersion relation of the acoustic wave equation, a new wave equation for tilted ellipsoidal anisotropy is derived and generalized for other types of anisotropy. The results section shows five examples to illustrate the functionality of the method, including solutions for homogeneous and inhomogeneous anisotropic velocity fields and field specific examples.

2 Theory

The theoretical treatment starts with the dispersion relation

$$\omega^2 = k_1^2 + k_2^2 + k_3^2, \tag{4.1}$$

of the acoustic wave equation in three dimensions

$$\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} = c^2 \nabla^2 u(\mathbf{x},t), \tag{4.2}$$

where ω is the angular frequency, c is the wave velocity, k_i is the wave number in the direction i, ∇^2 is the Laplacian operator and $u(\mathbf{x},t)$ is a scalar function. Equation (4.1) can be divided by ω^2 and represents a slowness surface

$$1 = |p_1|^2 + |p_2|^2 + |p_3|^2, \tag{4.3}$$

where $p_i = k_i/\omega$. The slowness surface in the form (4.3) represents a spherical wave front in the phase space with coordinates $p_1,\ p_2,\ p_3$ of an acoustic wave, traveling in homogeneous media with the wave velocity $v = 1\ m/s$ for a travel time of $T = 1\ s$ [7]. From this idea, various slowness surfaces can be constructed depending on the anisotropy one wants to model. The surfaces are in general not restricted to sixth-order polynomials like surfaces for waves in an elastic medium. Even though the method can be applied to a large number of surfaces, the focus in this work will be on velocity surfaces that can be described as a super-ellipsoid (Figure 4.2) in the form

$$1 = \left|\frac{x_1}{a}\right|^n + \left|\frac{x_2}{b}\right|^n + \left|\frac{x_3}{c}\right|^n, \tag{4.4}$$

since the resulting derivation of the corresponding wave equations are mathematically simpler and the numerical treatment is less complicated. In Equation (4.4), a,b and c represent the lengths of the semi-principal axes, which equal one in our case since the stretching is performed by the transformation into a new metric space. For $n = 2$ equation (4.4) represents a sphere and will build the basis for tilted ellipsoidal anisotropy. From the slowness surface the corresponding dispersion relation can be derived.

The procedure to derive an acoustic wave equation in anisotropic media will be described using the example of tilted ellipsoidal anisotropy. The starting point is the dispersion relation for an isotropic medium (4.1) which must be transformed into a new metric space for the velocity. The corresponding tensor describing a new basis is given as

$$\widehat{V}(\mathbf{x}) = \begin{pmatrix} \widehat{V}_{11}(\mathbf{x}) & \widehat{V}_{12}(\mathbf{x}) & \widehat{V}_{13}(\mathbf{x}) \\ \widehat{V}_{21}(\mathbf{x}) & \widehat{V}_{22}(\mathbf{x}) & \widehat{V}_{23}(\mathbf{x}) \\ \widehat{V}_{31}(\mathbf{x}) & \widehat{V}_{32}(\mathbf{x}) & \widehat{V}_{33}(\mathbf{x}) \end{pmatrix}, \tag{4.5}$$

where $\widehat{V}_{i1},\ \widehat{V}_{i2},\ \widehat{V}_{i3}$ are orthogonal vectors of the new basis. The tensor $\widehat{V}$ describes a possibly spatially dependent basis and leads to a new group velocity surface at each point in space, like the standard basis for the Euclidean space gives the group (and phase) velocity surface for isotropic wave propagation; therefore, it will be referred to as velocity tensor in the course of the paper. The velocity tensor defines a metric space at each model point. The corresponding metric space shall be called the velocity space. The velocity

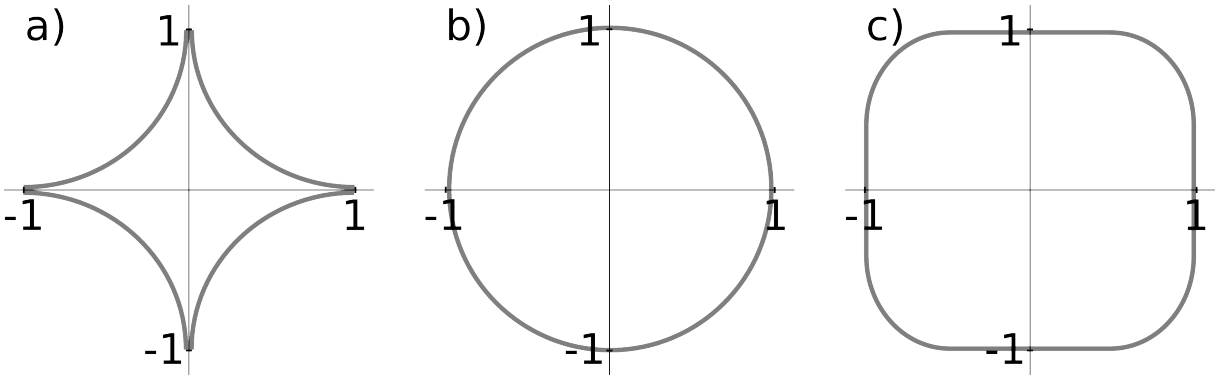


Figure 4.2: The figure shows different shapes of super-ellipses in the form $1 = x^n + y^m$. Super-ellipses can be used to describe slowness or velocity surfaces in the phase space with coordinates $p_1,\ p_2,\ p_3$ or $v_1,\ v_2,\ v_3$ respectively. Shape of the super-ellipse with a) $n = m = \frac{1}{2}$, b) $n = m = 2$, c) $n = m = 4$. Note that for non-integer exponents the procedure would lead to fractional derivatives in the wave equation which are numerically more difficult to handle. Super-ellipse surfaces are shown here as simple examples for possible velocity or slowness surfaces. However, the method is applicable to various surface shapes.

tensor must not be confused with the velocity itself which can be set to $1\ m/s$ everywhere. The dispersion relation (4.1) can also be transformed to a new basis $\widehat{S}$ by applying

$$\widehat{S}^{-1}\mathbf{k} = \begin{pmatrix} \widehat{S}_{11}^{-1}k_1 + \widehat{S}_{12}^{-1}k_2 + \widehat{S}_{13}^{-1}k_3 \\ \widehat{S}_{21}^{-1}k_1 + \widehat{S}_{22}^{-1}k_2 + \widehat{S}_{23}^{-1}k_3 \\ \widehat{S}_{31}^{-1}k_1 + \widehat{S}_{32}^{-1}k_2 + \widehat{S}_{33}^{-1}k_3 \end{pmatrix}, \tag{4.6}$$

where $\widehat{S}$ is a tensor for the new basis that yields the slowness surface and will be referred to as slowness tensor in the course of the paper. The slowness tensor describes a corresponding metric space that shall be called the slowness space. The translation of the velocity space into the slowness space can, for ellipsoidal anisotropy, be approximated by preserving the direction of each basis vector and inverting its length and is given by

$$\widehat{S}_{ij} = \frac{\widehat{V}_{ij}}{\widehat{V}_{1j}^2 + \widehat{V}_{2j}^2 + \widehat{V}_{3j}^2}. \tag{4.7}$$

The slowness space can tilt and stretch the slowness surface just like the velocity space $\widehat{V}$ can stretch and tilt the velocity surface. The components of the vector of equation (4.6) are the k_is in the new metric space. Therefore, inserting the vector components (4.6) in the dispersion relation (4.1) yields

$$\begin{aligned} \omega^2 =& (\widehat{S}_{11}^{-1}k_1 + \widehat{S}_{12}^{-1}k_2 + \widehat{S}_{13}^{-1}k_3)^2 + \\ & (\widehat{S}_{21}^{-1}k_1 + \widehat{S}_{22}^{-1}k_2 + \widehat{S}_{23}^{-1}k_3)^2 + \\ & (\widehat{S}_{31}^{-1}k_1 + \widehat{S}_{32}^{-1}k_2 + \widehat{S}_{33}^{-1}k_3)^2. \end{aligned} \tag{4.8}$$

Multiplying both sides of equation (4.8) with the wave field in the Fourier domain $u(\mathbf{k}, \omega)$ and performing an inverse Fourier transformation ($k_i \rightarrow -j\frac{\partial}{\partial x_i},\ \omega \rightarrow j\frac{\partial}{\partial t}$), where $j = \sqrt{-1}$, leads to the acoustic wave

equation for tilted ellipsoidal anisotropy

$$\begin{aligned}\frac{\partial^2 u(\mathbf{x},t)}{\partial t^2} = & \\ & \frac{\partial^2 u(\mathbf{x},t)}{\partial x_1^2}(\widehat{S}_{11}^{-1}+\widehat{S}_{21}^{-1}+\widehat{S}_{31}^{-1})+ \\ & \frac{\partial^2 u(\mathbf{x},t)}{\partial x_2^2}(\widehat{S}_{12}^{-1}+\widehat{S}_{22}^{-1}+\widehat{S}_{32}^{-1})+ \\ & \frac{\partial^2 u(\mathbf{x},t)}{\partial x_3^2}(\widehat{S}_{13}^{-1}+\widehat{S}_{23}^{-1}+\widehat{S}_{33}^{-1})+ \\ & 2\frac{\partial^2 u(\mathbf{x},t)}{\partial x_1 \partial x_2}(\widehat{S}_{11}^{-1}\widehat{S}_{12}^{-1}+\widehat{S}_{21}^{-1}\widehat{S}_{22}^{-1}+\widehat{S}_{31}^{-1}\widehat{S}_{32}^{-1})+ \\ & 2\frac{\partial^2 u(\mathbf{x},t)}{\partial x_1 \partial x_3}(\widehat{S}_{11}^{-1}\widehat{S}_{13}^{-1}+\widehat{S}_{21}^{-1}\widehat{S}_{23}^{-1}+\widehat{S}_{31}^{-1}\widehat{S}_{33}^{-1})+ \\ & 2\frac{\partial^2 u(\mathbf{x},t)}{\partial x_2 \partial x_3}(\widehat{S}_{12}^{-1}\widehat{S}_{13}^{-1}+\widehat{S}_{22}^{-1}\widehat{S}_{23}^{-1}+\widehat{S}_{32}^{-1}\widehat{S}_{33}^{-1}). \end{aligned} \tag{4.9}$$

Since this derivation leads to a wave equation that is only valid at one model point $\mathbf{x}$, $\widehat{S}$ in equation (4.8) can be treated as a spatial constant. Equation (4.9) represents the acoustic wave equation for tilted ellipsoidal anisotropy. It can also be seen as an acoustic wave equation describing a wave traveling in isotropic media in a given metric space. The two formulations illustrate the duality of anisotropy and metric spaces. The presented procedure is similar for any other chosen velocity or slowness surface and therefore for many types of anisotropy.

For illustrative reasons, an alternative approach is shown to derive equation (4.9). The same result can be obtained by using the acoustic wave equation (4.2) directly and transforming the differential operator $(\partial/\partial x_i)$ into the slowness space. This approach leads to

$$\widehat{S}^{-1}\begin{pmatrix}\frac{\partial}{\partial x}\\ \frac{\partial}{\partial y}\\ \frac{\partial}{\partial z}\end{pmatrix} = \begin{pmatrix}\widehat{S}_{11}^{-1}\frac{\partial}{\partial x}+\widehat{S}_{12}^{-1}\frac{\partial}{\partial y}+\widehat{S}_{13}^{-1}\frac{\partial}{\partial z}\\ \widehat{S}_{21}^{-1}\frac{\partial}{\partial x}+\widehat{S}_{22}^{-1}\frac{\partial}{\partial y}+\widehat{S}_{23}^{-1}\frac{\partial}{\partial z}\\ \widehat{S}_{31}^{-1}\frac{\partial}{\partial x}+\widehat{S}_{32}^{-1}\frac{\partial}{\partial y}+\widehat{S}_{33}^{-1}\frac{\partial}{\partial z}\end{pmatrix}. \tag{4.10}$$

Inserting the new differential operators (4.10) in the acoustic wave equation (4.2) leads to the same result as equation (4.9). This equivalent approach shows that the only difference between the isotropic and the tilted ellipsoidal anisotropic case is the underlying metric space.

From equation (4.9) the eikonal equation

$$\begin{aligned}1 = & (\frac{\partial T(\mathbf{x})}{\partial x_1})^2(\widehat{S}_{11}^{-1}+\widehat{S}_{21}^{-1}+\widehat{S}_{31}^{-1})+ \\ & (\frac{\partial T(\mathbf{x})}{\partial x_2})^2(\widehat{S}_{12}^{-1}+\widehat{S}_{22}^{-1}+\widehat{S}_{32}^{-1})+ \\ & (\frac{\partial T(\mathbf{x})}{\partial x_1})^2(\widehat{S}_{13}^{-1}+\widehat{S}_{23}^{-1}+\widehat{S}_{33}^{-1})+ \\ & 2\frac{\partial T(\mathbf{x})}{\partial x_1}\frac{\partial T(\mathbf{x})}{\partial x_2}(\widehat{S}_{11}^{-1}\widehat{S}_{12}^{-1}+\widehat{S}_{21}^{-1}\widehat{S}_{22}^{-1}+\widehat{S}_{31}^{-1}\widehat{S}_{32}^{-1})+ \\ & 2\frac{\partial T(\mathbf{x})}{\partial x_1}\frac{\partial T(\mathbf{x})}{\partial x_3}(\widehat{S}_{11}^{-1}\widehat{S}_{13}^{-1}+\widehat{S}_{21}^{-1}\widehat{S}_{23}^{-1}+\widehat{S}_{31}^{-1}\widehat{S}_{33}^{-1})+ \\ & 2\frac{\partial T(\mathbf{x})}{\partial x_2}\frac{\partial T(\mathbf{x})}{\partial x_3}(\widehat{S}_{12}^{-1}\widehat{S}_{13}^{-1}+\widehat{S}_{22}^{-1}\widehat{S}_{23}^{-1}+\widehat{S}_{32}^{-1}\widehat{S}_{33}^{-1}) \end{aligned} \tag{4.11}$$

can be derived. Equation (4.11) can be used to compute the travel times of a wave front propagating in media with tilted ellipsoidal anisotropy.

The translation of a velocity surface into the corresponding slowness surface is a non-trivial problem since the actual slowness surface is created by inverting the radii in all directions and can no longer be described by taking the polynomial surface for the velocity and transforming it into the slowness space. For better understanding, we can have a look at ellipsoidal anisotropy. In the case of ellipsoidal anisotropy, the ellipsoid with semi-principal axes $a,\ b,\ c$ describing the velocity surface leads to an ellipsoid with semi-principal axes $1/a,\ 1/b,\ 1/c$ describing the slowness surface even though this is only correct along the axes. It is in general not the case that the actual slowness surface which is obtained by inverting the radii of the velocity surface in all directions, resembles the slowness surface that is obtained by inverting the length of the axes. This approximation was used to preserve the simplicity of the method and leads to inaccuracies in the space between the axes. The issue seems less problematic if the velocity and slowness surfaces are considered to be approximations in practice and the real surfaces are unknown. Therefore, in the case of ellipsoidal anisotropy, the errors made by inverting only the radii in axes direction is smaller than the error made by approximating anisotropy as a known surface. For other surfaces the induced error can be larger. Another way to work around the problem is to give the slowness surface and the slowness tensor in the first step of the solution process, thereby omitting the need to translate between velocity and slowness. In this work, the velocity is chosen as a starting point for illustrative reasons. The issue of translating between velocity and slowness surfaces will be addressed in a more descriptive way in the next sections.

Using the procedure described above, a wave equation can be given for any super-ellipsoidal slowness surface

$$\begin{aligned}\frac{\partial^\phi u(\mathbf{x},t)}{\partial t^\phi} = \mathcal{F}^{-1}\Big[\Big(&|\widehat{S}_{11}^{-1}k_1 + \widehat{S}_{12}^{-1}k_2 + \widehat{S}_{13}^{-1}k_3|^\phi \\ &+|\widehat{S}_{21}^{-1}k_1 + \widehat{S}_{22}^{-1}k_2 + \widehat{S}_{23}^{-1}k_3|^\phi \\ &+|\widehat{S}_{31}^{-1}k_1 + \widehat{S}_{32}^{-1}k_2 + \widehat{S}_{33}^{-1}k_3|^\phi\Big)\Big]u(\mathbf{x},t),\end{aligned} \tag{4.12}$$

with the associated eikonal equation

$$\begin{aligned}1 = \Big(&|\widehat{S}_{11}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{12}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{13}^{-1}\frac{\partial T}{\partial x_3}|^\phi \\ &+|\widehat{S}_{21}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{22}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{23}^{-1}\frac{\partial T}{\partial x_3}|^\phi \\ &+|\widehat{S}_{31}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{32}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{33}^{-1}\frac{\partial T}{\partial x_3}|^\phi\Big)^{\frac{1}{\phi}},\end{aligned} \tag{4.13}$$

where ϕ is the exponent describing the shape of the super-ellipsoid. Equation (4.12) has a simple solution for all $\phi \in \mathbb{N}$. The general form of the eikonal equation (4.13) is used later to compute the wave fronts in sandstone.

The procedure described in this section could also be reformulated to extract the metric tensor g_{ij} on a Riemannian manifold. For that, the metric tensor g_{ij} in the basis formed by normalizing the vectors in the slowness tensor is given by filling its diagonal with the slowness values in axes direction.

3 Numerical Experiments and Results

Five experiments are presented in this section. If not mentioned explicitly, the experiments were executed using a grid of size $192 \times 192 \times 192$ and a spacing of $dx = dy = dz = 0.7\ meter$. The first experiment shows the solution of the wave equation (4.9) for an isotropic velocity field. The solution was compared to an analytical solution of the eikonal equation to verify the validity of the proposed method. Next, a solution for a homogeneous velocity field with anisotropy is presented. This example can be motivated by the desire to approximate wave propagation through a homogeneously moving medium. The following example shows the result for an anisotropic and inhomogeneous velocity field as it could appear in simple real-life applications, motivated by an electric wave propagating through the heart muscles. The anisotropy is induced by the muscle fiber direction. For Experiment 4, we chose a velocity model that approaches real-life complexity as it comprises sharp velocity contrasts as found in many applications, especially in seismology. The last example shows the functionality of the method in media that shows a vertical transverse isotropy. This experiment is motivated by wave-motion modeling, executed in the scope of seismology.

