Computationally Efficient Methods for Seismic Modeling and Inversion

Martin Sarajærvi

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2020

UNIVERSITY OF BERGEN

Computationally Efficient Methods for Seismic Modeling and Inversion

Martin Sarajærvi

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 18.11.2020

Year: 2020

Title: Computationally Efficient Methods for Seismic Modeling and Inversion

Name: Martin Sarajærvi

Print: Skipnes Kommunikasjon / University of Bergen

Contents

Preface

This thesis is submitted as a partial fulfillment of the requirements for the degree of Philosophiae Doctor (Ph.D.) at the University of Bergen. The advisory committee consists of Henk Keers (University of Bergen), Jan Øystein Haavig Bakke (Schlumberger) and Volker Oye (Norsar).

The Ph.D. project has been financially supported by the Research Council of Norway, Petromaks2 program, grant number 245587.

Acknowledgments

First of all, I would like to thank my supervisor Henk Keers. He has patiently shared of his deep knowledge on the topics of waveform modeling and inversion. This has lead to many interesting discussions and has been of great help during my research work. Henk has also introduced me to both scientific writing and publishing, with thorough feedback and encouragement along the way.

I would also like to thank my co-supervisors Jan Øystein Haavig Bakke and Volker Oye, as well as my colleagues at Schlumberger Stavanger Research. In particular, Michael Nickel for his support throughout the research work, and the late Lars Sønneland for giving me the opportunity to pursue a PhD in Schlumberger.

Abstract

This thesis presents new methods for the modeling of seismic waveforms based on ray theory and ray-Born integrals. The waveforms are used in forward modeling, imaging and full waveform inversion and are computed in 3D acoustic as well as viscoelastic media. These tasks are computationally expensive, especially in viscoelastic media. A main focus is therefore development of efficient methods, while retaining accuracy as much as possible.

I first present a method for 3D acoustic ray-Born modeling in the time-domain using isochron surfaces. Here, integrals over isochrons are used to accurately compute seismograms for band-limited signals. This is a new development where the 3D ray-Born integral is reduced to a number of surface integrals. The surface integral formulation is optimal because the integration is only carried out for points that correspond to specific two-way traveltimes. I validate the new method by comparing ray-Born seismograms to seismograms computed with finite-differences.

An alternative approach to the time-domain technique is to compute waveforms in the frequency domain. In this case, I focus on hardware oriented methods and use the graphics processing unit (GPU) for doing efficient computations. The frequency-domain GPU method is used to not only evaluate 3D viscoelastic ray-Born integrals for modeling, but also for the computation of viscoelastic imaging integrals. The GPU method is compared to a parallel implementation on CPU and the results show significant improvements in computational efficiency.

The improved efficiency for evaluation of 3D ray-Born integrals on GPU is also of benefit for 3D full waveform inversion, a method that stands at the centre of current research in the industry to create models of the subsurface. The 3D inversion problem is analyzed by combining the GPU method with a memory efficient optimization scheme (L-BFGS). This approach allows for 3D waveform inversion on a workstation. As in the forward modeling case, I justify use of the ray-Born integrals for full waveform inversion by numerical examples and comparisons to finite-difference based inversion.

List of publications

Journal papers

- Sarajaervi, M. and Keers, H., 2018. Computation of ray-Born seismograms using isochrons. Geophysics, 83(5), T245-T256.
- Sarajaervi, M. and Keers, H., 2019. Ray-based modeling and imaging in viscoelastic media using graphics processing units. Geophysics, 84(5), S425-S436.
- Sarajaervi, M. and Keers, H., Efficient 3D viscoelastic waveform inversion using ray-Born integrals. (submitted)

Extended conference abstract

- Sarajaervi, M. and Keers, H., 2018, June. Efficient Modelling and imaging in Viscoelastic Media Using GPUs. In 80th EAGE Conference and Exhibition 2018 (Vol. 2018, No. 1, pp. 1-5). European Association of Geoscientists & Engineers.

Chapter 1

Background

The propagation of seismic waves is caused by a force such as an earthquake or an air gun (used in seismic exploration) that sets particles of a medium in motion. Below the surface, we cannot observe these motions directly because they are buried by layers of rocks. However, we can probe the motions by putting sensors in boreholes or at the surface to record seismograms.

In a typical seismic acquisition, a large number of seismograms from different locations are recorded. For example, in permanent reservoir monitoring, receivers are positioned at the seafloor with a towed source firing from a boat at the surface of the water. After the acquisition is completed, the recorded seismograms are passed through a sequence of data processing steps (e.g. Yilmaz, 2001). The end goal of these steps is to produce an image of the subsurface (e.g. Biondi, 2006) or a detailed model of the subsurface's elastic properties (e.g. Virieux and Operto, 2009).

In Figure 1.1, three standard data processing phases are shown: a signal processing phase (yellow boxes), a velocity model building phase (blue boxes), and finally, an inversion phase (gray boxes). In each phase, the individual boxes represent processing steps that are discussed in the following sections.

The goal of this chapter is to provide a general introduction to seismic processing. Special emphasis is given to the theory in the parts that discusses waveform modeling and inversion which directly concerns the work presented in this PhD thesis.

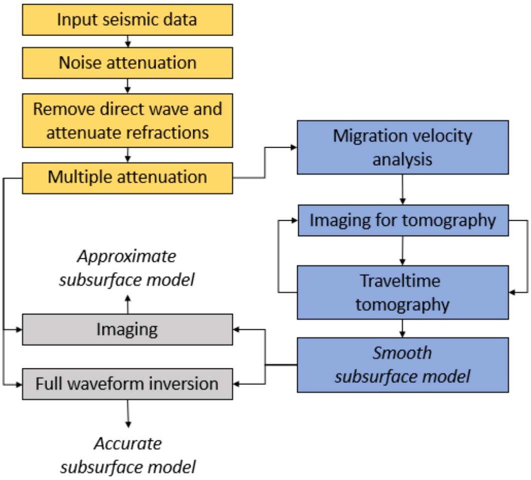


Figure 1.1: The main steps in a seismic data processing sequence, split in three phases: the signal processing phase (yellow), the velocity model building phase (blue) and finally, the imaging and full waveform inversion phase.

1.1 Signal processing

The first phase of seismic data processing is usually related to attenuation of noise and attenuation of direct, refracted and multiply scattered waves (indicated with yellow boxes in Figure 1.1). However, the exact details of the processing steps depend on the type of acquisition environment. For example, an acquisition with permanent installations on the seafloor provides a low noise environment, but may have challenges related to coarsely sampled data. A towed marine acquisition may have different challenges such as, noise related to effects such as swells, barnacles stuck to the cable or boat tug, but may have an advantage of more densely spaced receivers. It is therefore up to a processing geophysicist to make the more detailed decision of what processing steps or algorithms to use and in what order to apply them. The processing is typically done using seismic processing software, either free (Stockwell, 1997) or commercially available products.

A variety of processing algorithms exist, many which are covered in the book by Yilmaz (2001). Of great importance are the transform methods that converts seismograms from the time-space domain to some other domain. Naturally, the Fourier transform method is fundamental. First of all, for filtering of frequencies that are outside the seismic range, but also for separating dipping (coherent) noise from the primary seismic events (e.g. Lacoss et al., 1969; Canales, 1984). Furthermore, Radon transform methods (i.e. linear and parabolic) are useful (e.g. Thorson and Claerbout, 1985) for separating seismic events based on their normal moveout velocities. For example, as a technique for removal of waveforms that have traveled in the water layer (e.g., the direct wave in a marine acquisition), but also with applications for attenuation of various types of coherent noise and also multiples.