3.1 The Isotropic Homogeneous Velocity Field

For the first experiment, we are assuming the case of an isotropic homogeneous velocity field. The first velocity tensor is given at every point in the model space by

$$\widehat{V} = \widehat{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.14}$$

Equation (4.14) represents the standard basis of the Euclidean space. Therefore, the velocity and slowness surfaces are spheres and the modeled velocity field is isotropic and homogeneous. Figure 4.3 shows the solution of the computation of equation (4.9). For proof of accuracy and correctness of equation (4.9), the analytic solution of the eikonal equation for isotropic media is included in Figure 4.3. For the given metric, the derived wave and eikonal equations could also be directly simplified to the equations for the isotropic and homogeneous case.

3.2 A Homogeneous Anisotropic Velocity Field

For Experiment 2, we are investigating a homogeneous, anisotropic velocity field. Now, the metric space is constant in the entire model and is given by the tensor

$$\widehat{V} = \begin{pmatrix} 1 & -2.0 & 0 \\ 1 & 2.0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.15}$$

For illustrative purposes, the basis is shown with respect to the standard basis in Figure 4.4. The corresponding velocity surface can be obtained by applying

$$\widehat{V}^{-1}\mathbf{v} = \begin{pmatrix} 0.5v_1 + 0.5v_2 \\ -0.25v_1 + 0.25v_2 \\ v_3 \end{pmatrix}, \tag{4.16}$$

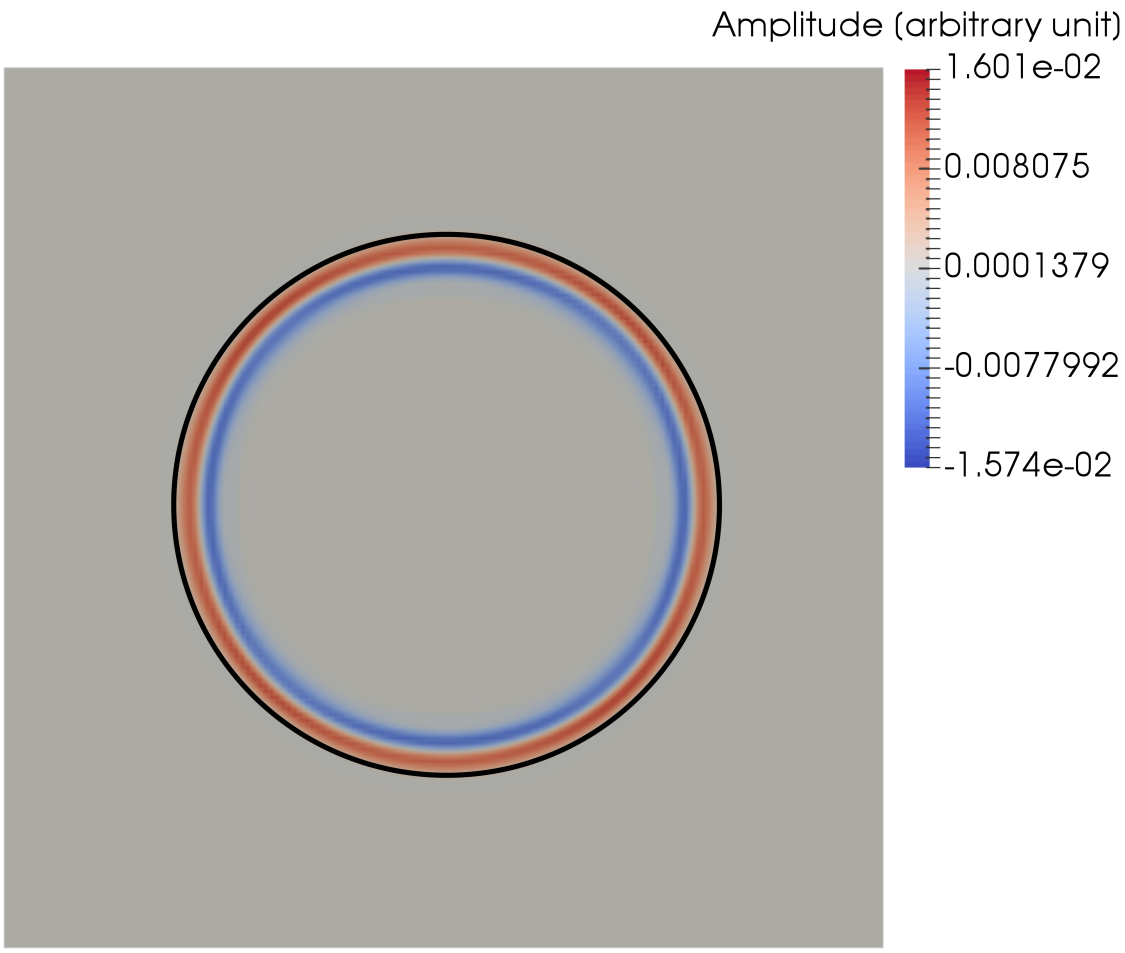


Figure 4.3: A two-dimensional slice of the three-dimensional wave field for the homogeneous velocity field. The wave front of the solution is compared to the analytical solution of the eikonal equation (black). The analytic solution of the isotropic eikonal equation aligns with the wave front of the solution of the wave equation (4.9).

where the v_is are the velocity components with respect to the original basis. Inserting the vector components of equation (4.16) in the velocity surface for isotropic wave propagation leads to

$$1 = (0.5v_1 + 0.5v_2)^2 + (-0.25v_1 + 0.25v_2)^2 + v_3^2. \tag{4.17}$$

An approximation of the basis describing the slowness space can be obtained, as described in the theory section, by inverting the length of the basis vectors of the velocity tensor while maintaining their directions. The slowness surface can then be obtained by multiplying the inverse of the slowness tensor by the wave number $\mathbf{k}$. For other forms of velocity surfaces the slowness surface is potentially much more difficult to find. This issue can be avoided by using the slowness surface in the solution process instead of the velocity surface. For this example, the velocity surface is chosen for illustrative reasons. However, the resulting slowness surface in this case is given by

$$1 = (p_1 + p_2)^2 + (-2p_1 + 2p_2)^2 + p_3^2. \tag{4.18}$$

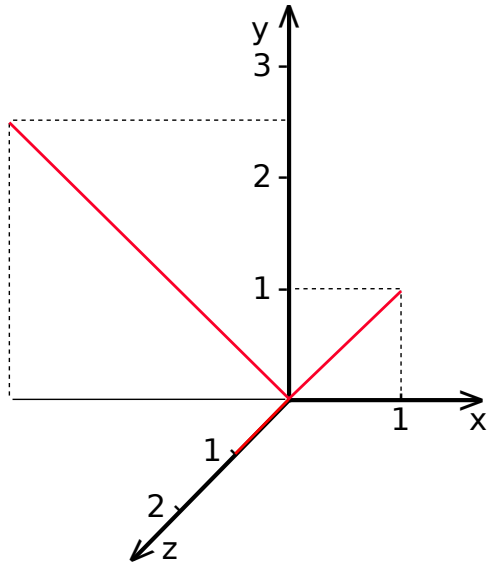


Figure 4.4: The basis of the new metric space (4.15) (in red) is shown with respect to the standard basis (black), for illustrative purposes. Note that the longest axis of the new metric is also the longest semi-principal axis of the ellipsoidal-shaped wave field shown in Figure 4.5.

In the case of ellipsoidal anisotropy, the required inverse of the slowness tensor is the transpose of the velocity tensor used to describe the basis for the velocity. This circumstance leads to faster computations since the creation of the slowness space and the inversion of the tensor can be omitted. After inserting $p_i = k_i/\omega$, multiplying by a function in the Fourier domain and an inverse Fourier transformation, as described in the theory section, equation (4.18) leads to the wave equation (4.9) for the given metric (4.15). A snapshot of the moving wave is shown in Figure 4.5. The results of this experiment could, for example, be applied to approximate wave propagation in a homogeneously moving medium.

3.3 A Wave in Inhomogeneous Anisotropic Media

Wave propagation in inhomogeneous anisotropic media is the most important example for real-life applications and is investigated in Experiment 3. The given tensor depends on the position in the modeled space. The tensor field representing the basis and defining the metric, and therefore, the velocity anisotropy, is represented by its respective longest vector $\widehat{V}_{i1}$ in Figure 4.6 together with the corresponding wave field. The velocity tensor is given by

$$\widehat{V} = \begin{pmatrix} \frac{-3x_1}{\sqrt{x_1^2+x_2^2}} & \frac{x_2}{\sqrt{x_1^2+x_2^2}} & 0 \\ \frac{3x_2}{\sqrt{x_1^2+x_2^2}} & \frac{x_1}{\sqrt{x_1^2+x_2^2}} & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{4.19}$$

The axes $\widehat{V}_{i2}$ and $\widehat{V}_{i3}$ of the given basis are pairwise perpendicular to $\widehat{V}_{i1}$, and one third of the length of $\widehat{V}_{i1}$. A wave of this kind can be found in inhomogeneously moving media or in organs like the muscle tissue of the heart. In this case, the muscle fiber direction is responsible for the anisotropy.

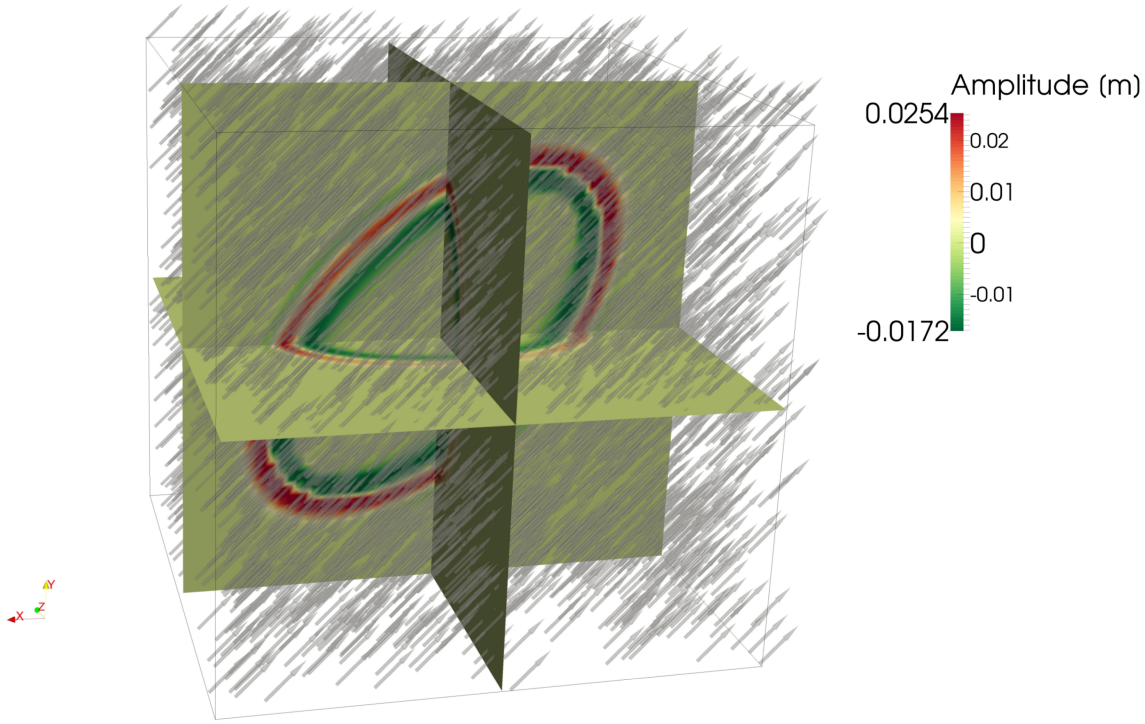


Figure 4.5: Three two-dimensional slices of the three-dimensional wave field of a homogeneous velocity field in the metric shown in Figure 4.4. The arrows show the longest axis $\widehat{V}_{i2}$ of the velocity metric. The wave front represents a circle in the metric given by (4.15). Note, that the wave front resembles an ellipsoid in the Euclidean metric with the semi-principal axes given by the tensor (4.15).

3.4 Wave Propagation Through a Geological Subsurface

Most real-life applications of the proposed method will involve wave propagation through complex media. It is therefore important to test the method regarding its behavior when dealing with sharp velocity contrasts. One particular complex example of this kind is wave propagation through the geological subsurface and is the focus of Experiment 4. The velocity field in Figure 4.7 is defined on a grid of 128^3 nodes and is given by

$$\widehat{V} = \begin{cases} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 20 & 0 \\ 50 & 0 & 0 \end{pmatrix} & \text{if} \sqrt{x_1^2 + x_2^2} < 80 \\ \begin{pmatrix} \frac{-50x_1}{\sqrt{x_1^2+x_2^2}} & \frac{x_2}{\sqrt{x_1^2+x_2^2}} & 0 \\ \frac{50x_2}{\sqrt{x_1^2+x_2^2}} & \frac{x_1}{\sqrt{x_1^2+x_2^2}} & 0 \\ 0 & 100 & -1 \end{pmatrix} & \text{if} \sqrt{x_1^2 + x_2^2} < 80 \wedge x_3 > 80 \\ \begin{pmatrix} \frac{x_1 x_3}{\sqrt{x_1^2+x_1^2}} & \frac{x_1 x_2}{\sqrt{x_1^2+x_1}} & 0 \\ \frac{-x_1 x_3}{\sqrt{x_1^2+x_1^2}} & \frac{x_1 x_2}{\sqrt{x_1^2+x_1^2}} & \frac{1}{x_1} \\ \frac{x_3}{\sqrt{x_1^2+x_2^2}} & 0 & 30 \end{pmatrix} & \text{else.} \end{cases} \tag{4.20}$$

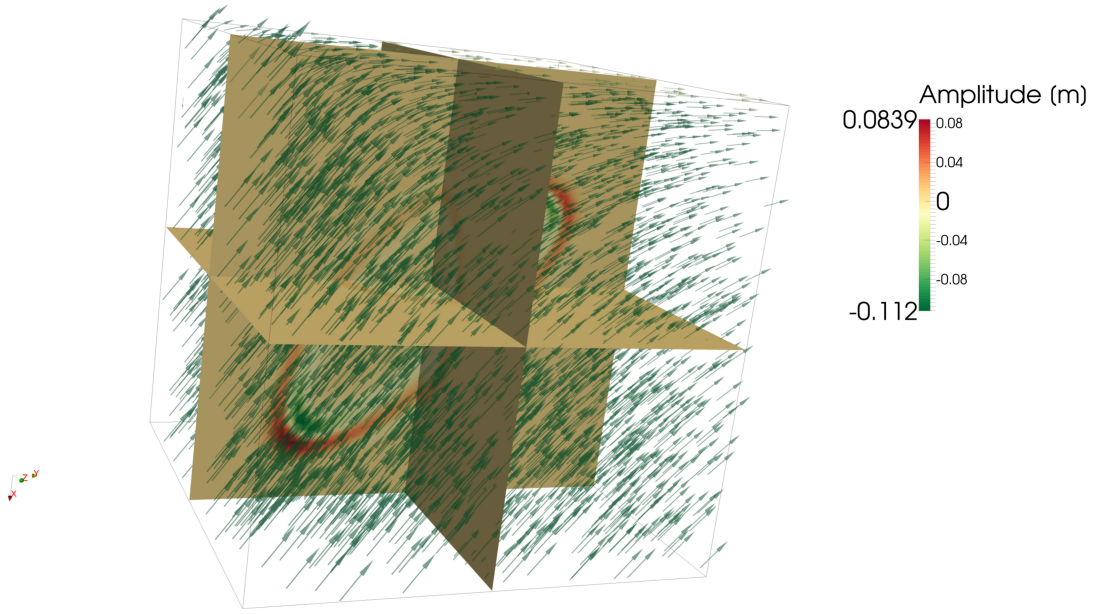


Figure 4.6: Three two-dimensional slices of the three-dimensional tensor field describing the velocity space and the corresponding wave field. The vectors are showing the longest axis $\widehat{V}_{i1}$ of the given basis (4.19). $\widehat{V}_{i2}$ and $\widehat{V}_{i3}$ are perpendicular to the shown vector in each point and one third in length. Note, that the wave seems to follow a preferred direction given in each point in space by the tensor (4.19).

The velocity model in Figure 4.7 comprises two layers with different preferred propagation directions and a body whose preferred propagation direction is perpendicular to the ones of the two layers. The model contains sharp interfaces and strong velocity variations. Snapshots of the three-dimensional wave field are shown in Figure 4.8.

3.5 An Eikonal Equation for S-Wave Propagation in Sandstone

Slowness or velocity surfaces in real materials are often not elliptical. To verify the functionality of the method for wave propagation in materials showing other velocity-surface shapes, Experiment 5 presents a result for sandstone. Sandstone is typically considered to have vertical transverse isotropy (VTI). This name describes a medium whose parameters are invariant regarding a rotation around the z-axis [28]. As represented in Figure 4.9, the super-ellipsoid $|x_1|^{\frac{3}{2}} + |x_2|^{\frac{3}{2}} + |x_3|^{\frac{3}{2}} = 1$ describes the velocity surface of the s-wave with reasonable accuracy considering a certain simplicity which shall be maintained. From Figure 4.9, the following slowness surface can be derived

$$\omega^{\frac{3}{2}} = k_1^{\frac{3}{2}} + k_2^{\frac{3}{2}} + k_3^{\frac{3}{2}}. \tag{4.21}$$

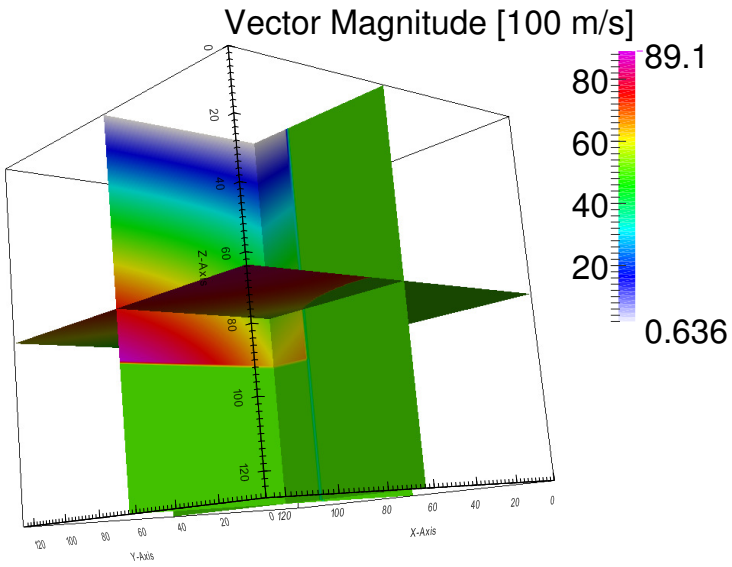


Figure 4.7: Velocity field comprising sharp velocity contrasts. The colors indicate the magnitude of the longest vector in the tensor describing the underlying metric space. Note, that the color, in this illustration, gives no information about the direction of the longest vector of the tensor. The directions are given in equation (4.20).

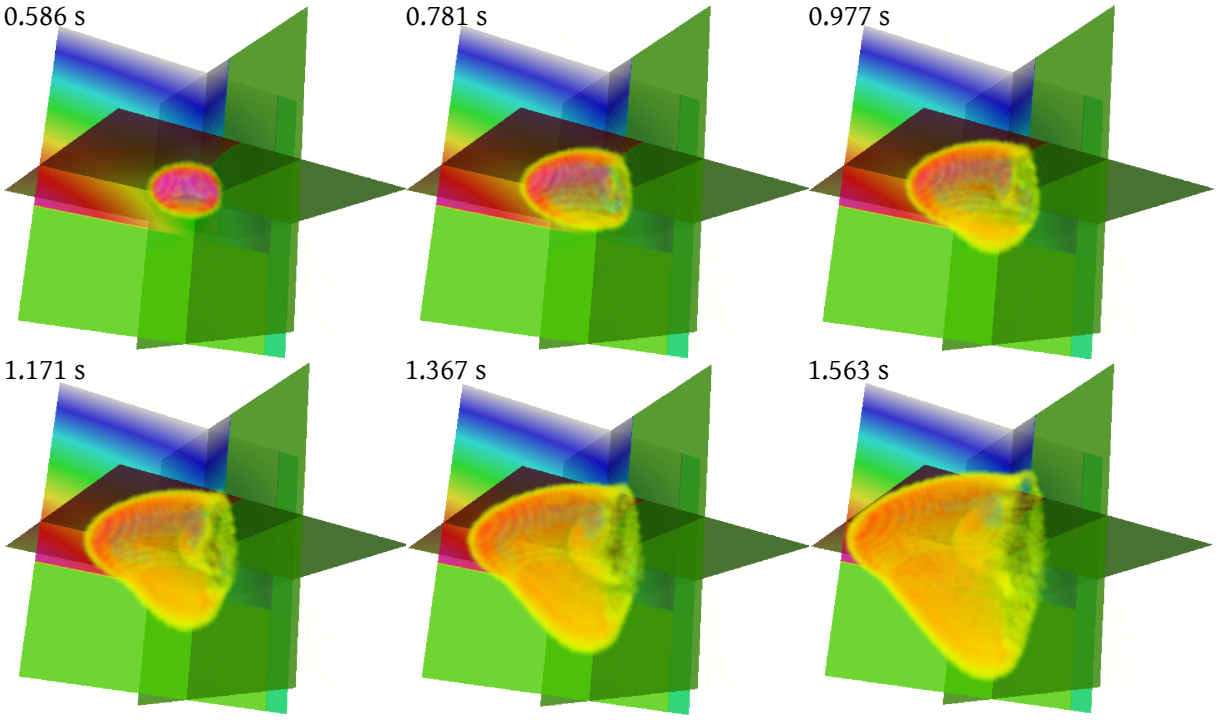


Figure 4.8: Snapshot of the wave field after indicated times. Note the behavior of the solution at interfaces between materials with different preferred directions of wave propagation. Note, that no artifacts emerge in the solution.

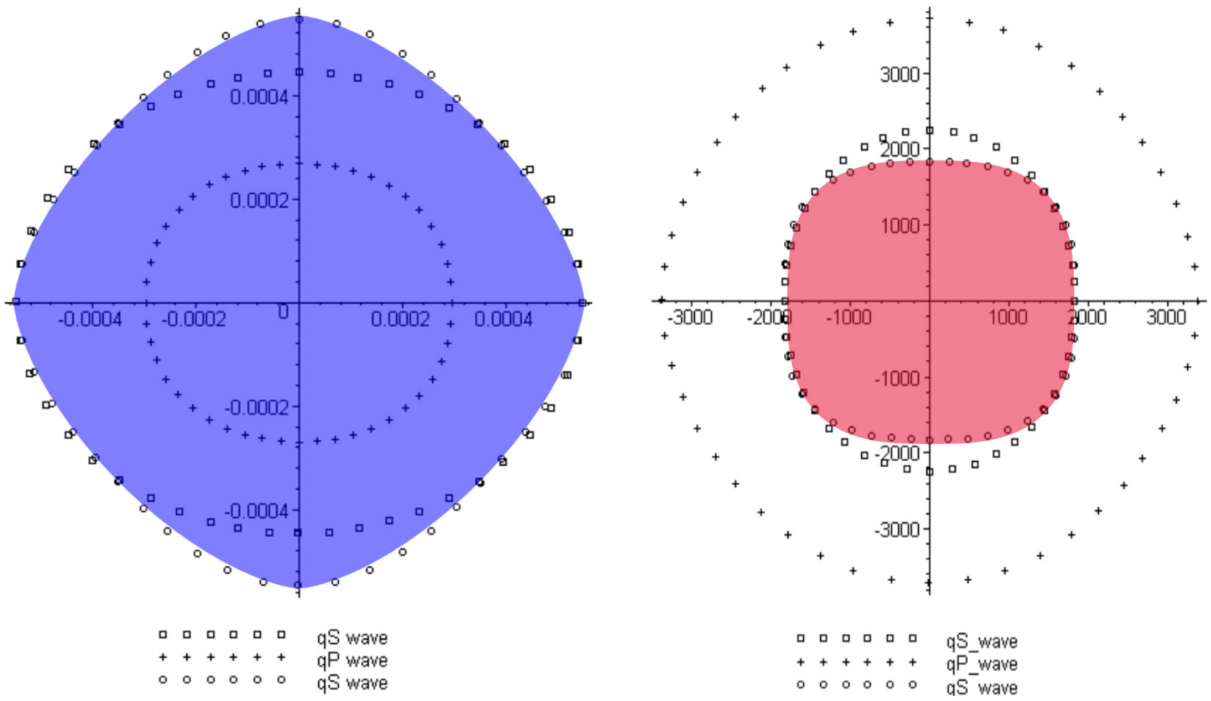


Figure 4.9: The velocity and slowness surfaces of different wave modes in sandstone. The super-ellipsoid $x_1^{\frac{3}{2}} + x_2^{\frac{3}{2}} + x_3^{\frac{3}{2}} = 1$ (in blue) corresponds to the s-wave slowness surface. The solution resembles the shape marked in red. Figure modified from Piedrahita et al. [28].

Using the proposed method, the following eikonal equation can be derived

$$
\begin{aligned}
1 = \Bigg(& |\widehat{S}_{11}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{12}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{13}^{-1}\frac{\partial T}{\partial x_3}|^{\frac{3}{2}} \\
& + |\widehat{S}_{21}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{22}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{23}^{-1}\frac{\partial T}{\partial x_3}|^{\frac{3}{2}} \\
& + |\widehat{S}_{31}^{-1}\frac{\partial T}{\partial x_1} + \widehat{S}_{32}^{-1}\frac{\partial T}{\partial x_2} + \widehat{S}_{33}^{-1}\frac{\partial T}{\partial x_3}|^{\frac{3}{2}} \Bigg)^{\frac{2}{3}}.
\end{aligned} \tag{4.22}
$$

The solution of equation (4.22) is presented in Figure 4.10. For simplicity, the slowness tensor is spatially independent and not tilted. The eikonal equation (4.22) and the associated wave equation are valid for tilted and inhomogeneous sandstone.

4 Discussion

The results showed the accuracy and the functionality of the method for anisotropic, homogeneous and inhomogeneous velocity fields. The comparison with an analytical solution of the eikonal equation proved that the wave front of the solution resembled the analytical wave front. Figure 4.3 showed that the analytical solution of the eikonal equation aligns with the solution of the derived wave equation (4.9).

The result of the second experiment (Figure 4.5) showed the solution of equation (4.9) for a homogeneous anisotropic velocity field, approximating, for example, a moving medium. The wave front of the resulting wave has the expected shape of an ellipsoid given by a transformed sphere into the new metric space (4.15).

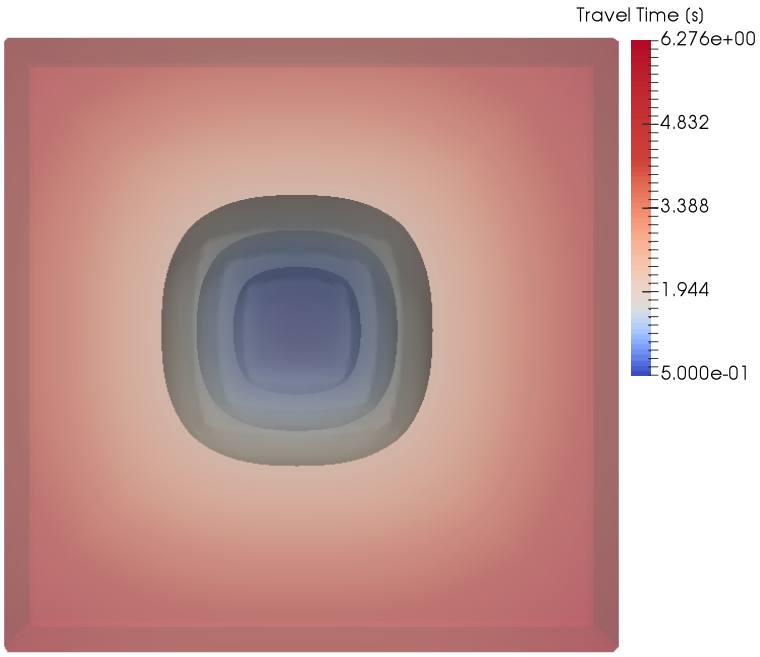


Figure 4.10: The solution of equation (4.22). Note that there is a high level of resemblance between the solution and the shape shown in Figure 4.9.

The result of the third experiment (Figure 4.6) showed the solution of equation (4.9) for an inhomogeneous anisotropic velocity. Such a velocity field can occur in nature, for instance, in muscle tissue or in inhomogeneously moving media. The resulting wave field showed that the wave follows the preferred propagation direction given in every point in space by the tensor (4.19).

Experiment 4 tested the method regarding the wave propagation through a synthetic, geological subsurface. It turned out, that the method handles sharp velocity contrasts in a stable manner. No artifacts appear in the solution as in Alkhalifah [1, 2]. This is due to the fact that, from a physical perspective, the method propagates acoustic waves in a homogeneous, isotropic medium; what changes is the underlying space. The last experiment showed the solution of an eikonal equation derived by the proposed method of a s-wave propagating through sandstone. In this case, sandstone exhibits vertical transverse isotropy. The resulting wave front (see Figure 4.10) resembles the expected wave front depicted in Figure 4.9. Again, artifacts do not appear in the solution as in Alkhalifah [2]. This result demonstrates the ability of the method to be applied to other slowness surfaces and therefore types of anisotropy apart from elliptical ones. However, the equations can become complex for non-integer exponents in the dispersion relation.

The problem of transforming the velocity surface into a slowness surface is not method-specific and can be avoided by dealing with slowness surfaces in the solution process in the first place, or by acknowledging that the accurate velocity and slowness surfaces are unknown in practice. The proposed derivation method for acoustic wave propagation problems offers a straight-forward derivation and

implementation. In cases where only travel times are important, the proposed method can be used to derive eikonal equations for various types of anisotropy. The proposed method can also be used to derive wave equations for any kind of velocity or slowness surface. Here, velocity surfaces in the form of a super-ellipsoid were discussed for simplicity. The corresponding wave and eikonal equations are simple to implement and the computation proved to be stable. The solution does not show any artifacts. The method therefore, has a large potential benefit for research and industry fields in which wave propagation in anisotropic media plays a significant role. Especially the fields of bio-computing, and seismology could benefit from the method. Wave propagation through metamaterials can be described and modeled in an efficient and simple manner. Also, acoustic wave propagation through moving media like waves traveling through water or air can be approximated in a straight-forward way.

We are claiming that the proposed method leads to a simpler derivation of the governing equations, a simple implementation and an efficient and stable computation. These assertions will be challenged in this paragraph. The simple derivation mainly stems from the fact, that we are interpreting changes of material parameters in certain directions as a change of the underlying metric space. The benefits are two fold. Firstly, we are only dealing with velocity surfaces instead of elastic parameters, which is very comprehensible. Secondly, we can use a clean description of velocity by using bases and norms, which allows for a simple derivation of the governing equations. A drawback is that we have to calculate a tensor representation of the velocity, in case we are dealing with elastic parameters only. The methods works best if the starting points are velocity surfaces. In this case, we can challenge the simplicity statement. We can compare the derivation of the eikonal equation by Alkhalifah [1] to the derivation of the eikonal equation (4.22) for sandstone in this paper. Equation (4.22) has a simpler structure. However, the complexity of an equation remains subject to personal preference. Another great example for the computation of wave phenomena in anisotropic media is the work of Joets and Ribotta [18]. The computation of the eikonal equation obtained by using the proposed method includes the computation of all ray directions and has a simpler form. However, the method described by Joets and Ribotta [18] is more general. An advantage of the proposed method is, that the equations will only change slightly for different kinds of anisotropy within certain limits, which are discussed in more depth later on. Again, if the starting point are elastic parameters, the derivation by Alkhalifah [1] is about as simple as the proposed approach. For a further evaluation of simplicity, we can have a look at Cervený et al. [7]. The eikonal equation for anisotropy is derived by using the eigenvalues of the Christoffel matrix, which is not a simple concept compared with dealing with changes in the metric and basis transformation. The simple implementation comes from the fact that the algorithm is basically a solver of the acoustic, isotropic wave or eikonal equation in homogeneous media. The only additional work goes into changing exponents and inverting simple matrices. The close relation to an acoustic solver for homogeneous, isotropic media is also the reason for the stability and efficiency of the computation. Here, we have to address another limitation. The proposed method can lead to fractal derivatives in the wave equation which can compromise the computational efficiency. The inclusion of boundary conditions, using the proposed approach, is simple and follows the procedure for the derivation of wave equations. The computational efficiency stems mainly from the fact that the additional computations, namely an inversion of a 3 matrix can be done efficiently on GPU cores, which is the preferred architecture for wave-motion simulations. Therefore, the computational efficiency of the solution of the eikonal equation does not depend on the kind of anisotropy within our defined set of anisotropies. In general, it can be said, that the method's strength is repeatability of derivations and

implementations.

The limitations of the method can be clearly formulated. The method is, by construction an approximation. However, depending on the allowed complexity, this approximation can induce smaller errors than, for example, the approximation of the real medium as a model given some information. Also, the method breaks down as soon as triplications occur in the slowness surface. In this work, the slowness surface had to be in the form of a super-ellipsoid. However, this is just a limitation of derivation, not a basis limitation of the method, since it is, in theory, possible to extend the derivation to more general surfaces.

In this paper, the focus was on super-ellipsoidal surfaces and, in particular, ellipsoidal and vertical transverse isotropy, which was approximated by the proposed method. In future work, other kinds of surfaces could be investigated with respect to complexity and computational feasibility. Special interest lies on velocity surfaces described by super-ellipsoids with exponents that are not elements of the natural numbers and higher order surfaces. Also, more general shapes, like spherical harmonics, could be used to derive wave equations for complex types of anisotropy. The proposed theory could, because of its simplicity of derivation and application, build a new basis for the investigation of acoustic wave propagation in anisotropic media.

5 Conclusion

A new method for deriving wave and eikonal equations for acoustic wave propagation in anisotropic media was presented and validated by experiments. The proposed theory generalizes various types of anisotropy by narrowing the procedure down to the selection of a slowness or velocity surface, and a tensor field defining a new metric space at each spatial model point, thereby simplifying the derivation of the governing equation. Since all the changes are with regard to the underlying space, the numerical computations are as stable as the computations in isotropic media. No artifacts can be seen in the solutions as in Alkhalifah [1, 2]. In this work, we covered surfaces which can be described as a super-ellipsoid. A greater variety of slowness surfaces will be addressed in future work.

Acknowledgements

The presented work was funded by Kalkulo AS and the Research Council of Norway under grant 238346. The work has been conducted at Kalkulo AS, a subsidiary of Simula Research Laboratory. Are Magnus Bruaset for beneficial comments and support. We also want to acknowledge Qiang Lan and Kristin McLeod for their help to make the paper more comprehensible.

Bibliography

[1] Tariq Alkhalifah. An acoustic wave equation for anisotropic media. *Geophysics*, 65(4):1239–1250, 2000. doi: 10.1190/1.1444815.

[2] Tariq Alkhalifah. An acoustic wave equation for orthorhombic anisotropy. *Geophysics*, 68(4): 1169–1172, 2003. doi: 10.1190/1.1598109.

[3] Umair bin Waheed, Tariq Alkhalifah, and Hui Wang. Efficient traveltime solutions of the acoustic ti eikonal equation. *Journal of Computational Physics*, 282:62–76, 2015. doi: http://dx.doi.org/10.1016/j.jcp.2014.11.006.

[4] AV Borovskikh. Eikonal equations for an inhomogeneous anisotropic medium. *Journal of Mathematical Sciences*, 164(6):859–880, 2010. doi: 10.1007/s10958-010-9770-y.