Multiples are parts of the wavefield that have been reflected more than once. They can be separated in two categories. First, the surface related multiples that come from the (relatively strong) free surface reflections. Second, the interbed related multiples that are produced by multiple reflections between seismic interfaces. As indicated in Figure 1.1, the removal of multiples (in practice this is attenuation of multiples) is a necessary, and important, processing step that needs to be done before the velocity model building phase. This is because we most often seek to only use the primary reflections when deriving a velocity model.

Different approaches are taken for attenuation of multiples, but to obtain the best possible result, it is common to test various approaches and in some cases also a combination of these. The standard approaches are predictive deconvolution (Peacock and Treitel, 1969), moveout discrimination techniques (i.e. Radon transforms) and the data-driven convolutional methods (Verschuur et al., 1992; Moore and Dragoset, 2008). When combined with adaptive subtraction techniques (Guitton and Verschuur, 2004) that are used to account for small inaccuracies in amplitudes of the multiple predictions, testing of the various approaches can be time consuming, both in terms of parameter testing and computational time. Nevertheless, this is done routinely for large seismic datasets in the industry.

After the signal processing phase is completed, seismograms are ready for the next phase which is velocity model building (indicated with blue boxes in Figure 1.1). Here the goal is to build a long-wavelength model of the seismic velocities, referred to as the background model. For this, a waveform modeling method (e.g. Carcione et al., 2002) is necessary. This is discussed in the next section.

1.2 Waveform modeling

Seismic modeling is a technique used for numerical simulation of waveforms in a model of the Earth's subsurface. For elastic waves in a heterogeneous Earth, such a model is given by the elastic tensor $\mathbf{c}$ and density ρ. In the frequency-domain, with angular frequency ω, the simulation is governed by the elastic wave equation

$$-\rho\omega^2\mathbf{G} + \nabla\cdot(\mathbf{c} : \nabla\mathbf{G}) = \mathbf{I}\,\delta(\mathbf{r}-\mathbf{s}), \tag{1.1}$$

for a point source at $\mathbf{s}$, with the wavefield $\mathbf{G}$ recorded at the receiver $\mathbf{r}$ and with $\mathbf{I}$ denoting the identity matrix (see Snieder (2001) for a detailed index notation of tensors).

For the numerical simulation, a solution to equation 1.1 is necessary. In heterogeneous media, these solutions are approximate, but various approximate methods have been developed. The two most popular methods are: ray tracing (Červený, 2005; Chapman, 2004) done in combination with Born integrals (Snieder, 2001; Dahlen et al., 2000) which is a high-frequency approximation, and finite-differences (Moczo et al., 2014; Virieux et al., 2016) which is considered approximate in the sense that numerical derivatives are used. The research work in this thesis is devoted to the ray method and the Born integral which is discussed in more detail in the next sections.

1.2.1 Born integrals

By the decomposition of $\mathbf{c}$ and ρ into a slowly varying background and a rapidly varying term, respectively, denoted by subscripts 0 and 1, we have

$$\mathbf{c} = \mathbf{c}_0 + \mathbf{c}_1 \tag{1.2}$$

$$\rho = \rho_0 + \rho_1 \tag{1.3}$$

where from perturbation theory it follows that, to first-order, an approximate solution to equation 1.1 is

$$\mathbf{G} = \mathbf{G}_0 + \mathbf{G}_1. \tag{1.4}$$

This is the Born approximation where $\mathbf{G}_0$ is a solution to the wave equation in the background medium and $\mathbf{G}_1$ is the scattered wavefield, written in terms of $\mathbf{G}_0$ as:

$$\begin{aligned} \mathbf{G}_1(\mathbf{r},\mathbf{s},\omega) = & \,\omega^2 \int \mathbf{G}_0^{\mathrm{T}}(\mathbf{r},\mathbf{x})\rho_1(\mathbf{x})\mathbf{G}_0(\mathbf{x},\mathbf{s})\,\mathrm{d}\mathbf{x} \\ & + \int \nabla\mathbf{G}_0^{\mathrm{T}}(\mathbf{r},\mathbf{x}) : \mathbf{c}_1 : \nabla\mathbf{G}_0(\mathbf{x},\mathbf{s})\,\mathrm{d}\mathbf{x}. \end{aligned} \tag{1.5}$$

This expression is of foremost importance in geophysics (Hudson and Heritage, 1981; Tarantola, 1984b, 1986; Snieder, 2001; Ikelle and Amundsen, 2005), in particular for the seismic inverse problem, because it gives a linear relationship between an Earth model and the scattered wavefield. In equation 1.5, the waveform $\mathbf{G}_0(\mathbf{x},\mathbf{s})$ propagates in the background medium from the source located at $\mathbf{s}$ to a scatterer at point $\mathbf{x}$, interacts with the scatterer, and then propagates as $\mathbf{G}_0(\mathbf{r},\mathbf{x})$ from the scatterer to the receiver at $\mathbf{r}$. The scatterer is either a density perturbation or a perturbation in the elastic parameters. The total scattered wavefield $\mathbf{G}_1$ is found by summation of the integrand over all scattering points $\mathbf{x}$, but to compute this, an expression for $\mathbf{G}_0$ is necessary. This is found using ray theory.

1.2.2 Ray theory

Seismic ray theory is an approximate solution to the wave equation in slowly varying media where we consider the wavefront to be separated into individual elementary waves that are described by individual rays. This separation allows for tracking of the wavefield using amplitudes and travel times along the ray paths. The advantage of this is that waveforms can be computed for specific seismic events at a very low computational cost. This allows for interpretation of specific seismic interfaces or specific events in a seismogram, but also, rays can be computed over a full volume of the subsurface.

Ray theory is extensively covered in the literature, see for example the books by Červený (2005) and Chapman (2004) for details and further discussions. Alternatively, the review articles by Gjøystdal et al. (2002); Virieux and Lambaré (2007); Červený et al. (2007) and references therein.

Improved accuracy in waveform modeling is achieved if attenuation effects are included (Carcione, 2015). The media is then viscoelastic. Normally, viscoelastic modeling is done by using the quality factor Q. This can be a constant, for example based on regional knowledge, or a heterogeneous Q model can be estimated using imaging methods (Ribodetti and Virieux, 1998) and tomographic methods (Dutta and Schuster, 2016). For extensions of ray theory in viscoelastic media, see the articles on complex rays by Kravtsov et al. (1999); Hanyga and Seredyńska (2000). Furthermore, the work on real ray tracing by Gajewski and Pšenčík (1992); Vavryčuk (2008) is useful in both smooth weakly attenuating and anisotropic media. Another practical alternative is to include attenuation effects only in the travel times (e.g. Dahlen et al., 2000; Keers et al., 2001). In this case, the attenuation effects do not have an effect on the ray geometry. Still, the method is effective in weakly attenuating media (i.e. $Q \geq 40$) as shown by examples in Sarajaervi and Keers (2019)

As indicated in the last section, the ray method plays an important role in modeling and inversion when ray theoretical expressions are employed in the Born integral 1.5. This is then called ray-Born modeling (see, for example, Moser (2012) for a review) and allows for computation of the scattered wavefield in slowly varying (smooth) heterogeneous Earth models.

1.2.3 Ray-Born integrals

By employing ray theoretical expression for $\mathbf{G}_0$, combined with evaluation of $\nabla\mathbf{G}_0$ in the high-frequency approximation, the ray-Born integral is given as

$$\mathbf{G}_1(\mathbf{r},\mathbf{s},\omega) = \int [\mathbf{\Omega}(\mathbf{x}) \cdot \mathbf{m}_1(\mathbf{x})]\mathbf{v}(\mathbf{r},\mathbf{s},\mathbf{x},\omega)\,\mathrm{d}\mathbf{x}. \tag{1.6}$$

where $\mathbf{m}_1$ is the perturbation vector, $\mathbf{\Omega}$ is a vector containing the Rayleigh factors and $\mathbf{v}$ is a vector containing the ray quantities

$$\mathbf{v}(\mathbf{r},\mathbf{s},\mathbf{x},\omega) = A_{rs}(\mathbf{x}) \exp[-i\omega T_{rs}(\mathbf{x},\omega)]\mathbf{p}_r(\mathbf{x})\mathbf{p}_s(\mathbf{x}). \tag{1.7}$$

See Zhao and Dahlen (1996), Dahlen et al. (2000) and Snieder (2001) for detailed derivations of $\mathbf{v}$. In equation 1.7, $\mathbf{p}_r$ is a polarization vector corresponding to a waveform arriving at the receiver and $\mathbf{p}_s$ is a polarization vector corresponding to a waveform leaving the source. The term

$$A_{rs}(\mathbf{x}) = A(\mathbf{r}, \mathbf{x})A(\mathbf{s}, \mathbf{x}) \tag{1.8}$$

contains geometrical spreading, group velocities and densities in the amplitude term A, from a source position $\mathbf{s}$ to a receiver position $\mathbf{r}$ and then to a scattering position $\mathbf{x}$. Similarly, the travel time function is given as

$$T_{rs}(\mathbf{x}, \omega) = T(\mathbf{r}, \mathbf{x}, \omega) + T(\mathbf{s}, \mathbf{x}, \omega). \tag{1.9}$$

Here I allow for the complex valued and frequency-dependent travel times T (see Keers et al., 2001, equation 1) where the quality factor is used to describe attenuation in the background model.

As presented here, it is not made explicit whether the background medium is isotropic or anisotropic. However, note that in the (weakly) anisotropic case, I speak of quasi-P and quasi-S waves because the polarization vectors $\mathbf{p}$ are not in the direction of wave propagation. In this case, the terms in $\mathbf{v}$ then have a slightly different meaning, compared to the isotropic case. Details on anisotropic ray theory is covered by Červený (2005) and references therein.

Finally, the Rayleigh coefficients $\mathbf{\Omega}$ and perturbations $\mathbf{m}_1$ are necessary in order to describe the scattering using equation 1.6. In terms of wave velocities α, β for respectively P- and S-waves in isotropic media, the scattering $\mathbf{\Omega} \cdot \mathbf{m}_1$ term is

$$(\Omega_\alpha, \Omega_\beta, \Omega_\rho) \cdot (\Delta\alpha, \Delta\beta, \Delta\rho) \tag{1.10}$$

with the coefficients given by Dahlen et al. (2000). Similarly, for the anisotropic case

$$(\Omega_\alpha, \Omega_\beta, \Omega_\rho, \Omega_\delta, \Omega_\varepsilon, \Omega_\gamma) \cdot (\Delta\alpha, \Delta\beta, \Delta\rho, \Delta\delta, \Delta\varepsilon, \Delta\gamma) \tag{1.11}$$

where δ, ε and γ denote the weakly anisotropic parameters (Thomsen, 1986) and with the coefficients detailed by Calvet et al. (2006) for the transversely isotropic case.

1.2.4 Ray-Born seismograms

In the band-limited case, a source-time function $\mathbf{f}$ (with forces in three directions) is convolved with the Green functions to compute seismograms

$$\mathbf{u}_0(\mathbf{r}, \mathbf{s}, \omega) + \mathbf{u}_1(\mathbf{r}, \mathbf{s}, \omega) = [\mathbf{G}_0(\mathbf{r}, \mathbf{s}, \omega) + \mathbf{G}_1(\mathbf{r}, \mathbf{s}, \omega)]\mathbf{f}(\omega). \tag{1.12}$$

The convolution is done in the frequency domain and after that the seismograms transformed to the time-domain using the Fourier transform.

An example of simulated viscoelastic seismic data is shown in Figure 1.2 for a synthetic 3D isotropic heterogeneous SEG/EAGE Overthrust model (Aminzadeh et al., 1997). The simulation is done with a source acting at the center of the model and with receivers positioned in a plane at the surface. The direct P-wave arrives first with a quite strong amplitude. This is followed by PP, PS, and SP scattered waves that are, by nature, weaker. Below that, a timeslice is positioned approximately at the arrival of the direct S-wave.

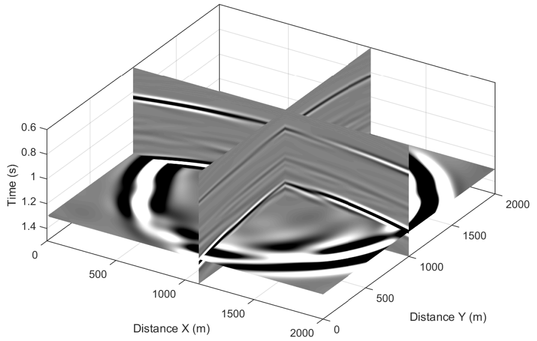


Figure 1.2: Record sections and timeslice of 3D viscoelastic waveforms, recorded in a plane of receivers on the surface.

It should be noted that, many times, we need to consider only the elastic isotropic case. This happens when we have limited knowledge of anisotropy and attenuation effects. Moreover, in a marine acquisition, we may consider only compressional waves and use acoustic waveforms. These waveforms are relatively cheap to compute compared to the more accurate anisotropic and viscoelastic waveforms and are therefore popular.

1.2.5 Finite-difference

Compared to ray based solutions to the wave equation, finite-differences solves the wave equation in another way by using a fully numerical approach. Here, derivatives of the wave equation are computed numerically and modeling is done by forward propagating the wavefield step by step in time. See e.g. Robertsson et al. (1994); Moczo et al. (2007, 2014); Virieux et al. (2016) for further discussion.

Waveforms are also in this case approximate because the numerical derivatives are not exact and inaccuracies can, for example, occur at curved interfaces in the model (which is important in practice for strongly folded areas and salt domes). However, the method can be applied in more general media compared to ray theory and complicated wave propagation effects, such as multiple scattering and multipathing are accounted for without special treatment. This is a great advantage of the method and is mainly what separates the method from other solutions. Unfortunately, the time stepping scheme is quite expensive to compute because a large number of parameters needs to be calculated at each time step. This is a limitation that can become problematic in practice. In particular, when waveforms needs to be computed for relatively high frequencies (> 30 Hz) in 3D media.

Comparing the ray-based modeling methods to finite-differences is useful in numerical examples. Not only for validation of new codes, but also for analysis of the effects of multipathing and multiple scattering.

1.3 Velocity model building

In this and the following sections, I discuss three types of velocity models that are used in seismic exploration. First, a relatively rough, but smooth velocity model that is obtained by measuring signal coherency along hyperbolic trajectories on common-midpoint gathers. A process referred to as migration velocity analysis (e.g. Symes, 2008). Second, a smooth, but improved migration velocity model, constrained by travel time tomography (Woodward, 1992). Third, a rapidly varying velocity model (high-resolution) obtained from full waveform inversion (Pratt, 1999; Virieux and Operto, 2009). The two first models represent the so-called background model (see equation 1.2). The third represents the perturbations.