[5] Anna R Bruss. The eikonal equation: Some results applicable to computer vision. *Journal of Mathematical Physics*, 23(5):890–896, 1982. doi: http://dx.doi.org/10.1063/1.525441.

[6] Frederick W Byron and Charles J Joachain. Eikonal theory of electron-and positron-atom collisions. *Physics Reports*, 34(4):233–324, 1977. doi: http://dx.doi.org/10.1016/0370-1573(77)90014-X.

[7] V. Cervený, I.-A Molotkov, and I. Pšenčík. *Ray Method in Seismology*. Univerzita Karlova Press, 1977.

[8] Huanyang Chen and C. T. Chan. Acoustic cloaking in three dimensions using acoustic metamaterials. *Applied Physics Letters*, 91(18):183518, October 2007. ISSN 0003-6951, 1077-3118. doi: http://dx.doi.org/10.1063/1.2803315.

[9] Yongyao Chen, Haijun Liu, Michael Reilly, Hyungdae Bae, and Miao Yu. Enhanced acoustic sensing through wave compression and pressure amplification in anisotropic metamaterials. *Nature Communications*, 5:5247, October 2014. ISSN 2041-1723. doi: 10.1038/ncomms6247.

[10] Steven A. Cummer and David Schurig. One path to acoustic cloaking. *New Journal of Physics*, 9(3): 45, 2007. ISSN 1367-2630. doi: http://dx.doi.org/10.1088/1367-2630/9/3/045.

[11] Joe A Dellinger. *Anisotropic Seismic Wave Propagation*. PhD thesis, Stanford University, 1991.

[12] Romain Fleury, Dimitrios L. Sounas, Caleb F. Sieck, Michael R. Haberman, and Andrea Alù. Sound Isolation and Giant Linear Nonreciprocity in a Compact Acoustic Circulator. *Science*, 343(6170): 516–519, January 2014. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1246957.

[13] Lee Ren Fok. Anisotropic and Negative Acoustic Index Metamaterials. 2010.

[14] T. Gillberg. *Fast and Accurate Front Propagation for Simulation of Geological Folds*. PhD thesis, Falculty of mathematics and natural sciences, University of Oslo, 2013.

[15] Pierre Gouédard, Huajian Yao, Fabian Ernst, and Robert D van der Hilst. Surface wave eikonal tomography in heterogeneous media using exploration data. *Geophysical Journal International*, 191 (2):781–788, 2012. doi: https://doi.org/10.1111/j.1365-246X.2012.05652.x.

[16] Samuel H Gray and William P May. Kirchhoff migration using eikonal equation traveltimes. *Geophysics*, 59(5):810–817, 1994. doi: http://dx.doi.org/10.1190/1.1443639.

[17] Klaus Helbig and Leon Thomsen. 75-plus years of anisotropy in exploration and reservoir seismics: A historical review of concepts and methods. *Geophysics*, 70(6):9ND–23ND, November 2005. ISSN 0016-8033, 1942-2156. doi: 10.1190/1.2122407.

[18] A Joets and R Ribotta. A geometrical model for the propagation of rays in an anisotropic inhomogeneous medium. *Optics communications*, 107(3-4):200–204, 1994.

[19] A Joets and R Ribotta. A geometrical model for the propagation of rays in an anisotropic inhomogeneous medium. *Optics communications*, 107(3):200–204, 1994. doi: http://dx.doi.org/10.1016/0030-4018(94)90020-5.

[20] S. Luo, J. Qian, and H. Zhao. Higher-order schemes for 3–D first-arrival traveltimes and amplitudes. *Geophysics*, 77(2):T47–T56, 2012. doi: 10.1190/geo2010-0363.1.

[21] Songting Luo, Shingyu Leung, and Jianliang Qian. An adjoint state method for numerical approximation of continuous traffic congestion equilibria. *Communications in Computational Physics*, 10 (05):1113–1131, 2011. doi: https://doi.org/10.4208/cicp.020210.311210a.

[22] Guancong Ma and Ping Sheng. Acoustic metamaterials: From local resonances to broad horizons. *Science Advances*, 2(2):e1501595, February 2016. ISSN 2375-2548. doi: 10.1126/sciadv.1501595.

[23] Nina Meinzer, William L. Barnes, and Ian R. Hooper. Plasmonic meta-atoms and metasurfaces. *Nature Photonics*, 8(12):889–898, December 2014. ISSN 1749-4885. doi: 10.1038/nphoton.2014.247.

[24] Graeme W. Milton, Marc Briane, and John R. Willis. On cloaking for elasticity and physical equations with a transformation invariant form. *New Journal of Physics*, 8(10):248, 2006. ISSN 1367-2630. doi: http://dx.doi.org/10.1088/1367-2630/8/10/248.

[25] Xingjie Ni, Zi Jing Wong, Michael Mrejen, Yuan Wang, and Xiang Zhang. An ultrathin invisibility skin cloak for visible light. *Science*, 349(6254):1310–1314, September 2015. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aac9411.

[26] Marcus Noack and Tor Gillberg. Fast computation of eikonal and transport equations on graphics processing units computer architectures. *Geophysics*, 80(5):T183–T9, 2015. doi: 10.1190/geo2014-0556.1.

[27] J. B. Pendry, D. Schurig, and D. R. Smith. Controlling Electromagnetic Fields. *Science*, 312(5781): 1780–1782, June 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1125907.

[28] Carlos Piedrahita, Trino Salinas, Hernando Altamar, and Karen Pachano. Slowness surface calculation for different media using the symbolic mathematics language Maple®. *Earth sciences research journal*, 8(1):63, 2004.

[29] P. Podvin and I. Lecomte. Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools. *Geophysical Journal International*, 105(1), 1991. doi: https://doi.org/10.1111/j.1365-246X.1991.tb03461.x.

[30] D. Schurig, J. J. Mock, B. J. Justice, S. A. Cummer, J. B. Pendry, A. F. Starr, and D. R. Smith. Metamaterial Electromagnetic Cloak at Microwave Frequencies. *Science*, 314(5801):977–980, November 2006. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1133628.

[31] Maxime Sermesant, Ender Konukoglu, Hervé Delingette, Yves Coudière, Phani Chinchapatnam, Kawal S Rhode, Reza Razavi, and Nicholas Ayache. An anisotropic multi-front fast marching method for real-time simulation of cardiac electrophysiology. In *Functional Imaging and Modeling of the Heart*, pages 160–169. Springer, 2007. doi: 10.1007/978-3-540-72907-5_17.

[32] Alexey Stovas and Tariq Alkhalifah. Mapping moveout approximations in ti media. *Geophysics*, 79 (1):C19–C26, 2013. doi: 10.1190/geo2013-0039.1.

[33] Leon Thomsen and Joe Dellinger. On shear-wave triplication in transversely isotropic media. *Journal of Applied Geophysics*, 54(3):289–296, 2003. doi: http://dx.doi.org/10.1016/j.jappgeo.2003.08.008.

[34] Ilya Tsvankin. Anisotropic parameters and P -wave velocity for orthorhombic media. *Geophysics*, 62(4):1292–1309, July 1997. ISSN 0016-8033, 1942-2156. doi: 10.1190/1.1444231.

[35] W. Voigt. *Lehrbuch Der Kristallphysik*. Teubner, Leipzig, 1910.

[36] Steven Weinberg. Eikonal method in magnetohydrodynamics. *Physical Review*, 126(6):1899, 1962. doi: https://doi.org/10.1103/PhysRev.126.1899.

[37] Robert J Young and Alexander V Panfilov. Anisotropy of wave propagation in the heart can be modeled by a riemannian electrophysiological metric. *Proceedings of the National Academy of Sciences*, 107(34):15063–15068, 2010. doi: 10.1073/pnas.1008837107.

[38] Linbin Zhang, James W Rector, and G Michael Hoversten. Finite-difference modelling of wave propagation in acoustic tilted TI media. *Geophysical Prospecting*, 53(6):843–852, 2005. doi: 10.1111/j.1365-2478.2005.00504.x.

Chapter 5

A Proposed Hybrid Method for Function Optimization

Article published in Elsevier's *Journal of Computational and Applied Mathematics*, May 2017
DOI: 10.1016/j.cam.2017.04.047

Optimization, as a principle, is one of the best tools we have to explain natural processes. Furthermore, it is at the basis of countless processes and algorithms in various fields in research and industry such as artificial intelligence, biology, medicine, finance, engineering, fossil and renewable energy, and image processing [2, 12, 16, 18, 26, 45, 46, 53]. Therefore, it is highly desirable to be able to optimize functions efficiently and reliably. Unfortunately, most optimization problems suffer from non-convexity of the objective function and non-uniqueness of the optima, which complicates the computational challenge drastically.

Optimization methods can be divided into two main categories: global and local optimization schemes. Global optimization methods are successful in finding the global optimum eventually [38] but are inefficient or even useless in high-dimensional spaces. Additionally, they cannot assess the uniqueness of a solution since the curvature of the objective function is not used. Local methods, on the other hand, are computationally inexpensive but can reliably find the global optimum of convex functions only. They can assess whether an optimum accommodates many solutions because the Hessian of the function is available. However, the probability to converge in a local optimum is high for local optimization schemes acting on non-convex functions.

The trade-off between reliability and efficiency of current optimization methods and the linked challenges in many fields in research and industry motivate the work that follows[1]. The proposed method was found to be worth protecting by a US patent.

[1]Research Paper 4 was written in British English due to author preference

Hybrid Genetic Deflated Newton Method for Global Optimisation

Marcus M. Noack[1,2,3] and Simon W. Funke[2,3]

[1]Kalkulo AS, P.O.Box 134, 1325 Lysaker, Norway
[2]Simula Research Laboratory, P.O.Box 134, 1325 Lysaker, Norway
[3]Department of Informatics, University of Oslo, Gaustadalleen 23 B, 0373 Oslo, Norway

Abstract

Optimisation is a basic principle of nature and has a vast variety of applications in research and industry. There is a plurality of different optimisation procedures which exhibit different strengths and weaknesses in computational efficiency and probability of finding the global optimum. Most methods offer a trade-off between these two aspects. This paper proposes a hybrid genetic deflated Newton (HGDN) method to find local and global optima more efficiently than competing methods. The proposed method is a hybrid algorithm which uses a genetic algorithm to explore the parameter domain for regions containing local minima, and a deflated Newton algorithm to efficiently find their exact locations. In each iteration, identified minima are removed using deflation, so that they cannot be found again. The genetic algorithm is adapted as follows: every individual of every generation of offspring searches its adjacent space for optima using Newton's method; when found, the optimum is removed by deflation, and the individual returns to its starting position. This procedure is repeated until no more optima can be found. The deflation step ensures that the same optimum is not found twice. In the subsequent genetic step, a new generation of offspring is created, using procreation of the fittest. We demonstrate that the proposed method converges to the global optimum, even for small populations. Furthermore, the numerical results show that the HGDN method can improve the number of necessary function and derivative evaluations by orders of magnitude.

1 Introduction

Optimisation is one of the most fundamental principles of nature. Most physical principles can be formulated in the structure of an optimisation problem. Additionally, inversions, like seismic tomography and weather predictions, are typical optimisation problems. It is therefore important to develop efficient methods to optimise functions. This paper is concerned with the problem of finding the local and global minimisers $\{\mathbf{x}^*\}$ of a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$. More precisely, we are seeking points $\mathbf{x}^* \in \mathbb{R}^n$ for which the optimality condition

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{x}^*|| < r \tag{5.1}$$

holds for a sufficiently small $r > 0$. The search for maxima is analogous and will be treated accordingly. The objective function f can be highly non-linear, but we assume that it is continuous and at least twice

differentiable. In many practical applications, evaluating f or its derivatives involves computationally expensive operations, such as the solution of a discretised partial differential equation. Therefore, it is crucial to solve the problem with as few functional and derivative evaluations as possible.

Solving problem (5.1) is numerically challenging, because f can have multiple local and global optima. Local knowledge about the function, such as evaluations and derivatives, is therefore not sufficient to find the global solution or to identify whether an optimum is a local or a global optimum [7]. Hence, existing local optimisation methods can not be applied directly. Instead, a solution strategy must explore the global parameter space. Genetic algorithms and simulated annealing are popular methods that use randomised search strategies motivated from natural processes. They are robust, find the proximity of the optimal solutions in a reasonable time for a small number of dimensions, are parallelisable and easy to implement [9, 6]. Furthermore, they have little assumptions on the objective function f. However, they require many function evaluations, especially in high dimensional spaces. To improve the efficiency, hybrid schemes have been proposed which combine the efficiency of local optimisation methods with the generality of global methods [15, 17, 1, 14]. Renders and Flasse [14] in particular showed that hybrid methods can offer a significant improvement compared to genetic algorithms. We are referring to these hybrid methods as traditional hybrid methods in the course of the paper.

This paper presents a new hybrid optimisation method that combines a genetic algorithm with a fast, local optimisation method. The algorithm is based on two basic components: a global search method based on the genetic algorithm, and a local search method. For the local search, we employ a deflated Newton scheme [2]. The deflated Newton method efficiently identifies multiple local minima or maxima in proximity of the starting point, and deflates the function accordingly. As a result, a smaller population size is sufficient to efficiently map the local and global optima of f, which we show, can result in a significant performance increase of the overall algorithm. The key to the success of the deflated Newton method is that the found optima are "removed", meaning that a deflation is placed where the optimum was located. A subsequent Newton search will not converge to the same point, but will find another optimum or diverge, meaning that there are no optima in the vicinity of the individual. This leads to a performance gain of the overall algorithm compared to traditional hybrid methods. The following genetic algorithm will relocate the individuals by using procreation of the fittest. In the new locations, all offspring individuals will again start the search for optima. The proposed method is easy to implement because existing implementations of genetic and local optimisation methods can be reused. The overall goal is to minimise the required function and derivative evaluations to find local and global optima of a function.

The remainder of the paper is organised as follows. In Section 2, the ingredients of the proposed method are mentioned and explained briefly. The following sections will give some information about global and local optimisation schemes. Next, it is explained how these methods work together to form the basis of the proposed hybrid method. Afterwards, the method of deflation is described and used to improve the existing hybrid methods. The proposed method was applied in several standard problems and benchmarked against genetic and traditional hybrid optimisation methods. The numerical results are shown in Section 3.

2 Methodology and Theory

Two classes of methods exist when it comes to optimising functions: local methods and global methods. In most fields, the use of either local or global methods means a trade-off between computing time and

Algorithm 1: Recombination(Population)

select fittest individuals;
perform crossover;
perform mutation;
return Population

probability of finding the global optimum. This section will give an introduction to local and global optimisation schemes and will use them to draw the path to the proposed method.

2.1 Global Methods

Global methods, like the Monte-Carlo method or the genetic algorithm are randomised algorithms and can guarantee to find the global optimum [11]. For our purposes the genetic algorithm is particularly interesting. This algorithm is inspired by the natural selection in biological evolution and works as follows. The core of a genetic algorithm is called recombination and is shown in Algorithm 1. A random population is created and placed in the search space. We refer to a population as a plurality of chosen points (individuals) in the search space. The fittest individuals have the best chance to procreate and produce offspring. The fitness in this context is the function value at the point that is associated with a certain individual. The offspring is built by crossover of the genome (the location in the search space) of the parent individuals. There are many different types of crossover, ranging from one-point crossover to completely random methods [8]. After the crossover, mutation can happen randomly, which gives the genetic algorithm the ability to find the global optimum eventually. Mutation means a random change of the genome (the location) of an affected individual. Mutation, and the chance for individuals that are not among the fittest to procreate, give the genetic algorithm an unbiased behaviour. However, for real life applications, global optimisation methods are often not feasible because the number of necessary function evaluations exceeds feasibility for a high number of dimensions of the search space. It is potentially useful to consider other global methods besides the genetic algorithm. The Monte-Carlo [3] method and particle swarm optimisation [10] could, among others, also lead to satisfying results.

2.2 Local Methods

Local methods are mostly derivative-based methods, like the steepest decent and the Newton method. Local methods are computationally cheap compared to global methods, but they have a high risk of converging to a local optimum. The Newton method is of particular interest for our purpose for reasons that will become apparent later on. The Newton method computes the gradient and the Hessian at a certain point of the function and uses this information to predict a new location for the individual by solving

$$H(\mathbf{x})\gamma = -\nabla f(\mathbf{x}), \tag{5.2}$$

where H is the Hessian matrix, $H(\mathbf{x})_{ij} = \frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j}$, $\mathbf{x}$ is the current position and γ is the improvement from the current to the next position. For strictly convex functions, the Newton method is successful in finding the global optimum. The pseudo code of the Newton method is shown in Algorithm 2. In practice, most functions we seek to optimise show more complicated properties. If a function is not convex, the Newton method can converge in a local optimum or a saddle point. It is possible to find several stationary points using the Newton method repeatedly. The deflated Newton method is an extension to the Newton method

Algorithm 2: Function: Newton(Position $\mathbf{x}$)

while *change in position* $> \epsilon$ **do**
  compute gradient and Hessian of function at position $\mathbf{x}$;
  Solve $H(\mathbf{x})\gamma = -\nabla f(\mathbf{x})$ for example with CG or MINRES method
  update position $\mathbf{x} = \mathbf{x} + \gamma$;

that allows to identify multiple local minima [16, 2, 5]. The deflation process removes an identified root from a function to make sure that in a subsequent Newton step individuals can not find the same optimum again. The deflated Newton method works as follows: The Newton optimisation method searches for stationary points of f, i.e. points $\mathbf{x}^*$ where $f'(\mathbf{x}^*) = 0$. Let $\mathbf{x}_1, .., \mathbf{x}_N$ be stationary points that have already been identified. Then, further stationary points of f can be found by considering the deflated gradient of the function f by using

$$\nabla f_{x_0}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{\prod_{i=1}^{N} \|\mathbf{x} - \mathbf{x}_i\|^2}, \tag{5.3}$$

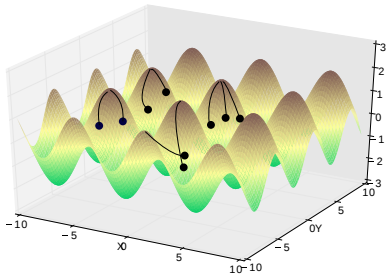
where N is the number of deflated points. The deflated function has no roots at the known optima $\mathbf{x}_i$, and hence the local search method will not converge to these roots again (see Figure 5.1d). We will implement the method of deflation, using a genetic algorithm for many individuals, which presents the main novelty of the proposed method.