In Figure 1.1, the first two models are derived in the velocity model building phase (indicated with blue boxes), discussed in this section. The third type is derived using either the ray-based or finite-difference based full waveform inversion (indicated with gray boxes) which are discussed in the next section.

Migration velocity analysis begins by computing semblance panels (Yilmaz, 2001) for common-midpoint gathers on a coarse surface grid that covers the seismic survey area. For each semblance panel, a root-mean-square velocity profiles in 1D are picked based on high coherency values. This is done either manually or using automated methods (Adler and Brandwood, 1999; Fabien-Ouellet and Sarkar, 2020) that are often necessary due to the sheer size of seismic surveys. After that, the velocity profiles are converted to interval velocities in depth and smoothed in 3D. The result of this process is a model that acts as the starting point for both travel time tomography and imaging.

The method of travel time tomography seeks to update the migration velocity model by minimizing residual moveout on common-image gathers. This is achieved by an iterative inversion that, in simplified terms, can be described in two steps (see Figure 1.1). First, ray-based seismic imaging is done in order to produce the common-image gathers. Second, the tomographic inverse problem is solved by analyzing residual moveout on the gathers, which combined with ray tracing in the latest background model, produces a large linear system of equations. By solving these equations in terms of the background model, the residual moveout becomes smaller and the initial rough migration model is improved incrementally for each tomography iteration. For more details on this approach, see Woodward et al. (2008) and the references therein.

The velocity model building described here is computationally expensive. To some extent because the tomography method requires computation of a very large inverse problem, but mainly because the tomography loop requires imaging using the latest background model, at each iteration.

1.4 Waveform inversion

After the velocity model building phase, the final background model is used to compute a final image or model of the subsurface. In this inverse problem, an accurate background model (for example obtained from tomography) increases the chance of achieving a successful inversion result. Both in terms of spatial resolution and correct positioning of seismic events in depth.

For the purpose of discussing inversion, I begin by introducing the operator L and write the ray-Born integral in equation 1.6 as

$$\mathbf{u}_1 = L(\mathbf{m}_1), \tag{1.13}$$

mapping model perturbations $\mathbf{m}_1$ to seismograms $\mathbf{u}_1$, simplifying the integral equation in a more compact form. Because equation 1.13 is a linear transform, the expression can also be written, after discretization, in matrix notation as

$$\mathbf{u_1} = \mathbf{L}\mathbf{m}_1. \tag{1.14}$$

In this case, $\mathbf{m}_1$ is a vector covering a 2D- or 3D-space, and $\mathbf{u}_1$ is a vector that contains seismograms for all sources, receivers and frequencies. Consequently, the matrix $\mathbf{L}$ is very large and in general not invertible. This is addressed in the next sections.

Seismic inversion is often put in two categories. First, imaging, also called prestack depth migration, which is a relatively efficient, but approximate solution to the inverse problem. Second, full waveform inversion which is the more accurate, but also a more computationally expensive inversion method. Both methods are indicated with gray boxes in Figure 1.1 and are discussed in more detail in the sections below.

In addition to these two categories, inversion methods are often classified by the modeling method that is employed in the inverse operator. For example, ray-Born and Kirchhoff imaging/inversion methods (Beylkin, 1985; Bleistein, 1987) are based on ray theory, beam methods are based on ray and beam theory (Hill, 2001), while the so-called wave equation methods are based one-way wave propagators and a finite-difference based method is called reverse time migration (see Mulder and Plessix (2004) for comparisons). The next section is devoted to inverse operators based on the ray-Born integral 1.6 and solutions to equation 1.14.

1.4.1 Least-squares objective function

A natural way to solve the inverse problem of finding $\mathbf{m}_1$ given some $\mathbf{u}_1$, is to define an objective function C using the squared difference between observed seismograms $\mathbf{d}$ and ray-Born seismograms $\mathbf{u}_1$:

$$C(\mathbf{m}_1) = \frac{1}{2}||\mathbf{u}_1 - \mathbf{d}||^2. \tag{1.15}$$

A direct solution to equation 1.15, in a least-squares sense, is obtained when $\nabla C = \mathbf{0}$:

$$\mathbf{L}^{\mathrm{T}}\mathbf{L}\mathbf{m}_1 = \mathbf{L}^{\mathrm{T}}\mathbf{d}, \tag{1.16}$$

where $\mathbf{L}^T$ is the conjugate transpose of $\mathbf{L}$. Re-arranging the terms of equation 1.16 then gives the normal equations as

$$\mathbf{m}_1 = \mathbf{H}^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{d}, \tag{1.17}$$

where it is assumed that the Hessian $\mathbf{H} = \mathbf{L}^{\mathrm{T}}\mathbf{L}$ is invertible. Unfortunately, as with the direct inversion of $\mathbf{L}$, this assumption is problematic because $\mathbf{H}$ is too large for most practical problems (mainly in terms of computer memory) and also singular, without a regular inverse.

1.4.2 Imaging

To overcome the problems related to inversion of the Hessian, I use approximations to $\mathbf{H}$ that allow us to compute an approximate inverse solution. One way to solve equation 1.17 is to use a diagonal approximation by rewriting the equation as

$$\mathbf{m}_1 \approx \mathbf{D}^{-1}\mathbf{L}^{\mathrm{T}}\mathbf{d}. \tag{1.18}$$

With a regularization parameter λ and an identity matrix $\mathbf{I}$, the diagonal matrix $\mathbf{D}$ is then

$$\mathbf{D} = \mathrm{diag}(\mathbf{L}^{\mathrm{T}}\mathbf{L}) + \lambda\mathbf{I}. \tag{1.19}$$

This is called imaging (cf. integral equation 9 in Sarajaervi and Keers, 2019) and is less accurate than what can be achieved if off-diagonal elements of $\mathbf{H}$ can be taken into account. However, the diagonal imaging approximation 1.19 is computationally efficient and the results are useful for structural interpretation, the tomographic inversion process and also for amplitude studies where relative amplitude values are compared.

Although the diagonal Hessian approximation is useful (Tarantola, 1984b; Beydoun and Mendes, 1989), it is possible to design so-called asymptotic inversion operators that also take elements in the vicinity of the diagonal of $\mathbf{H}$ into account. This was presented by Beylkin (1985); Beylkin and Burridge (1990) who derived a quite accurate approximation for seismic imaging using an asymptotic Hessian. Their work was continued by other authors who focused on improving accuracy using an iterative asymptotic approach (Jin et al., 1992) that has, with later developments, been demonstrated for practical applications (Thierry et al., 1999; Lambaré et al., 2003; Operto et al., 2003). Both the diagonal and asymptotic Hessian approximations are routinely employed in the industry.

Figure 1.3 shows an imaging result using a synthetic example. A true perturbation model (Figure 1.3a) is used to produce synthetic seismograms with finite-differences. These seismograms are then used in equation 1.18 to produce an approximate image $\mathbf{m}_1$.

1.4.3 Full waveform inversion

Since the early works by Lailly (1983) and Tarantola (1984a, 1986, 1988), the optimization approach to seismic inversion has proven useful (Pratt, 1999; Virieux and Operto, 2009). This is called full waveform inversion and a successful full waveform inversion results goes beyond what can be achieved with imaging and also gives an opportunity to accurately de-couple the various elastic parameters and the density.