2.3 Traditional Hybrid Methods

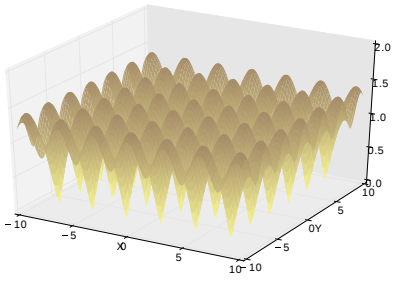
To combine the strengths of global and local optimisation, hybrid methods have been developed in the past [14]. Hybrid methods are using a global search algorithm to explore the search space on a global level. At this stage, we will focus on hybrid schemes that use the genetic algorithm as a global search scheme and the Newton method as a local scheme. After each iteration of the genetic algorithm, all individuals perform a Newton search to find a stationary point of the function. When all individuals have converged, the genetic algorithm chooses the fittest individuals and creates offspring. The next generation is, in general, fitter than the last one which leads to the convergence of the algorithm. After a new generation is created, all individuals start again the search using the Newton scheme.

2.4 The New Hybrid Scheme

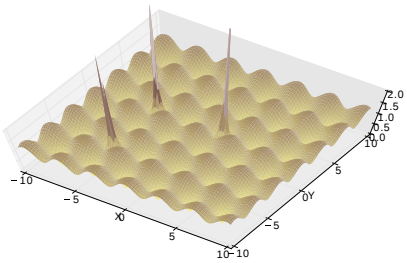
How can the existing hybrid methods further be accelerated? Consider one genetic iteration of such a hybrid scheme as visualised in Figure 5.1a. An obvious drawback is that the local search method might compute the same optima for different individuals; also, these optima might be re-identified over and over again in each genetic iteration. Therefore, a significant amount of computational effort is potentially utilised identifying already known optima. Only if an individual is positioned sufficiently close to a new optimum, then the local search will converge to this new optimum. At this point, the key idea comes into play. Combining hybrid schemes with the knowledge about deflations, a new scheme can be created. The new scheme places a plurality of individuals randomly in the search space. Subsequently, every individual uses a Newton method to find a stationary point of the function. Once at the stationary point, the gradient of the function is deflated (see Figure 5.1b and 5.1c) at this location and the point and its function value are stored in a list of stationary points. When a certain amount of individuals have converged, all individuals



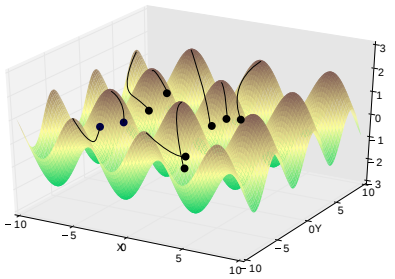
(a) The hybrid method applies local optimisation to each individual. In the next iteration, the same optima can be identified again.



(b) The Euclidean norm of the gradient of the function. Optima in the original function are roots in the gradient.



(c) The gradient of the function is deflated at the optima of the function. The roots are removed.



(d) The hybrid genetic deflated Newton method removes any identified optima after each iteration. Therefore, already found optima cannot be identified again. Every individual can find new optima.

Figure 5.1: Illustration of the key idea of deflation. Traditional hybrid methods can find already identified optima. Deflation, on the other hand, prevents the algorithm from identifying the same optima again.

are set back to their original locations to start a new local search. The individuals can not converge to the same points thanks to the deflations (see Figure 5.1d). This procedure is repeated until a user defined percentage of individuals can not find a stationary point anymore. When this happens, a genetic iteration relocates the population by creating offspring using procreation of the fittest and the local search is started again. This way, progressively more stationary points are found. The search can be terminated when no new stationary points can be found or when a certain amount of new found optima show a similar function value.

The modified algorithm for the hybrid genetic deflated Newton method is given in Algorithms 3 and 4. If a hybrid method is already implemented, the required changes are minimal: the only change is to replace the Newton method with the deflated Newton method, and to keep a list of all identified minima.

2.5 Deflation Operators

When coping with challenging optimisation problems, the simple deflation strategy given in equation (5.3) is often numerically not robust and alternative deflation operators need to be investigated. A more

Algorithm 3: Top-level structure of the new hybrid algorithm.

```
initialise Population of individuals of length n;
initialise OptiList = []
Population = DNewton(Population, OptiList);
while Change in population fitness > ϵ do
    Population = Recombination(Population);
    Population = DNewton(Population, OptiList);
```

Algorithm 4: Function: DNewton(Population, OptiList)

```
while Convergence criterion not fulfilled do
    for all individuals in parallel do
        while Change in fitness of the individual > ϵ do
            deflate gradient of function based on OptiList
            compute deflated gradient and Hessian of function at position x;
            Solve H(x)γ = −∇f(x) for example with CG or MINRES method
            update position x = x + γ;
        append x to OptiList;
    set Population back to initial position;
return fittest N components of OptiList
```

general version of the basic deflation operator (5.3) for deflating a single root $\mathbf{x}_0$ is

$$\nabla f_{\mathbf{x_0}}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{\|\mathbf{x} - \mathbf{x}_0\|^p}, \tag{5.4}$$

where $p \in \mathbb{N}$. The application of the deflation operator (5.4) is numerically problematic, because the deflated function converges to 0 for $\mathbf{x} \to \infty$ if f is bounded. Subsequently, applying Newton to $f_{\mathbf{x}_0}$ might diverge. This can be avoided by introducing a shifting of the deflation operator of the form

$$\nabla f_{\mathbf{x}_0}(\mathbf{x}) = \left(\frac{1}{\|\mathbf{x} - \mathbf{x}_0\|^p} + 1\right) \nabla f(\mathbf{x}), \tag{5.5}$$

as shown by Farrell et al. [5]. Here, for $\mathbf{x} \to \infty$ the deflated function behaves as the original function.

A main drawback of the deflation operators (5.4) and (5.5) is that they alter the function globally, which results in degeneration if the deflation is applied many times. This can be seen when applying the deflation to two different roots $\mathbf{x}_0$ and $\mathbf{x}_1$

$$\begin{aligned} \nabla f_{\mathbf{x}_0,\mathbf{x}_1}(\mathbf{x}) &= \left(\frac{1}{\|\mathbf{x} - \mathbf{x}_0\|^p} + 1\right) \nabla f_{\mathbf{x}_1}(\mathbf{x}) \\ &= \left(\frac{1}{\|\mathbf{x} - \mathbf{x}_0\|^p} + 1\right) \left(\frac{1}{\|\mathbf{x} - \mathbf{x}_1\|^p} + 1\right) \nabla f(\mathbf{x}). \end{aligned} \tag{5.6}$$

The function is multiplied by a scalar coefficient > 1, which grows quickly if many deflations are applied. Traditional deflation, as well as shifted deflation, lead to an altering of the function outside a vicinity of the deflation and are therefore numerically problematic when many deflations occur. For many deflations, the function values can fall below machine precision or grow beyond feasibility which leads to numerical problems. The principle is illustrated in Figure 5.2a.

To solve this problem, we introduced the *localised deflation operator*. The localised deflation operation uses a bump function (a smooth function with compact support) to affect an area close to the deflation

only. The normalised bump function in n dimensions is given by

$$b_{\mathbf{x_0}}(\mathbf{x}) = \begin{cases} \prod_{i=1}^{n} \frac{\exp(\frac{-\alpha}{r^2-(x_i-x_{i0})^2})}{\exp(\frac{-\alpha}{r^2})} & \text{if } x_{i0} - r < x_i < x_{i0} + r \\ 0 & \text{else} \end{cases} \tag{5.7}$$

where $\mathbf{x}_0$ is the location of the center of the deflation, r is the radius of the deflation and α is a coefficient to adjust the shape of the bump function as can be seen in Figure 5.3. An adapted shape of the bump function leads to a more efficient prevention of individuals converging into the deflated optimum. The deflated function is then given by

$$\nabla f_{\mathbf{x_0}}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{1 - b_{\mathbf{x_0}}(\mathbf{x})}. \tag{5.8}$$

The employment of the localised deflation with coefficient α allows for highly shapeable deflations at the right locations, without altering the function anywhere else. Highly shapeable bump functions mean a better avoidance of the convergence of an individual in the Newton step. The effect of different deflation operators can be seen in Figure 5.2.

3 Experiments and Results

To prove the functionality of the proposed hybrid method, four experiments were conducted. The experiments are standard benchmark examples for genetic algorithms [4] and introduced hereafter. The proposed method, referred to as hybrid genetic-deflated Newton method (HGDN), was challenged to find the global maximum of the example functions using less function and derivative evaluations than a genetic algorithm and a traditional hybrid genetic/Newton algorithm [14], referred to as Genetic-Newton method. The number of function and derivative evaluations is, as a measure, more meaningful than the overall computing time, which highly depends on the implementation, the forward problem and the computer architecture.

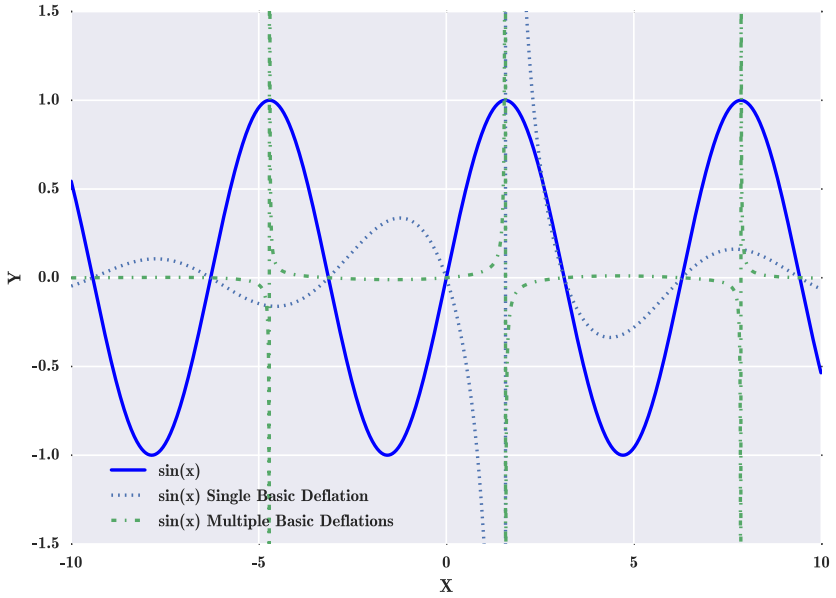
The genetic algorithm uses a crossover which creates a child individual by using alternate genes from the mother and the father individual. In all our experiments, mutation of one percent of the genome value could occur with a probability of 80 percent. To improve the unbiased behaviour of the algorithm, a second stage of mutation could occur with a probability of ten percent. This kind of mutation changed the value of the genome randomly in the limits of the search space. The Newton algorithm is terminated when the change in location from one iteration to the next falls below a certain user-defined threshold (10^{-6} for our experiments).

3.1 Ackley's Function

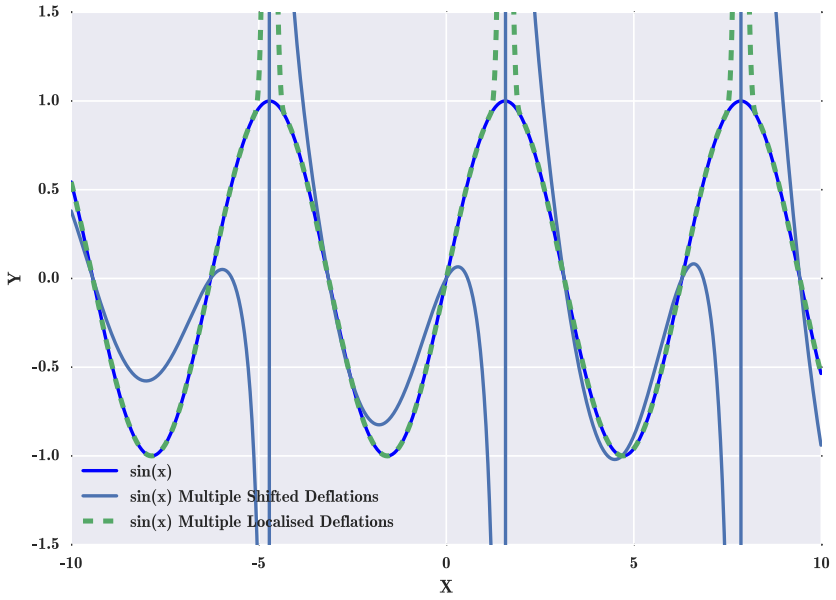
The Ackley's function in n dimensions is defined as

$$f(\mathbf{x}) = 20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) + \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) - 20 - \exp(1) \tag{5.9}$$

and is shown in Figure 5.4. It is considered to be relatively easy to optimise for a genetic algorithm [4] due to the guiding slope giving a preferred search direction. We included Ackley's function because of its potential relevance in real world applications [4]. For the first experiment, we want to optimise Ackley's



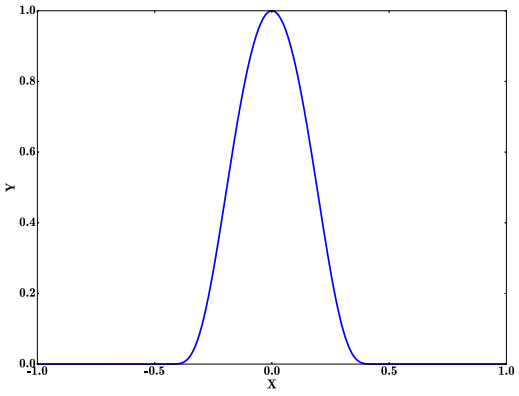
(a) An original sine function, and deflated sine functions, obtained by using the basic deflation operator (5.3). Note that the function is altered everywhere when deflation is applied. The single deflated sine function exhibits much smaller amplitudes and presents a phase shift. The multiple deflated sine function shows nearly no visible amplitude.



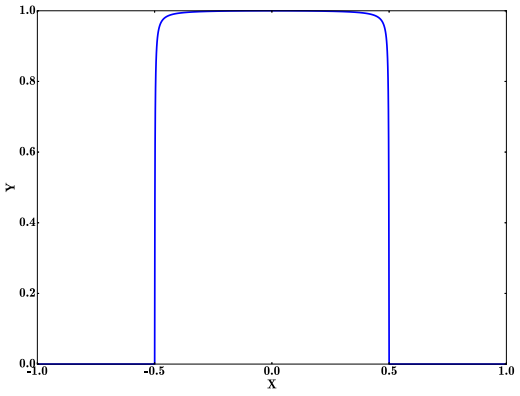
(b) A sine function, and the deflated sine functions, obtained by using shifted and localised deflation. Note, that applying the shifted deflation operator (5.5) multiple times yields to changed amplitudes and a phase shift in the adjacent areas of the deflation points. The localised deflation, on the other hand, leaves the function unaffected outside a certain radius.

Figure 5.2: The effect of different deflation operators applied to a sine function. Note, that only the localised deflation leaves the global shape of the function uncompromised.

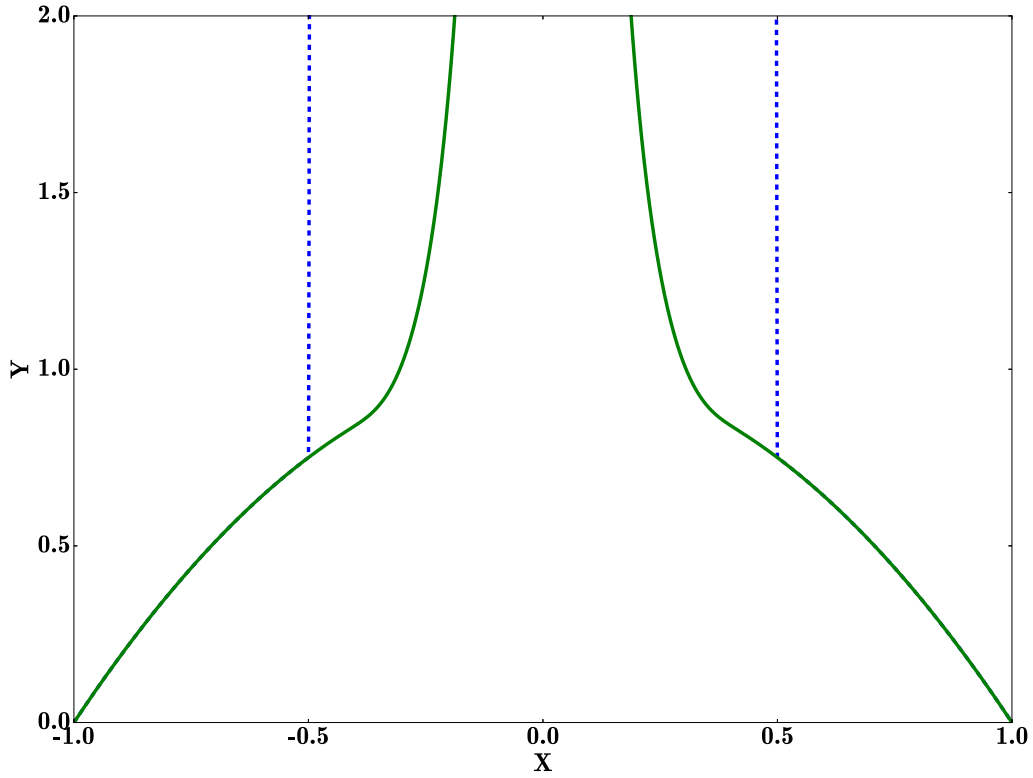
function in the limits $-10 \leq x_i \leq 10$ (see Figure 5.4). The search space gives the limits of the random placement of the first generation of individuals. Ackley's function comprises many local optima but shows



(a) A bump function with $\alpha = 1$.