As an alternative to the approach of using the normal equations (discussed above), the full waveform inversion method attempts to minimize the objective function 1.15 in the framework of iterative optimization (see, for example Tarantola, 2005). In this case, I seek roots of C in the vicinity of $\mathbf{m}_1$ by

$$\mathbf{m}_1^{(k+1)} = \mathbf{m}_1^{(k)} + \theta \Delta \mathbf{m}_1, \tag{1.20}$$

where θ is a parameter for the step-length (see Nocedal and Wright, 2006, chapter 3) and $\Delta\mathbf{m}_1$ is a step-direction for the kth iteration. The method starts with an initial model $\mathbf{m}_1^{(0)}$ containing zeros, or the diagonal approximation 1.18, and continues by making incremental updates to $\mathbf{m}_1$ in the direction

$$\Delta \mathbf{m}_1 = \mathbf{B}^{-1} \nabla \mathbf{C}. \tag{1.21}$$

The step directions can be purely gradient based (with $\mathbf{B}$ as the identity), but higher accuracy is obtained by using quasi-Newton methods (Brossier et al., 2009; Pan et al., 2017) where also the Hessian operator is taken into account. In equation 1.21, $\mathbf{B}$ denotes a quasi-Newton Hessian approximation (see Nocedal and Wright, 2006, chapter 6) and the gradient $\nabla \mathbf{C}$ is given by the ray-based Fréchet derivatives

$$\nabla C = \Re\{\mathbf{L}^{\mathrm{T}}(\mathbf{d} - \mathbf{L}\mathbf{m}_1)\} \tag{1.22}$$

(cf. Pratt et al., 1998, equation 10). Even though $\mathbf{L}$ also in this case covers a large parameter space, the practical memory requirements needed to compute ∇C are low. This is because the two matrix vector products in equation 1.22 are both computed without the need to form the $\mathbf{L}$ explicitly (Sarajaervi and Keers, 2019).

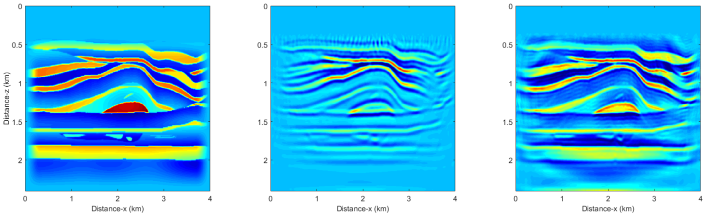
To compute $\mathbf{B}^{-1}$ in equation 1.21 for the parameter space covered by $\mathbf{L}$, the so-called limited-memory algorithms (see Nocedal and Wright, 2006, chapter 7) can be applied. A popular variant is the limited-memory BFGS method

(Byrd et al., 1995) where **B** is a compact approximation of the Hessian. The approximation is computed by storing a set of vectors that represent curvature information from the most recent iterations.

Finally, note that when ray-Born integrals are employed in the optimization scheme, the observed seismograms **d** should contain only the singly scattered waves. As shown in Figure 1.1, this is readily available after the multiple attenuation step that is done in preparation for velocity model building.

1.4.4 Synthetic example

Figure 1.3c shows a full waveform inversion result that uses the same input data as that used for imaging in Figure 1.3b. As expected, this demonstrates that full waveform version ($k = 25$ iterations) produces a detailed perturbation model compared to imaging and the true perturbation model (in Figure 1.3a).



(a)

(b)

(c)

Figure 1.3: A synthetic example of seismic imaging and inversion where a) is the true velocity perturbation model, b) is the imaging result and c) is a the full waveform inversion result after 25 iterations.

1.5 Computational aspects

A lot of effort in geophysics has focused on development of the theoretical background for seismic algorithms. This is important, but to make use of these theoretical developments, we need to move from theory to numerical implementations. In this process, it makes sense for the numerical implementations to take full advantage of hardware developments that can be used to run computations in parallel. Roughly speaking, there are two ways of doing this when considering a single computer. Either by using central processing units (CPUs) with multiple cores, or by using graphics processing units (GPUs).

A typical computer, by today's standard, has a CPU with multiple cores (8 - 64 cores, depending on hardware). These cores can run computations in parallel, but it is up to the programmer to make this happen (e.g. Williams, 2012). For some problems in seismic processing, this parallel programming is easy. For example, shot gathers can be processed in parallel during noise removal or during multiple attenuation when one shot does not depend on other shots. However, other problems may require more sophisticated parallel programming. One example is finite-difference based modeling which can be made parallel over the spatial domain. Here, data must be shared on the edges of each sub-domain leading to a complicated programming problem. In general, we therefore consider parallel programming to be more difficult and time-consuming compared to the standard sequential programming. Still, these efforts are necessary for computational problems that may run for days or weeks because, for many problems, efficiency can be increased by a factor that is close to the number of cores with a parallel implementation.

Another type of parallel hardware is the GPU. A GPU has less powerful compute cores than that of a CPU, but instead it has a greater number of cores (approximately 1000 - 3500 cores, depending on hardware). This is a benefit for computational problems that can be split in many small parts where each part runs in parallel. Broadly speaking, recognizing these parts is the main challenge of efficient GPU programming. If handled carefully, this type of parallel programming is, for many problems (including seismic), the most efficient way of doing hardware acceleration for algorithms.

For the seismic algorithms that are discussed above, numerical implementations of the forward modeling problem in equation 1.6, the imaging problem in equation 1.18 and the full waveform inversion problem in equation 1.22 should be implemented using a parallel strategy. If this is not done, it is not practical to solve the equations in 3D media, especially in elastic and viscoelastic cases.

1.6 Scientific contributions

The focus of the work presented in this thesis is the development and analysis of ray-Born integrals that are used in seismic modeling, imaging and full waveform inversion. As indicated in the above sections and in Figure 1.1, these are fundamental topics in seismic data processing and geophysical interpretation. Of course, numerical solutions to these problems have been successfully applied in the industry for many years, but there is still a need for analysis of the methods and new developments that can increase both accuracy and computational efficiency. My scientific contributions to these topics are presented in the form of three scientific papers.

In the first paper, I present a new method for the computation of ray-Born integrals in the time-domain. Here I address issues related to numerical dispersion and efficiency in seismic modeling by carrying out numerical integration of ray-Born integrals along isochron surfaces of two-way travel time.

The second paper analyses a method for computing both ray-Born and imaging integrals in the frequency-domain. In this case, the numerical integration comes at a larger computational cost compared to the time-domain implementation as large multi-dimensional integrals are evaluated repeatedly (at each frequency), but the advantage is that frequency dependent attenuation effects can more easily be included. To increase efficiency for these computations, I have developed a method for solving the numerical integration problem using GPUs.

In the last paper, I use the GPU method to solve the 3D viscoelastic full waveform inversion problem in the frequency domain. Here I present a scheme that is practical for inversion of large parameter spaces by combining the GPU method with a limited-memory BFGS method for the numerical optimization.

The following chapter gives a more detailed summary of each of these papers.

Chapter 2

Summary of publications

The scientific contributions of this dissertation comes in the form of two published journal papers (Sarajaervi and Keers, 2018a, 2019) and one submitted journal paper. The second paper (Sarajaervi and Keers, 2019) was also presented as an extended abstract (Sarajaervi and Keers, 2018b) at the international EAGE conference.

Below is a brief summary of the three journal papers. After that, the papers are attached in its original form.

2.1 Paper I

Title:	Computation of ray-Born seismograms using isochrons
Authors:	Martin Sarajaervi, Henk Keers
Journal:	Geophysics
DOI:	10.1190/geo2017-0669.1

Paper I concerns acoustic ray-Born modeling in the time-domain and here I derive a new band-limited form of the ray-Born integral where the waveforms are modelled accurately by having an integration grid that closely follows curved isochrons. This is an optimal approach with respect to how many scattering points are needed for the computation of the wavefield at a specific time and because the continuous ray-Born integral is expressed as a band-limited ray-Born integral, numerical dispersion is avoided.

The main result of the paper is the derivation of new expressions for the surface integrals, but I also investigate the accuracy of ray-Born modeling by comparing ray-Born seismograms to finite-difference seismograms. The comparisons are done in random models with varying perturbation strength and realistic models with a heterogeneous background model. Moreover, the comparisons demonstrate that for weak perturbations ray-Born and finite-difference seismograms are identical and for strong perturbations complicated wave propagation effects such as multiple scattering and multipathing, as expected, produce more complicated seismograms in the case of finite-differences.

2.2 Paper II

Title:	Ray-based modeling and imaging in viscoelastic media using graphics processing units
Authors:	Martin Sarajaervi, Henk Keers
Journal:	Geophysics
DOI:	10.1190/geo2018-0510.1

There are two main challenges with the time-domain implementation presented in Paper I. One challenge is inclusion of frequency dependent attenuation effects. Another is the computational cost of obtaining accurate isochron surfaces. In paper II, I therefore study viscoelastic waveform modeling in the frequency domain. Unfortunately, computing ray-Born integrals in the frequency-domain is computationally expensive compared to the time-domain. This is because, in 3D, the ray-Born integrals are volume integrals that needs to be computed for every frequency in the signal. The numerical integration must therefore be fast. To address this, I make use of the parallel computational power of GPUs for the numerical integration. The GPU implementation evaluates the ray-Born integrand in parallel and computes partial sums in a tree structure over the GPU cores. In the same way, the algorithm is used to evaluate viscoelastic imaging integrals.

The main contribution of Paper II is the unified GPU approach for both efficient 3D viscoelastic modeling and imaging, but the paper also compares computational performance of the GPU implementation against a parallel CPU implementation. This is done using numerical examples showing that, against a 'standard' compute node (8 CPU cores), an increase in performance of up to 48 times is achieved by using GPUs for imaging. Similarly, an increase in performance of up to 54 times is achieved for modeling.

2.3 Paper III

Title:	Efficient 3D viscoelastic waveform inversion using ray-Born integrals
Authors:	Martin Sarajaervi, Henk Keers
Journal:	Submitted

Paper III is an extension of the work in Paper II, demonstrating how the GPU method can be used in 3D viscoelastic full waveform inversion. The development from modeling and imaging to full waveform inversion is natural. First, because modeling is used directly in the waveform inversion optimization scheme. Second, because imaging and the computation of gradients are similar operations.

The iterative optimization scheme uses an approximate Hessian based on the limited-memory BFGS algorithm (Nocedal and Wright, 2006). In this case, the first L-BFGS updates are closely related to a band or diagonal approximation of the Hessian (Tarantola, 1984a; Beydoun and Mendes, 1989; Beylkin, 1985; Jin et al., 1992), while subsequent L-BFGS updates incrementally add more details to the inversion result as long as convergence is achieved. In principle, the combination of the memory efficient L-BFGS method and the GPU based computation of waveforms are the necessary ingredients for doing 3D viscoelastic full waveform inversion on a workstation.

Analysis of the ray-based full waveform inversion is done by computing numerical examples for a set of perturbation models of varying strength (relative to a fixed background model). The examples demonstrate how stronger perturbations lead to increasingly complicated wave propagation effects (modelled with finite-differences) and how this influences accuracy of the inversion.

Chapter 3

Outlook

A major part of the work presented in this thesis has been focused on computational efficiency and I have analyzed how these developments are useful for three important applications in geophysics: modeling, imaging and full waveform inversion. The development of efficient methods is of course useful for practical applications, but it also enables more efficient testing of the methods. This is an advantage for further research where several interesting aspects of ray theory can be analysed in more detail.

For example, we can increase accuracy by taking multipathing into account using beam (Červený et al., 1982) or Maslov (Chapman and Drummond, 1982) extensions of ray tracing. Moreover, anisotropic background models are important for practical inversion or imaging when longer offset data is available. Anisotropic ray-Born integrals should therefore be analysed using either anisotropic ray theory as presented by Červený (2005), first-order anisotropic ray theory by Pšenčík and Farra (2005); Farra (2005) or by adopting the anisotropic viscoelastic real ray theory presented by Vavryčuk (2008).

Below I give an outlook of other, more specific, research ideas that follow up on each of the papers that are presented in this thesis.

3.1 Efficient extraction of isochrons

The most important extension to the isochron method in paper I concerns how isochrons can be extracted efficiently. For this purpose, I have explored two directions.

In the first direction I used dynamic ray tracing to extract isochrons directly in the paraxial ray approximation (Červený et al., 1984). This is similar to what is done by Červený and Soares (1992) for the computation of Fresnel volumes and by Dahlen et al. (2000) for the computation of sensitivity kernels for finite-frequency travel times. The preliminary studies showed that for 3D heterogeneous background models the method is very efficient, but the relatively poor accuracy of the paraxial method (except in very close vicinity of the main ray) in these media prohibits applications of the method to seismic exploration. However, there may be applications for relatively simple models, and for transmission problems, where high accuracy is only required within the Fresnel volume.

The second direction is to use the so-called isochron rays (Iversen, 2004) in combination with a 'wavefront triangulation' procedure (e.g. Vinje et al., 1993) applied for isochrons. This gives the isochron directly without the need for ray tables on a regular grid and directly provides the surface needed for numerical integration with the ray-Born integral in Sarajaervi and Keers (2018a). The preliminary studies on this topic were encouraging with respect to computational efficiency, but also in this case the 3D heterogeneous background model poses a challenge for accuracy (although they are smooth enough for ray tracing). This is most likely because a first-order approximation is used in the formulation of isochron rays.

I have only briefly tested these methods with my own numerical implementations. A more systematic study for accuracy of these methods should therefore be done. In particular, for the more realistic seismic exploration models such as a smooth version of the 3D overthrust model (Aminzadeh et al., 1997).

3.2 Ray tracing on GPU

To leverage the full capacity of a GPU with the method presented in paper II, ray tracing should be done directly on the GPU. Not surprisingly, the GPU is very well suited for this task. My preliminary numerical experiments with parallel ray tracing on the GPU have demonstrated a formidable speed-up compared to CPU implementations. However, more work needs to be done on the interpolation of results for integration with, for example, the imaging integrals in Sarajaervi and Keers (2019).

If this integration is achieved, an advantage to the approach is that the need to transfer data between GPU and CPU memory is eliminated as all ray data is contained in the GPU memory. This also avoids having to read pre-computed ray tables from a disk or network connection which tends to be a bottle-neck for both the modeling and imaging. Instead ray tracing is done 'on the fly'. For example, by doing parallel ray tracing (over a variety of take-off angles) for just the source-receiver rays and scattering point in question, and then discarding the ray data after computing the ray-Born or imaging integrals.

3.3 Multiparameter inversion

Multiparameter waveform inversion that inverts for both density and elastic parameters is a straight-forward extension of the research presented in Paper III. It is mostly a matter of implementation, because the theory is already presented for multiple parameters and in principle, this can be handled directly by the L-BFGS based inverse Hessian.

Another natural next step for the ray-Born full waveform inversion is to apply the method to real data. This is necessary to understand more about how the method stands against the more common finite-difference based waveform inversion methods. Such a project would require resources, in terms of compute power, and a good quality seismic dataset that is pre-processed for the inversion and is therefore resource intensive. In this case, an accurate anisotropic background model is likely needed to achieve good estimates of perturbations in density, velocity and the weakly anisotropic parameters. See discussion in Virieux et al. (2017) and references therein.