(b) A bump function with $\alpha = 0.1$.



(c) The resulting change in the deflated function. The dashed curve shows the new deflation using the bump function with $\alpha = 0.1$.

Figure 5.3: Two different bump functions and their effect on the deflated function. Note that the boundaries of the deflation are much more distinct when using lower values for α.

a steep guiding slope towards the global optimum.

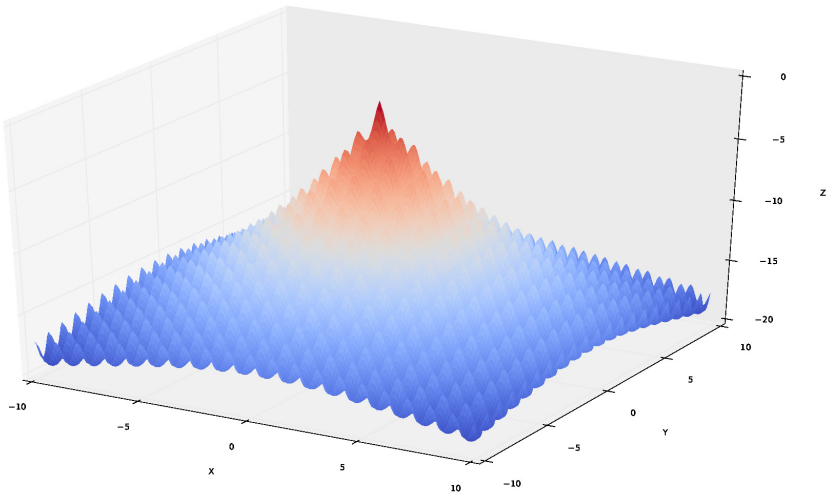


Figure 5.4: Ackley's function exhibits a steep guiding slope towards the global optimum, which is located in the center. The function contains many local optima, which can mislead local gradient based methods.

3.2 Rastrigin's Function

We define Rastrigin's function in n dimensions as

$$f(\mathbf{x}) = -10n - \sum_{i}^{n} x_i^2 - 10\cos(2\pi x_i). \tag{5.10}$$

Rastrigin's function is illustrated in Figure 5.5. It has a less steep guiding slope than Ackley's function which complicates the optimisation process. On the other hand, there are less local optima in the search space $-5.12 \leq x_i \leq 5.12$.

3.3 Schwefel's Function

The Schwefel's function in n dimensions is defined as

$$f(\mathbf{x}) = -418.9829n - \sum_{i}^{n} \begin{cases} -x_i \cdot \sin(\sqrt{|x_i|}) & \text{if } -500 \leq x_i \leq 500 \\ 0.02 \cdot x_i^2 & \text{else} \end{cases} \tag{5.11}$$

and is shown in Figure 5.6. The function does not exhibit a guiding slope which points in the direction of the global optimum. In addition, the function is less symmetric than the previous functions and the global optimum is located close to the border of the search space $-500 \leq x_i \leq 500$, which complicates the optimisation because the average distance of the randomly placed individuals of the first generation to the optimum is greater than for a centered global optimum.

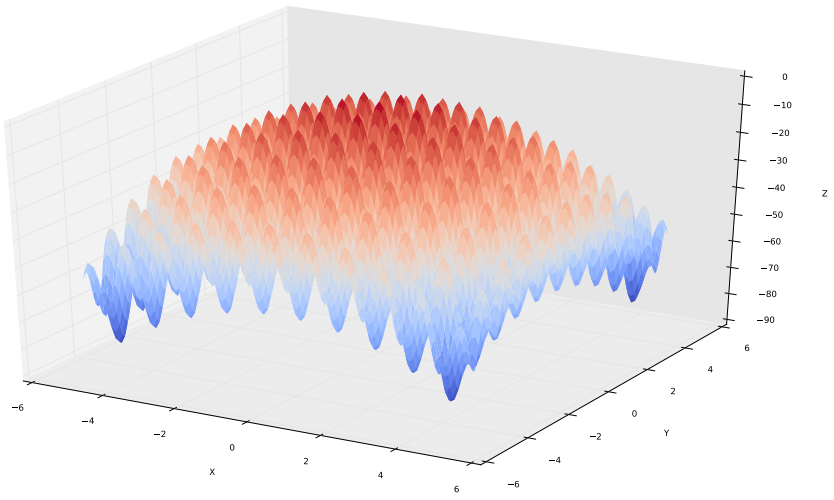


Figure 5.5: Rastrigin's function exhibits a less steep guiding slope towards the global optimum than Ackley's function. The function contains a plurality of optima in the search space, which can mislead local gradient based methods.

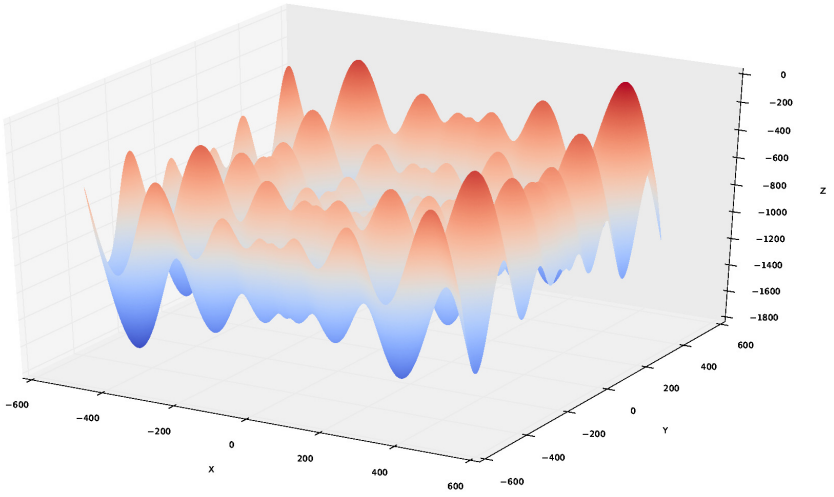


Figure 5.6: Schwefel's function exhibits no guiding slope towards the global optimum which is located close to the border at $x_i = 420.97$ for $i = 1, \ldots, n$.

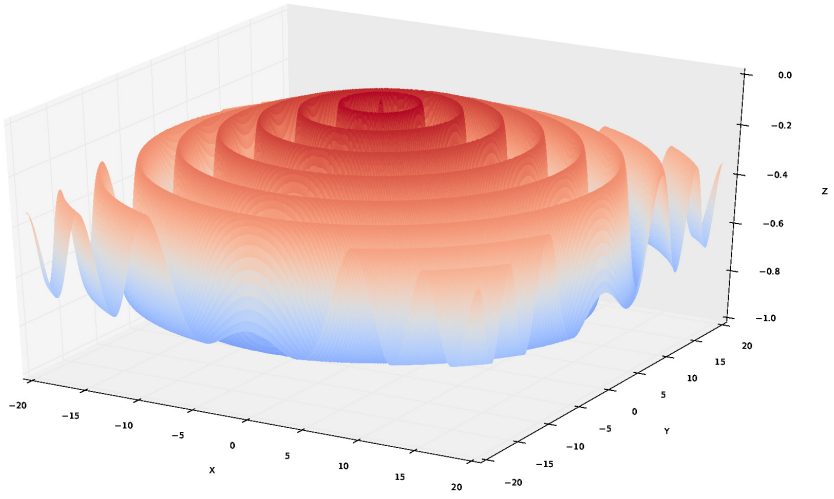


Figure 5.7: The difficulty in optimising the Schaffer's F6 function lies in the fact that the function value of the local minima increases towards the global maximum. Furthermore, the local optima of an n dimensional Schaffer's F6 function are $n-1$ dimensional, which complicates a successful deflation.

3.4 Schaffer's F6 Function

The Schaffer's F6 function in n dimensions is defined as

$$f(\mathbf{x}) = -0.5 - \frac{\sin^2(\sqrt{\sum_{i=1}^{n} x_i^2}) - 0.5}{(1 + 0.001(\sum_{i=1}^{n} x_i^2))^2} \tag{5.12}$$

and is shown in Figure 5.7. The Schaffer's F6 function represents a special challenge for the proposed algorithm since it exhibits local optima that are $n-1$ dimensional. More formally said, the null space of the Hessian at the optimum is $n-1$-dimensional. An infinite amount of deflations is necessary to deflate a $n-1$ dimensional optimum entirely, which poses a challenge for the HGDN method. Another challenge lies in the increasing amplitude of the wave-like function towards the optimum. The search space was chosen to be $-20 \leq x_i \leq 20$.

3.5 Comparison of Number of Function and Derivative Evaluations

In most inversion problems, the forward modelling is the most costly step; therefore, it is desirable to minimise the number of function evaluations of an optimisation procedure. Furthermore, the number of gradient/Hessian computations has a large impact on the computational performance and therefore, need to be minimised as well. The proposed method was up against a genetic algorithm and the hybrid Genetic-Newton method. The three methods were supposed to find the global optimum of the above introduced functions. All experiments were run 50 times with random starting populations and optimised parameters, and sorted with respect to the number of function evaluations. The break condition was reaching the global optimum. The number of starting individuals is given in Table 5.1. The number of starting individuals was chosen in order to guarantee the convergence, in most cases, to allow for a fair

	Ackley	Rastrigin	Schwefel	Schaffer F6
Genetic 2d	20	200	200	200
Genetic-Newton 2d	20	50	20	10
HGDN 2d	20	40	20	10
Genetic-Newton 10d	200	50	20	10
HGDN 10d	20	40	20	10

Table 5.1: Table showing the number of starting individuals. Note that the proposed method used the same number or less starting individuals than the competing methods.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic Alg.	1670.0	4451.5	0	0
Genetic-Newton	4354.5	11932.7	12695.3	34830.1
HGDN	845.8	2865.3	6308.0	21669.2

Table 5.2: Means and variances of 50 runs using Ackley's function in two dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic-Newton	9820.0	2115.2	36163.1	7452.5
HGDN	1221.2	268.4	7962.3	1736.2

Table 5.3: Means and variances of 50 runs on Ackley's function in ten dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

comparison. The genetic algorithm was not applied to the ten-dimensional case because the superiority of the Genetic-Newton and the HGDN method is already apparent in the two-dimensional example. Also, the high number of individuals needed, renders the genetic algorithm uncompetitive in ten dimensions.

The number of function evaluations to optimise Ackley's function (Figure 5.4) is presented in Figure 5.8 and Tables 5.2 and 5.3. The guiding slope leads to a fast convergence of the genetic algorithm into the global optimum. Since the local gradient is not used in the genetic algorithm, the many local optima cannot degrade the convergence rate. The guiding slope, on the other hand, is used implicitly, since fitter individuals have a higher probability of procreating. The hybrid methods use the local gradient which degenerates the convergence. The small wave length structure of the function misleads the methods that use local gradient information, which leads to an increased number of function and derivative evaluations. Nevertheless, the proposed HGDN method showed a fast convergence towards the global optimum as shown in Figure 5.8. The superiority of the HGDN algorithm is more apparent in the ten dimensional case. It has to be stated here that the Genetic-Newton algorithm needed more individuals to guarantee the convergence in ten dimensions. Using the same number of individuals for all competing algorithms leads to many runs of the Genetic-Newton algorithm not converging to the global optimum within a given allowed maximum number of genetic steps. The HGDN method, on the other hand, needs fewer individuals to converge.

Rastrigin's function (Figure 5.5) does not show a steep guiding slope which leads to more function evaluations for the genetic algorithm. The HGDN method outperforms its opponents when optimising Rastrigin's function (see Figure 5.9 and Tables 5.4 and 5.5). The tables not only show clearly that

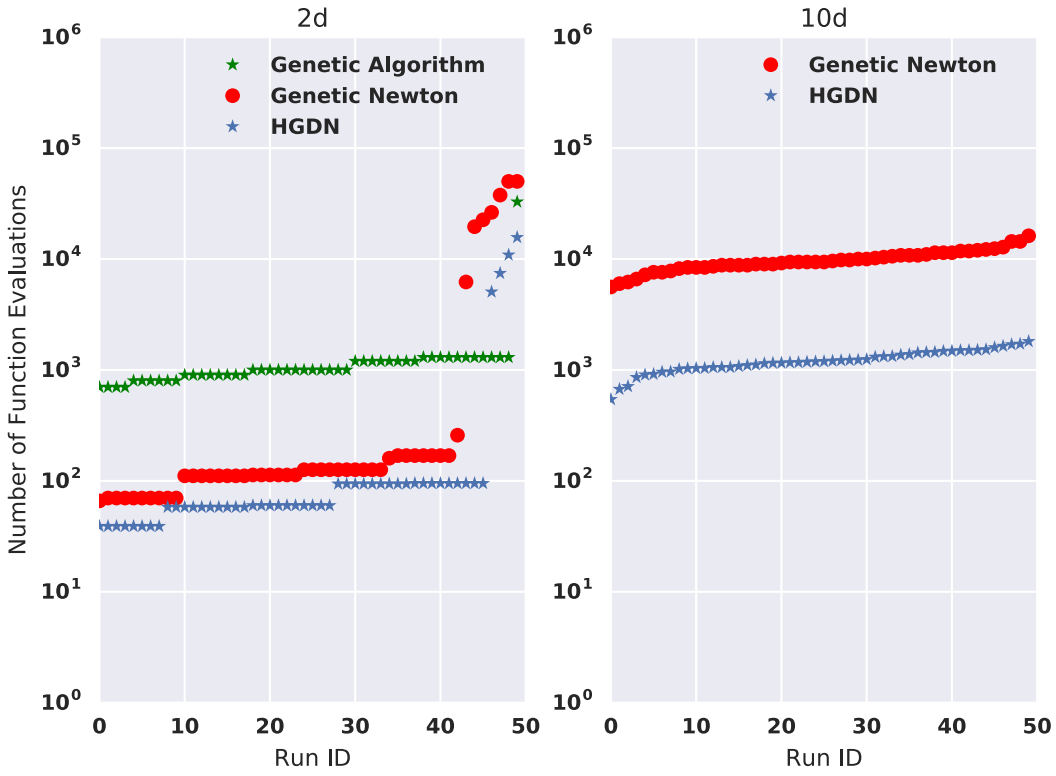


Figure 5.8: The number of function evaluations in two and ten dimensions for Ackley's function of 50 optimisation runs. The genetic algorithm is outperformed by the Genetic-Newton and the HGDN algorithm in 2 dimensions. In ten dimensions, the HGDN algorithm optimises Ackley's function in fewer function evaluations than the Genetic-Newton algorithm.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic Alg.	2908.0	8772.1	0	0
Genetic-Newton	1578.7	7353.0	4384.8	19954.1
HGDN	94.9	78.1	769.8	691.3

Table 5.4: Means and variances of 50 runs on Rastrigin's function in two dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic-Newton	20856.5	12712.8	70892.2	43082.6
HGDN	4677.5	2609.0	27766.5	14085.6

Table 5.5: Means and variances of 50 runs on Rastrigin's function in ten dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

the HGDN method needed fewer function and derivative evaluations, the variances are also smaller. Rastrigin's function was optimised reliably and efficiently by both hybrid algorithms. The reason is the quadratic shape which leads to a fast convergence of the local Newton method. However, the proposed method outperformed its opponents, which becomes more apparent in ten dimensions than in two.

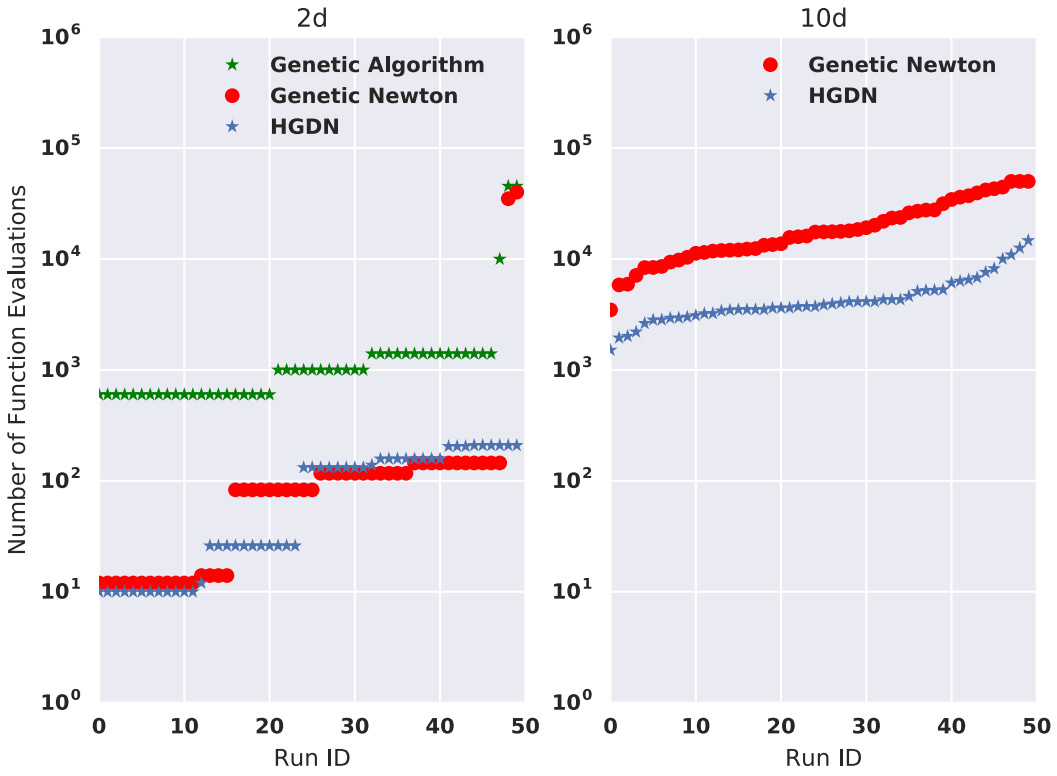


Figure 5.9: The number of function evaluations of Rastrigin's function in two and ten dimensions of 50 optimisation runs. Both hybrid methods perform well optimising Rastrigin's function in two and ten dimensions. The reason is the quadratic envelope of Rastrigin's function which leads to a fast convergence of the local Newton optimisation scheme. The HGDN method performs better because it avoids frequent convergence in one of the many local optima.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic Alg.	53400.0	63925.8	0	0
Genetic-Newton	1636.0	5427.1	7049.1	22575.3
HGDN	232.0	860.2	6557.7	26279.9

Table 5.6: Means and variances of 50 runs on Schwefel's function in two dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

Schwefel's function (Figure 5.6) is much more difficult to optimise due to the missing guiding slope, the missing symmetry and the non-centered optimum. The performance of the proposed method was of particular importance in this experiment because of its relevance for real life optimisation problems. The HGDN method outperforms both opponents (see Figure 5.10 and Tables 5.6 and 5.7). Again, the tables show a smaller and more stable number of function and derivative evaluations. The results also show that the proposed method benefits from a higher number of dimensions.