Bibliography

Adler, F. and Brandwood, S. (1999). Robust estimation of dense 3D stacking velocities from automated picking. In *SEG Technical Program Expanded Abstracts 1999*, pages 1162–1165. Society of Exploration Geophysicists.

Aminzadeh, F., Jean, B., and Kunz, T. (1997). *3-D salt and overthrust models.* SEG/EAGE 3-D Modeling Series No. 1, Society of Exploration Geophysicists.

Beydoun, W. B. and Mendes, M. (1989). Elastic ray-Born l_2-migration/inversion. *Geophysical Journal International*, 97(1):151–160.

Beylkin, G. (1985). Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized Radon transform. *Journal of Mathematical Physics*, 26(1):99–108.

Beylkin, G. and Burridge, R. (1990). Linearized inverse scattering problems in acoustics and elasticity. *Wave motion*, 12(1):15–52.

Biondi, B. (2006). *3D seismic imaging.* Number 14. Society of Exploration Geophysicists Tulsa.

Bleistein, N. (1987). On the imaging of reflectors in the Earth. *Geophysics*, 52(7):931–942.

Brossier, R., Operto, S., and Virieux, J. (2009). Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion. *Geophysics*, 74(6):WCC105–WCC118.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Calvet, M., Chevrot, S., and Souriau, A. (2006). P-wave propagation in transversely isotropic media: I. finite-frequency theory. *Physics of the Earth and Planetary Interiors*, 156(1-2):12–20.

Canales, L. L. (1984). Random noise reduction. In *SEG Technical Program Expanded Abstracts 1984*, pages 525–527. Society of Exploration Geophysicists.

Carcione, J. M. (2015). *Wave fields in real media: Wave propagation in anisotropic, anelastic, porous and electromagnetic media.* Elsevier.

Carcione, J. M., Herman, G. C., and ten Kroode, A. P. E. (2002). Seismic modeling. *Geophysics*, 67(4):1304–1325.

Červený, V. (2005). *Seismic Ray Theory.* Cambridge University Press, Cambridge UK.

Červený, V., Klimeš, L., and Pšenčík, I. (1984). Paraxial ray approximations in the computation of seismic wavefields in inhomogeneous media. *Geophysical Journal International*, 79(1):89–104.

Červený, V., Klimeš, L., and Pšenčík, I. (2007). Seismic ray method: Recent developments. *Advances in Geophysics*, 48:1–126.

Červený, V., Popov, M. M., and Pšenčík, I. (1982). Computation of wave fields in inhomogeneous media – Gaussian beam approach. *Geophysical Journal International*, 70(1):109–128.

Červený, V. and Soares, J. E. P. (1992). Fresnel volume ray tracing. *Geophysics*, 57(7):902–915.

Chapman, C. (2004). *Fundamentals of seismic wave propagation.* Cambridge University Press, Cambridge UK.

Chapman, C. and Drummond, R. (1982). Body-wave seismograms in inhomogeneous media using Maslov asymptotic theory. *Bulletin of the Seismological Society of America*, 72(6B):S277–S317.

Dahlen, F. A., Hung, S.-H., and Nolet, G. (2000). Fréchet kernels for finite-frequency traveltimes - I. theory. *Geophysics Journal International*, 141(1):157–174.

Dutta, G. and Schuster, G. T. (2016). Wave-equation Q tomography. *Geophysics.*

Fabien-Ouellet, G. and Sarkar, R. (2020). Seismic velocity estimation: a deep recurrent neural-network approach. *Geophysics*, 85(1):U21–U29.

Farra, V. (2005). First-order ray tracing for qS waves in inhomogeneous weakly anisotropic media. *Geophysical Journal International*, 161(2):309–324.

Gajewski, D. and Pšenčík, I. (1992). Vector wavefields for weakly attenuating anisotropic media by the ray method. *Geophysics*, 57(1):27–38.

Gjøystdal, H., Iversen, E., Laurain, R., Lecomte, I., Vinje, V., and Åstebøl, K. (2002). Review of ray theory applications in modelling and imaging of seismic data. *Studia geophysica et geodaetica*, 46(2):113–164.

Guitton, A. and Verschuur, D. (2004). Adaptive subtraction of multiples using the L1-norm. *Geophysical Prospecting*, 52(1):27–38.

Hanyga, A. and Seredyńska, M. (2000). Ray tracing in elastic and viscoelastic media. *Pure and Applied Geophysics*, 157(5):679–717.

Hill, N. R. (2001). Prestack Gaussian-beam depth migration. *Geophysics*, 66(4):1240–1250.

Hudson, J. A. and Heritage, J. R. (1981). The use of the Born approximation in seismic scattering problems. *Geophysical Journal International*, 66(1):221–240.

Ikelle, L. T. and Amundsen, L. (2005). *Introduction to petroleum seismology.* Society of Exploration Geophysicist.

Iversen, E. (2004). The isochron ray in seismic modeling and imaging. *Geophysics*, 69(4):1053–1070.

Jin, S., Madariaga, R., Virieux, J., and Lambaré, G. (1992). Two-dimensional asymptotic iterative elastic inversion. *Geophysical Journal International*, 108(2):575–588.

Keers, H., Vasco, D. W., and Johnson, L. R. (2001). Viscoacoustic crosswell imaging using asymptotic waveforms. *Geophysics*, 66(3):861–870.

Kravtsov, Y. A., Forbes, G., and Asatryan, A. (1999). Theory and applications of complex rays. *E. Wolf, ed.: Progress in optics*, 39:1–62.

Lacoss, R. T., Kelly, E. J., and Toksöz, M. N. (1969). Estimation of seismic noise structure using arrays. *Geophysics*, 34(1):21–38.

Lailly, P. (1983). The seismic inverse problem as a sequence of before stack migrations. In *Conference on Inverse Scattering – Theory and Application*, volume 11, page 206. Siam.

Lambaré, G., Operto, S., Podvin, P., and Thierry, P. (2003). 3D ray+Born migration/inversion – part 1: Theory. *Geophysics*, 68(4):1348–1356.

Moczo, P., Kristek, J., and Gális, M. (2014). *The finite-difference modelling of earthquake motions: Waves and ruptures.* Cambridge University Press.

Moczo, P., Robertsson, J. O., and Eisner, L. (2007). The finite-difference time-domain method for modeling of seismic wave propagation. *Advances in Geophysics*, 48:421–516.

Moore, I. and Dragoset, B. (2008). General surface multiple prediction: a flexible 3D SRME algorithm. *First Break*, 26(9).

Moser, T. J. (2012). Review of ray-Born forward modeling for migration and diffraction analysis. *Studia Geophysica et Geodaetica*, 56(2):411–432.

Mulder, W. A. and Plessix, R.-E. (2004). A comparison between one-way and two-way wave-equation migration. *Geophysics*, 69(6):1491–1504.

Nocedal, J. and Wright, S. (2006). *Numerical optimization.* Springer Science & Business Media.

Operto, S., Lambaré, G., Podvin, P., and Thierry, P. (2003). 3D ray+Born migration/inversion – part 2: Application to the SEG/EAGE overthrust experiment. *Geophysics*, 68(4):1357–1370.

Pan, W., Innanen, K. A., and Liao, W. (2017). Accelerating Hessian-free Gauss-Newton full-waveform inversion via L-BFGS preconditioned conjugate-gradient algorithm. *Geophysics*, 82(2):R49–R64.

Peacock, K. and Treitel, S. (1969). Predictive deconvolution: Theory and practice. *Geophysics*, 34(2):155–169.