Schaffer's F6 function poses (Figure 5.7) the problem of $n-1$ dimensional local optima. Nevertheless, the proposed approach outperforms its opponents, as seen in Figure 5.11 and Tables 5.8 and 5.9. The results for Schaffer's F6 function confirm earlier results. The proposed method optimises the function more efficiently than the competing methods. The HGDN method does not benefit from higher dimensions

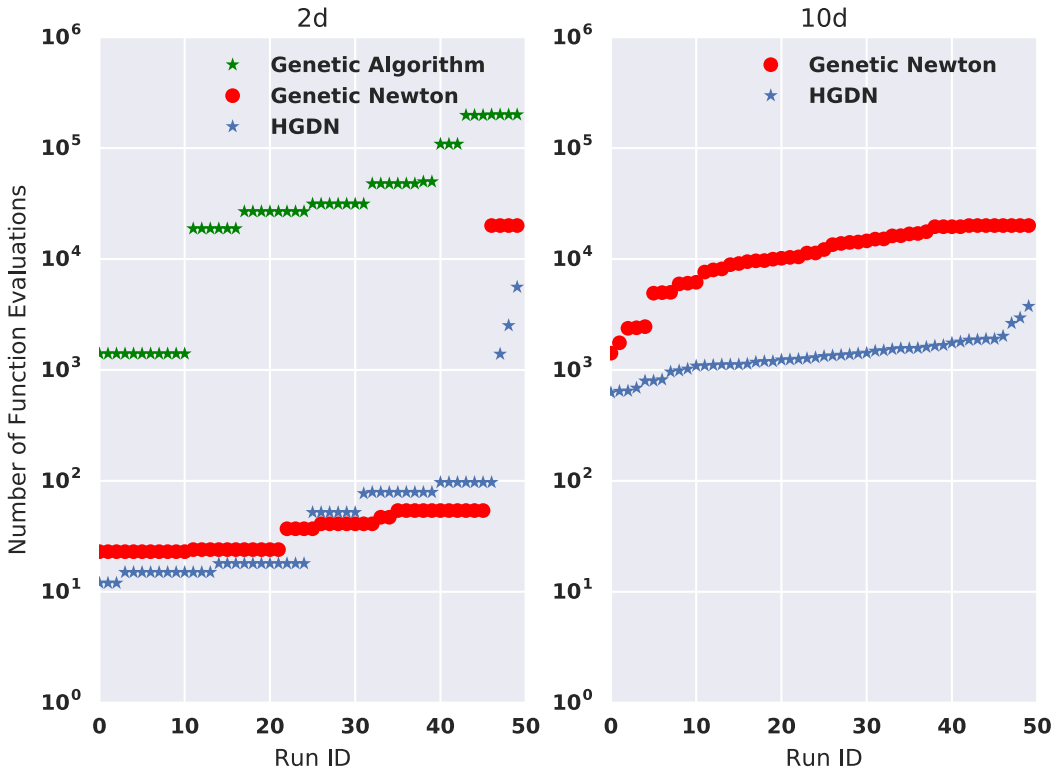


Figure 5.10: The number of function evaluations of Schwefel's function in two and ten dimensions of 50 optimisation runs. The HGDN method outperforms genetic and Genetic-Newton methods optimising Schwefel's function. In the two dimensional case, Genetic-Newton also converges fast towards the global optimum. In ten dimensions the Genetic-Newton is clearly outperformed by the proposed method.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic-Newton	12254.7	5908.6	162648.1	76878.9
HGDN	1408.7	566.4	34540.1	14329.3

Table 5.7: Means and variances of 50 runs on Schwefel's function in ten dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic Alg.	1336.0	1113.9	0	0
Genetic-Newton	1082.8	2803.6	3923.4	9956.5
HGDN	30.88	19.6	252.2	195.2

Table 5.8: Means and variances of 50 runs on Schaffer's F6 function in two dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

in this experiment. The reason is the $n-1$-dimensional optima of the function, which can not be deflated entirely by the deflation operators introduced in this paper.

It should be noted, that compared to the competing methods, the case of divergent runs never appears when applying the proposed method.

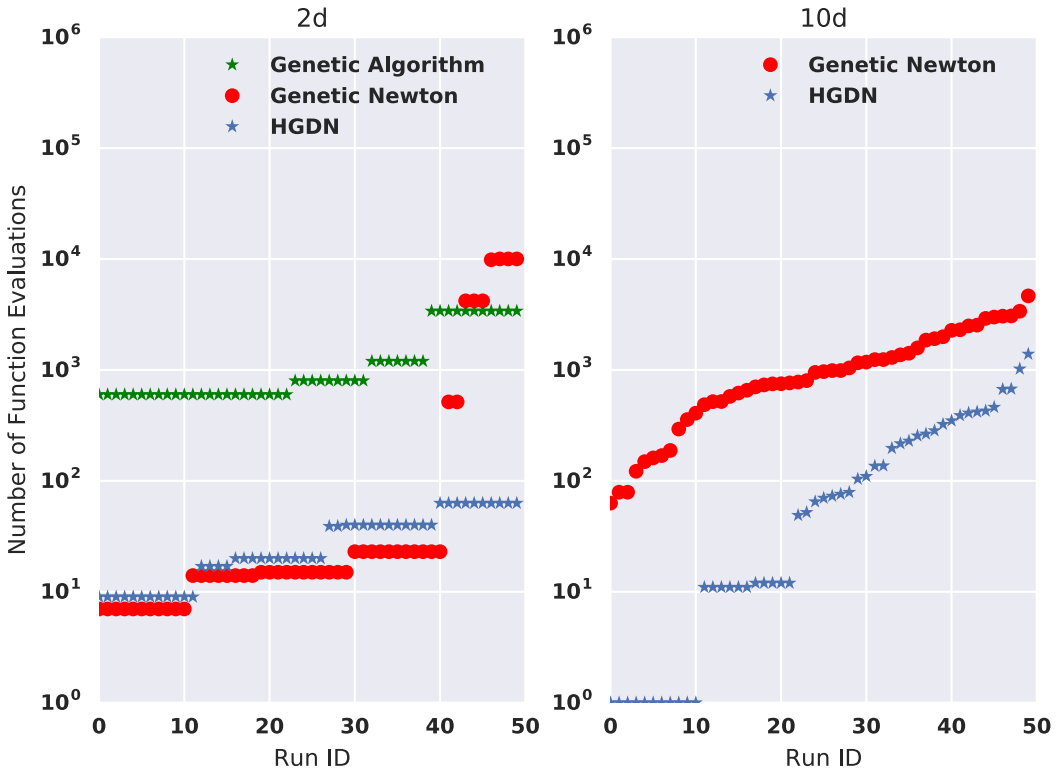


Figure 5.11: The number of function evaluations of Schaffer's F6 function in two and ten dimensions of 50 optimisation runs. Despite difficulties linked to the shape of the optima, the improvement in the number of function evaluations used for optimising Schaffer's function is significant compared with the two competing methods.

Method	$\overline{\text{Func. Eval.}}$	σ Func. Eval	$\overline{\text{Grad./H. Comp}}$	σ Grad./H. Comp.
Genetic-Newton	1231.4	1028.1	4824.7	3954.7
HGDN	181.5	275.9	1109.2	1577.6

Table 5.9: Means and variances of 50 runs on Schaffer's function in ten dimensions. Runs that did not converge and the corresponding runs of the competing methods were excluded.

4 Selected Comparisons to State-Of-The-Art Hybrid Methods

The HGDN method showed in the preceding sections that it can optimise complex functions efficiently compared to basic algorithms. To show that our method performs well compared to state-of-art optimisation procedures, we challenged the HGDN method against the published results of three hybrid methods. All results of the HGDN runs are averages of 50 runs. For the reader to comprehend the comparisons, we introduce two more test functions. The Rosenbrock function

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \tag{5.13}$$

(see Figure 5.12) has only one stationary point. In this case, the HGDN method becomes a hybrid Newton/Genetic algorithm. However, the shown results still confirm the efficiency of the HGDN method.

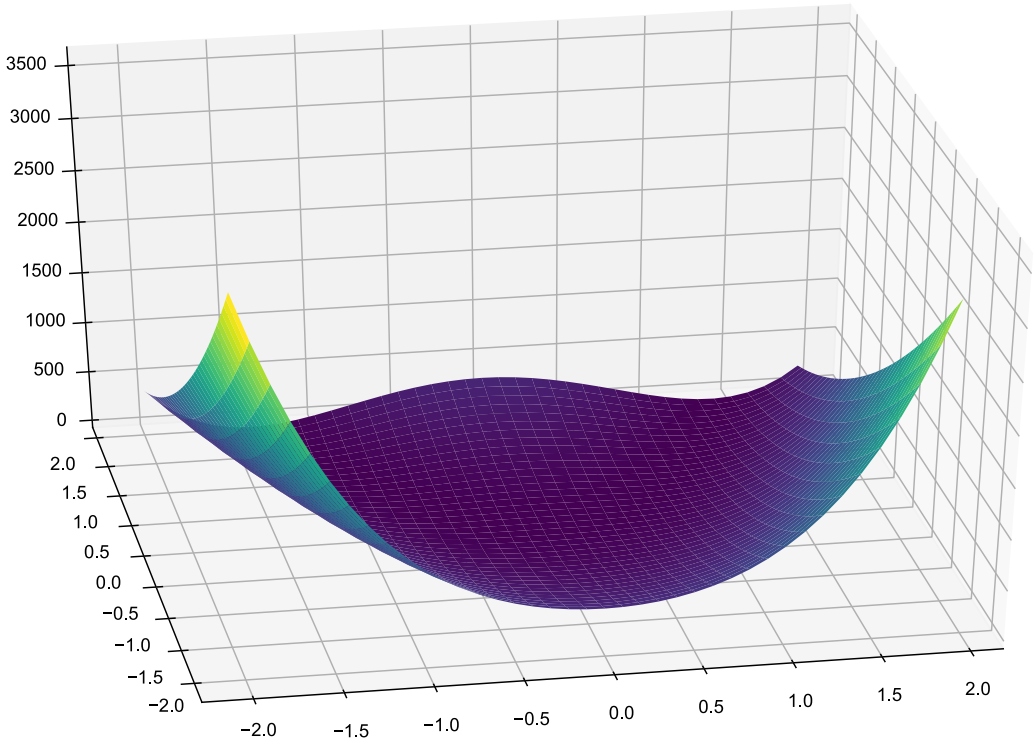


Figure 5.12: The Rosenbrock function depicted in equation (5.13). The function has one stationary point which can be costly to find since it is located in a curved valley.

The Rosenbrock function is commonly regarded as easy to optimise because the function has only one stationary point; however, the function is well suited to test the efficiency of algorithms since the optimum is located in a curved valley.

The HGDN method is designed to optimise functions with many optima. To show that, the method is challenged to optimise the test function

$$f(x,y) = -\cos(x) - \cos(y) - 1.5\sin(2\pi x)\sin(2\pi y) - 3e^{-\frac{x^2}{400}-\frac{y^2}{100}}, \tag{5.14}$$

illustrated in Figure 5.13, as efficient as possible. This test function has more than 162 billion optima in the chosen interval [18].

4.1 Comparison to EPSO

Miranda and Fonseca [13] proposed a hybrid method of evolutionary and particle swarm algorithm. The method was applied to several test functions i.e. the Schaffer's F6 and the Rosenbrock function. Miranda and Fonseca [13] reported the results presented in Table 5.10. The result of HGDN applied to the same problems was added to Table 5.10. It has to be stated here that the HGDN method was applied to the test functions as if we did not know which function we were optimising. Otherwise, the Rosenbrock function could be optimised more efficiently using only one individual; in which case, the HGDN method would fall back to a common Newton scheme.

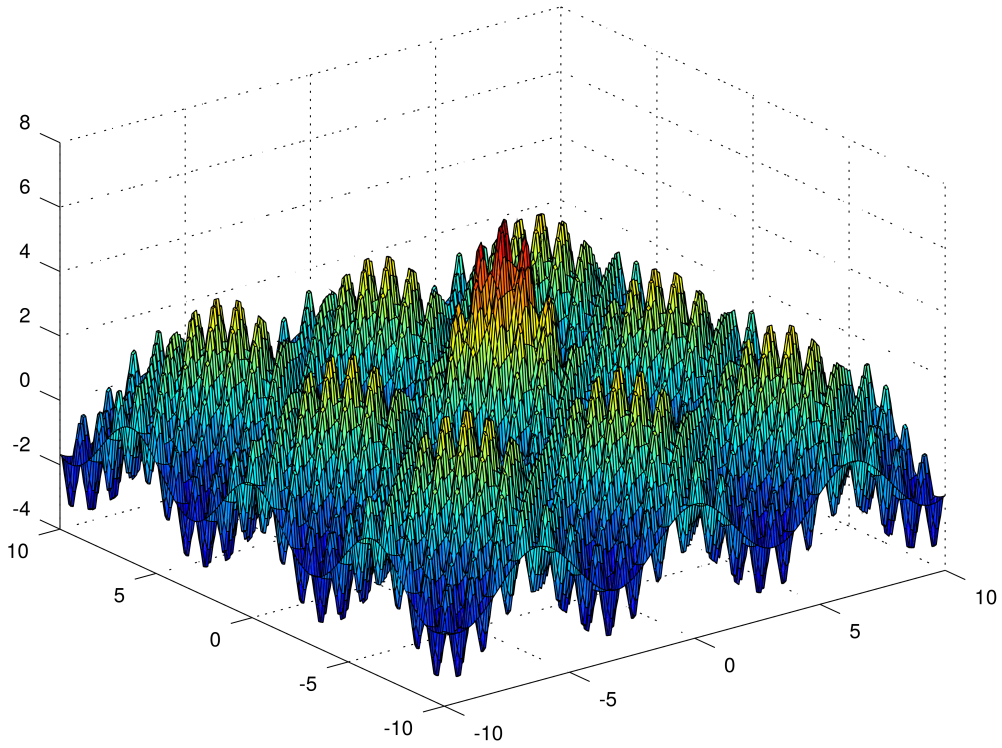


Figure 5.13: The test function depicted in equation (5.14). The function was chosen because of the vast number of optima. In addition, the function only shows a guiding slope in proximity to the optimum.

Test Function	Dimensions	Interval	F. Eval. EPSO	F. Eval. HGDN
Schaffer's F6	2	$[-50, 50]$	11862.1	1980.0
Rosenbrock	30	$[0, 30]$	27005.3	280.0 (15.0)

Table 5.10: Parameters and performance details reported by Miranda and Fonseca [13] for the EPSO method and results of HGDN applied to the same problems as specified in the table. For the Rosenbrock function the HGDN method can use only one individual in which case the method falls back to a common Newton scheme. The number of function evaluations in this case is presented in brackets.

4.2 Comparison to a Local Optimiser/Feasible Point Finder

Xu [18] proposed a global optimiser which employs local optimisation and a feasible point finder. Results are reported for several test functions. For the comparison we chose the test function (5.14) illustrated in Figure 5.13. Xu [18] reported the CPU time until the global optimum is found. The average CPU time to find the minimum of the chosen test function in 2 dimensions in the interval $[-10^6, 10^7] \times [-10^6, 10^7]$ is 10.77 seconds on a Pentium II 400 MHz processor. We executed the experiment on a 2.7 GHz Intel i7 Processor. Our average run time was 0.2 seconds.

4.3 Comparison to a Hybrid Simulated Annealing/Downhill Simplex Method

Liu et al. [12] proposed a hybrid simulated annealing/downhill simplex method in the scope of geophysics. The method was applied to the Rosenbrock function (5.13). Liu et al. [12] reported the results presented

Dimensions	Trials	F. Eval. Reported	F. Eval. HGDN	Newton. Eval. HGDN
10	2	3.0e4	17.2	255.2
50	2	8.0e5	25.0	1250.3
100	3	5.0e6	100.1	954.2

Table 5.11: Performance details reported by Liu et al. [12] for the hybrid simulated annealing/downhill simplex method and the performance details of HGDN applied to the same problems, as specified in the table.

in Table 5.11. The interval is not explicitly reported, therefore, we chose the interval to be $[-30, 30]^n$. Experiments show that the interval does not have a large impact on the performance.

5 Parallelisation of the Method

In times where processing units become rather cheaper than faster, it is important for a method to take advantage of parallel computer architectures. The presented approach offers an inherent possibility to be parallelised, since every individual can search for a local optimum independently. In the current implementation, this was achieved by using OpenMP to parallelise the loop over all individuals in Algorithm 4. The downside of this strategy is that two individuals can converge to the same optimum in one genetic step. However, since a small number of individuals can typically be chosen with the HGDN approach, the probability for this to occur is relatively low.

6 Discussion and Conclusion

We proposed a novel hybrid genetic-deflated Newton (HGDN) optimisation scheme which combines the benefits of local and global optimisation schemes, and a procedure called deflation, which can effectively remove roots from functions. The implementation of the scheme is simple, since implementations of genetic algorithms and the Newton algorithm can be reused. The proposed HGDN method leads to a significant performance gain compared to the other tested methods, namely the genetic algorithm and the Genetic-Newton algorithm, in most situations.

The HGDN method was challenged to optimise various test functions more efficiently than state-of-the-art optimisation methods. For that, we did not implement the competing methods ourselves, but used the published results to guarantee a fair comparison. The HGDN optimised the chosen test functions about ten times more efficiently. It is important to point out, however, that, though the HGDN method drastically improves on the necessary function evaluations, the method needs Newton steps (Gradient+Hessian) which can be costly to compute, depending on the problem at hand. Alternatively, the HGDN method could be executed using quasi-Newton or steepest-decent methods. In this case, Hessian computations would not be necessary but the local optimisation might require more iterations to converge. If the Hessian is computed, it can not only be used for a faster convergence, it also builds the basis for uncertainty and resolution analysis.

The success of the method depends mainly on the number of local optima within the search space and the null space at the optimum. However, even in the worst case, the proposed HGDN falls back to a Genetic-Newton method without causing any disadvantages. The method performs best compared to

other methods, when used for what it is designed for: search spaces which exhibit many local optima. It is important to note, that the superiority of the proposed method seems to benefit from higher dimensions of the search space which is linked to problems other search algorithms have in high dimensions, like the possibility for a vast plurality of local optima, which can be identified many times if deflation is not applied. An important observation is that the HGDN method performed better when the chosen number of individuals was low. The reason for this behaviour is the effect of deflation. The fewer individuals used, the faster optima can be removed from the function.

7 Acknowledgements

The work was partly funded by Kalkulo AS and the Research Council of Norway through grants 238346 and 251237, and a Centres of Excellence grant to the Center for Biomedical Computing at Simula Research Laboratory, project number 179578. The work has been conducted at Kalkulo AS, a subsidiary of Simula Research Laboratory. We want to thank Patrick Farrell for helpful comments and support.

Bibliography

[1] *Efficient global optimization using hybrid genetic algorithms*, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002.

[2] K. M. Brown and W. B. Gearhart. Deflation techniques for the calculation of further solutions of a nonlinear system. *Numerische Mathematik*, 16(4):334–342, 1971. doi: 10.1007/BF02165004.

[3] ED Cashwell and CJ Everett. Monte carlo method. *New York*, 1959.

[4] Johannes M Dieterich and Bernd Hartke. Empirical review of standard benchmark functions using evolutionary global optimization. *arXiv preprint arXiv:1207.4318*, 2012.

[5] P. E. Farrell, A. Birkisson, and S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 37:A2026–A2045, 2015. doi: 10.1137/140984798.

[6] Alex Fraser, Donald Burnell, et al. Computer models in genetics. *Computer models in genetics.*, 1970.

[7] Osman Güler. *Foundations of optimization*, volume 258. Springer Science & Business Media, 2010.

[8] Marek W Gutowski. Smooth genetic algorithm. *Journal of Physics A: Mathematical and General*, 27(23):7893, 1994.

[9] Chii-Ruey Hwang. Simulated annealing: theory and applications. *Acta Applicandae Mathematicae*, 12(1):108–111, 1988.

[10] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[11] Leo Liberti and Nelson Maculan. *Global optimization: from theory to implementation*, volume 84. Springer Science & Business Media, 2006.

[12] Pengcheng Liu, Stephen Hartzell, and William Stephenson. Non-linear multiparameter inversion using a hybrid global search algorithm: applications in reflection seismology. *Geophysical Journal International*, 122(3):991–1000, 1995.

[13] Vladimiro Miranda and Nuno Fonseca. New evolutionary particle swarm algorithm (epso) applied to voltage/var control. In *Proceedings of the 14th power systems computation conference (PSCC)*, pages 1–6, 2002.

[14] J.-M. Renders and S.P. Flasse. Hybrid methods using genetic algorithms for global optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(2):243–258, 1996. doi: 10.1109/3477.485836.

[15] Wen Wan and Jeffrey B. Birch. An improved hybrid genetic algorithm with a new local search procedure. *Journal of Applied Mathematics*, 2013, 2013. doi: 10.1155/2013/103591.

[16] J. H. Wilkinson. *Rounding errors in algebraic processes*, volume 32 of *Notes on Applied Science*. H.M.S.O., 1963.

[17] Peiliang Xu. A hybrid global optimization method: The multi-dimensional case. *Journal of Computational and Applied Mathematics*, 155(2):423–446, 2003. doi: 10.1016/S0377-0427(02)00878-6.

[18] Peiliang Xu. A hybrid global optimization method: the multi-dimensional case. *Journal of computational and applied mathematics*, 155(2):423–446, 2003.

Chapter 6

Distributed Wave-Source Optimization

Article submitted to Elsevier's *Wave Motion*

The HGDN method has proven to be efficient when applied to non-trivial benchmark functions. Now, the method is challenged to optimize a real-life functional in the frame of a Partial-Differential-Equation (PDE)-constrained optimization. There are many applications to choose from that play import roles in industry and research. One of the most complex and also useful optimization problem is to determine a distributed wave source by using only measurements of the wave outside the source.

PDE-constrained optimizations commonly need the simulation of a physical entity like electric, magnetic or gravitational fields, some temperature or material distribution or movement, or the probability distribution of an observable. These physical entities need simulations which involve the solution of a partial differential equation (PDE) of some sort. The characteristics of this solution shapes the search space of the inversion. Therefore, the search space is highly nonlinear and periodical when we use the wave equation to model the physical entity. The HGDN method was developed for problems with many optima. It is therefore well suited for inversions involving wave propagation such as distributed wave-source optimization. The idea is to find the distribution and characteristics of a wave-source just by using measurements of the wave field. Applications can be found in acoustics [34], earthquake science [15], tsunami research [19], electrodynamics [43] and even surveillance. The theory and the application is depicted in Research Paper 5.

Chapter 7

Summary and Conclusion

Article submitted to Elsevier's *Applied Geophysics*

The presented work leaves us with the following main achievements and remarks. Simulating wave fields, first arrivals, amplitudes and full wave forms, can now be done more efficiently on most computer architectures making use of parallelism where available. The division into sub-domains and the parallel solution process on sub-domain level make it possible to run large wave-motion computations in reasonable time on almost all computer architectures. Insights have been given in understanding and modeling of anisotropic wave propagation. Looking at anisotropy as a poorly chosen metric space instead of a change of propagation velocity with direction leads to a theory applicable to many kinds of anisotropies.

Function optimization is no longer an inefficient black box but rather a highly comprehensible and efficient procedure. This is achieved by carefully exploring the search space and delivering important information about its shape. This is only possible by altering certain areas of the objective function to avoid exploring the same area over and over gain. Deflation was used to achieve this goal. It was shown that the improvements in wave-propagation modeling can be combined with the proposed optimization methods to improve wave-source optimization methods.

The future of wave-propagation simulation methods will mostly be influenced by the advancements of computer architectures. Some of the methods presented in this thesis are mainly limited by the communication bandwidth between computing nodes. A faster communication could lead to a more accurate distinction between active and inactive regions of wave propagation which in turn leads to more efficient computations. Optimization methods could soon explore search spaces even more thoroughly. A mapping could be applied to not only find null-spaces but also to compute their extend. The result could help engineers to make decisions to significantly improve a design or to save costs.

A possibility how to utilize the advantages of the proposed methods can be found in the following section.

7.1 The Dawn of a New Age in Wave Imaging

The work presented in the last chapters give rise to new possibilities in fields that use any kind of wave imaging. Newest advances in wave-motion modeling allow for a never seen efficiency of wave-propagation computation. The saved time could be used to employ more sophisticated optimization methods such as

the one described in Chapter 5. Merging the new technologies would result in more comprehensive results since several solutions can be found and analyzed. Not any longer is optimization a black box operation with an uncertain outcome. How the proposed ideas can be merged to form a new generation of wave imaging tool is outlined in the next research paper.

Bibliography

[1] *Structural health monitoring of aerospace structural components using wave propagation based diagnostics*, 2012. DGZfP eV.

[2] Seyed Mohsen Sadatiyan Abkenar, Samuel Dustin Stanley, Carol J Miller, Donald V Chase, and Shawn P McElmurry. Evaluation of genetic algorithms using discrete and continuous methods for pump optimization of water distribution systems. *Sustainable Computing: Informatics and Systems*, 8:18–23, 2015.

[3] Volkan Akcelik, Jacobo Bielak, George Biros, Ioannis Epanomeritakis, Antonio Fernandez, Omar Ghattas, Eui Joong Kim, Julio Lopez, David O'Hallaron, Tiankai Tu, et al. High resolution forward and inverse earthquake modeling on terascale computers. In *Supercomputing, 2003 ACM/IEEE Conference*, pages 52–52. IEEE, 2003.

[4] Tariq Alkhalifah. An acoustic wave equation for anisotropic media. *Geophysics*, 65(4):1239–1250, 2000.

[5] Tariq Alkhalifah and Yunseok Choi. Taming waveform inversion non-linearity through phase unwrapping of the model and objective functions. *Geophysical Journal International*, 191(3): 1171–1178, 2012.

[6] RB Ashman, SC Cowin, WC Van Buskirk, and JC Rice. A continuous wave technique for the measurement of the elastic properties of cortical bone. *Journal of biomechanics*, 17(5):349–361, 1984.

[7] Aysegul Askan, Volkan Akcelik, Jacobo Bielak, and Omar Ghattas. Full waveform inversion for seismic velocity and anelastic losses in heterogeneous structures. *Bulletin of the Seismological Society of America*, 97(6):1990–2008, 2007.

[8] FE Borgnis. Specific directions of longitudinal wave propagation in anisotropic media. *Physical Review*, 98(4):1000, 1955.

[9] Dick Botteldooren. Finite-difference time-domain simulation of low-frequency room acoustic problems. *The Journal of the Acoustical Society of America*, 98(6):3302–3308, 1995.

[10] Anna R Bruss. The eikonal equation: Some results applicable to computer vision. *Journal of Mathematical Physics*, 23(5):890–896, 1982.

[11] Frederick W Byron and Charles J Joachain. Eikonal theory of electron-and positron-atom collisions. *Physics Reports*, 34(4):233–324, 1977.

[12] Gunnar Cedersund, Oscar Samuelsson, Gordon Ball, Jesper Tegnér, and David Gomez-Cabrero. Optimization in biology parameter estimation and the associated optimization problem. In *Uncertainty in Biology*, pages 177–197. Springer, 2016.

[13] Jon F Claerbout, Cecil Green, and Ida Green. *Imaging the earth's interior*, volume 6. Blackwell scientific publications Oxford, 1985.

[14] Yifeng Cui, Kim B Olsen, Thomas H Jordan, Kwangyoon Lee, Jun Zhou, Patrick Small, Daniel Roten, Geoffrey Ely, Dhabaleswar K Panda, Amit Chourasia, et al. Scalable earthquake simulation on petascale supercomputers. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–20. IEEE Computer Society, 2010.

[15] Steven M Day, Daniel Roten, and Kim B Olsen. Adjoint analysis of the source and path sensitivities of basin-guided waves. *Geophysical Journal International*, 189(2):1103–1124, 2012.

[16] Yezid Donoso and Ramon Fabregat. *Multi-objective optimization in computer networks using metaheuristics*. CRC Press, 2016.

[17] Ioannis Epanomeritakis, Volkan Akçelik, Omar Ghattas, and Jacobo Bielak. A newton-cg method for large-scale three-dimensional elastic full-waveform seismic inversion. *Inverse Problems*, 24(3): 034015, 2008.

[18] P. E. Farrell, A. Birkisson, and S. W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 37:A2026–A2045, 2015. doi: 10.1137/140984798.

[19] Yushiro Fujii, Kenji Satake, Shin'ichi Sakai, Masanao Shinohara, and Toshihiko Kanazawa. Tsunami source of the 2011 off the pacific coast of tohoku earthquake. *Earth, planets and space*, 63(7):815, 2011.

[20] Eric L Geist, Vasily V Titov, and Costas E Synolakis. Tsunami: wave of change. *Scientific American*, 294(1):56–63, 2006.

[21] Pierre Gouédard, Huajian Yao, Fabian Ernst, and Robert D van der Hilst. Surface wave eikonal tomography in heterogeneous media using exploration data. *Geophysical Journal International*, 191 (2):781–788, 2012.

[22] Samuel H Gray and William P May. Kirchhoff migration using eikonal equation traveltimes. *Geophysics*, 59(5):810–817, 1994.

[23] Bernard Hosten and Michel Castaings. Transfer matrix of multilayered absorbing and anisotropic media. measurements and simulations of ultrasonic wave propagation through composite materials. *The Journal of the Acoustical Society of America*, 94(3):1488–1495, 1993.

[24] Liangbin Hu and ST Chui. Characteristics of electromagnetic wave propagation in uniaxially anisotropic left-handed materials. *Physical Review B*, 66(8):085108, 2002.

[25] C. Kittel and P. McEuen. *Introduction to solid state physics*. Wiley new york, 1976.

[26] William La Cava, Kourosh Danai, Lee Spector, Paul Fleming, Alan Wright, and Matthew Lackner. Automatic identification of wind turbine models using evolutionary multiobjective optimization. *Renewable Energy*, 87:892–902, 2016.

[27] S. Leung, J. Qian, and R. Burridge. Eulerian Gaussian beams for high-frequency wave propagation. *Geophysics*, 72(5):SM61–SM76, 2007.

[28] Teong C Lim and GW Farnell. Search for forbidden directions of elastic surface-wave propagation in anisotropic crystals. *Journal of Applied Physics*, 39(9):4319–4325, 1968.

[29] Q Liu and YJ Gu. Seismic imaging: from classical to adjoint tomography. *Tectonophysics*, 566: 31–66, 2012.

[30] Shin-en Lo. *A Fire Simulation Model for Heterogeneous Environments Using the Level Set Method*. PhD thesis, California State University Long Beach, 2012.

[31] S. Luo, J. Qian, and H. Zhao. Higher-order schemes for 3–D first-arrival traveltimes and amplitudes. *Geophysics*, 77(2):T47–T56, 2012.

[32] Ravish Mehra, Nikunj Raghuvanshi, Lauri Savioja, Ming C Lin, and Dinesh Manocha. An efficient gpu-based time domain solver for the acoustic wave equation. *Applied Acoustics*, 73(2):83–94, 2012.

[33] A Mushtaq and HA Shah. Nonlinear zakharov–kuznetsov equation for obliquely propagating two-dimensional ion-acoustic solitary waves in a relativistic, rotating magnetized electron-positron-ion plasma. *Physics of Plasmas (1994-present)*, 12(7):072306, 2005.

[34] Philip A Nelson and Seong-Ho Yoon. Estimation of acoustic source strength by inverse methods: Part i, conditioning of the inverse problem. *Journal of sound and vibration*, 233(4):639–664, 2000.

[35] Kim B Olsen, Ralph J Archuleta, and Joseph R Matarese. Three-dimensional simulation of a magnitude 7.75 earthquake. *Science*, 270:8, 1995.

[36] D Papadopoulos and FP Esposito. Relativistic hydromagnetic wave propagation and instability in an anisotropic universe. *The Astrophysical Journal*, 257:10–16, 1982.

[37] N. Rawlinson and M. Sambridge. Seismic traveltime tomography of the crust and lithosphere. *Advances in Geophysics*, 46:81–197, 2005.

[38] J.-M. Renders and S.P. Flasse. Hybrid methods using genetic algorithms for global optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(2):243–258, 1996. doi: 10.1109/3477.485836.

[39] SI Rokhlin and L Wang. Stable recursive algorithm for elastic wave propagation in layered anisotropic media: Stiffness matrix method. *The Journal of the Acoustical Society of America*, 112 (3):822–834, 2002.

[40] Armen P Sarvazyan, Oleg V Rudenko, Scott D Swanson, J Brian Fowlkes, and Stanislav Y Emelianov. Shear wave elasticity imaging: a new ultrasonic technology of medical diagnostics. *Ultrasound in medicine & biology*, 24(9):1419–1435, 1998.

[41] Maxime Sermesant, Ender Konukoglu, Hervé Delingette, Yves Coudière, Phani Chinchapatnam, Kawal S Rhode, Reza Razavi, and Nicholas Ayache. An anisotropic multi-front fast marching method for real-time simulation of cardiac electrophysiology. In *Functional Imaging and Modeling of the Heart*, pages 160–169. Springer, 2007.

[42] J.-A. Sethian and A.-M. Popovici. 3-D traveltime computation using the fast marching method. *Geophysics*, 64(2):516–523, 1999.

[43] Fridon Shubitidze, Juan Pablo Fernández, Benjamin E Barrowes, Irma Shamatava, Alex Bijamov, Kevin O'Neill, and David Karkashadze. The orthonormalized volume magnetic source model for discrimination of unexploded ordnance. *IEEE Transactions on Geoscience and Remote Sensing*, 52 (8):4658–4670, 2014.

[44] DR Smith and D Schurig. Electromagnetic wave propagation in media with indefinite permittivity and permeability tensors. *Physical Review Letters*, 90(7):077405, 2003.

[45] Wu Song, Yong Wang, Han-Xiong Li, and Zixing Cai. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):414–431, 2015.

[46] Anastasia Spiliopoulou, Ioannis Papamichail, Markos Papageorgiou, Ioannis Tyrinopoulos, and John Chrysoulakis. Macroscopic traffic flow model calibration using different optimization algorithms. *Transportation Research Procedia*, 6:144–157, 2015.

[47] V. Červený. *Seismic ray theory*. Cambridge University Press, 2005.

[48] J. Vidale. Finite-difference calculation of travel times. *Bulletin of the Seismological Society of America*, 78(6):2062–2076, 1988.

[49] J. Vidale. Finite-difference calculations of traveltimes in three dimensions. *Geophysics*, 55(5):521–526, 1990.

[50] Robert J Young and Alexander V Panfilov. Anisotropy of wave propagation in the heart can be modeled by a riemannian electrophysiological metric. *Proceedings of the National Academy of Sciences*, 107(34):15063–15068, 2010.

[51] Linbin Zhang, James W Rector, and G Michael Hoversten. Finite-difference modelling of wave propagation in acoustic tilted ti media. *Geophysical Prospecting*, 53(6):843–852, 2005.

[52] H. Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250): 603–627, 2004.

[53] Jizhong Zhu. *Optimization of power system operation*, volume 47. John Wiley & Sons, 2015.