Pratt, R. G. (1999). Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model. *Geophysics*, 64(3):888–901.

Pratt, R. G., Shin, C., and Hick, G. (1998). Gauss–Newton and full Newton methods in frequency – space seismic waveform inversion. *Geophysical Journal International*, 133(2):341–362.

Pšenčík, I. and Farra, V. (2005). First-order ray tracing for qP waves in inhomogeneous, weakly anisotropic media. *Geophysics*, 70(6):D65–D75.

Ribodetti, A. and Virieux, J. (1998). Asymptotic theory for imaging the attenuation factor Q. *Geophysics*, 63(5):1767–1778.

Robertsson, J. O., Blanch, J. O., and Symes, W. W. (1994). Viscoelastic finite-difference modeling. *Geophysics*, 59(9):1444–1456.

Sarajaervi, M. and Keers, H. (2018a). Computation of ray-Born seismograms using isochrons. *Geophysics*, 83(5):1–58.

Sarajaervi, M. and Keers, H. (2018b). Efficient modelling and imaging in viscoelastic media using GPUs. In *80th EAGE Conference and Exhibition 2018*.

Sarajaervi, M. and Keers, H. (2019). Ray-based modeling and imaging in viscoelastic media using graphics processing units. *Geophysics*, 84(5):1–57.

Snieder, R. (2001). General theory of elastic wave scattering. In *Scattering and Inverse Scattering in Pure and Applied Science*, pages 528–541. Academic Press Inc.

Stockwell, J. W. (1997). Free software in education: A case study of CWP/SU: Seismic Unix. *The Leading Edge*, 16(7):1045–1050.

Symes, W. W. (2008). Migration velocity analysis and waveform inversion. *Geophysical prospecting*, 56(6):765–790.

Tarantola, A. (1984a). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.

Tarantola, A. (1984b). Linearized inversion of seismic reflection data. *Geophysical prospecting*, 32(6):998–1015.

Tarantola, A. (1986). A strategy for nonlinear elastic inversion of seismic reflection data. *Geophysics*, 51(10):1893–1903.

Tarantola, A. (1988). Theoretical background for the inversion of seismic waveforms including elasticity and attenuation. *Pure and Applied Geophysics*, 128(1-2):365–399.

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.

Thierry, P., Lambaré, G., Podvin, P., and Noble, M. S. (1999). 3-D preserved amplitude prestack depth migration on a workstation. *Geophysics*, 64(1):222–229.

Thomsen, L. (1986). Weak elastic anisotropy. *Geophysics*, 51(10):1954–1966.

Thorson, J. R. and Claerbout, J. F. (1985). Velocity-stack and slant-stack stochastic inversion. *Geophysics*, 50(12):2727–2741.

Vavryčuk, V. (2008). Real ray tracing in anisotropic viscoelastic media. *Geophysical Journal International*, 175(2):617–626.

Verschuur, D. J., Berkhout, A., and Wapenaar, C. (1992). Adaptive surface-related multiple elimination. *Geophysics*, 57(9):1166–1177.

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

Virieux, J., Asnaashari, A., Brossier, R., Métivier, L., Ribodetti, A., and Zhou, W. (2017). An introduction to full waveform inversion. In *Encyclopedia of Exploration Geophysics*, pages R1–1. Society of Exploration Geophysicists.

Virieux, J., Cruz-Atienza, V., Brossier, R., Chaljub, E., Coutant, O., Garambois, S., Mercerat, D., Prieux, V., Operto, S., Ribodetti, A., et al. (2016). Modelling seismic wave propagation for geophysical imaging. In *Seismic Waves - Research and Analysis*, chapter 13, pages 253–304. Masaki Kanao.

Virieux, J. and Lambaré, G. (2007). Theory and observations – body waves: Ray methods and finite frequency effects. *Treatise on Geophysics, vol 1, Seismology and structure of the Earth*, 1.

Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.

Williams, A. (2012). *C++ concurrency in action.* Manning Publications, London.

Woodward, M. J. (1992). Wave-equation tomography. *Geophysics*, 57(1):15–26.

Woodward, M. J., Nichols, D., Zdraveva, O., Whitfield, P., and Johns, T. (2008). A decade of tomography. *Geophysics*, 73(5):VE5–VE11.

Yilmaz, Ö. (2001). *Seismic data analysis: Processing, inversion, and interpretation of seismic data.* Society of exploration geophysicists.

Zhao, L. and Dahlen, F. (1996). Mode-sum to ray-sum transformation in a spherical and an aspherical earth. *Geophysical Journal International*, 126(2):389–412.

Paper I

A copy of paper I is included in this chapter.

GEOPHYSICS, VOL. 83, NO. 5 (SEPTEMBER-OCTOBER 2018); P. T245–T256, 11 FIGS., 1 TABLE.
10.1190/GEO2017-0669.1

Computation of ray-Born seismograms using isochrons

Martin Sarajaervi[1] and Henk Keers[2]

ABSTRACT

Seismic modeling in heterogeneous media is accomplished by using either approximate or fully numerical methods. A popular approximate method is ray-Born modeling, which requires the computation of 3D integrals. We have developed an integration technique for accurate and, under certain circumstances, efficient evaluation of the ray-Born integrals in the time domain. The 3D integrals are split into several 2D integrals, each of which gives the wavefield at a certain time, so that the waveform at each time step is computed independently of all other times. We compute seismograms for 3D heterogeneous acoustic media using this technique and compare these seismograms with seismograms computed using two other modeling methods: frequency-domain ray-Born modeling and finite-difference modeling of the acoustic wave equation. Our method can also be applied to elastic ray-Born modeling. Velocity models with smooth scatterers and the SEG/EAGE overthrust model are used for comparison. The ray-Born seismograms computed using the time- and frequency-domain ray-Born modeling methods are identical, as expected. The comparison between the ray-Born modeling and the finite-difference-modeling method indicates that the waveforms are similar for both types of velocity models. We evaluate the discrepancies in terms of multiple scattering and multipathing.

INTRODUCTION

Modeling of seismic waves is very important for studying the Earth's structure. For example, seismic algorithms such as full-waveform inversion require it. In practice, this modeling is often done using the acoustic wave equation, solved either by using fully numerical methods or by using approximate methods (e.g., Carcione et al., 2002; Virieux and Lambaré [2015], chapter 1.05). Fully numerical methods are popular and very useful, but they are not computationally efficient at higher frequencies, especially in large 3D models. They also have limitations in the case of complicated structural models (e.g., finite-difference seismograms contain artifacts in the case of strongly undulating interfaces). Approximate methods do not necessarily suffer from these disadvantages; for example, ray theory is particularly valid at high frequencies in smooth models. Therefore, it is sometimes beneficial to compute wavefields using approximate methods.

A useful technique for modeling singly scattered waves is the (first-order) ray-Born approximation. Several papers have investigated the theoretical aspects of the Born (e.g., Knopoff and Hudson, 1964; Hudson, 1977; Hudson and Heritage, 1981) and ray-Born approximations (see references listed by Chapman [2004], section 10.3 and Červený [2005], section 2.6.2). In studies focusing on the numerical aspects of ray-Born modeling, Červený and Coppoli (1992) generate synthetic seismograms in the case of a background medium with two curved interfaces and a perturbation consisting of isolated scatterers. For a model with similar characteristics, Gibson et al. (1993) use ray-Born modeling to examine reflections from vertical seismic profile data. Coates and Charrette (1993) compare the 2D isotropic elastic generalized Born approximation (Coates and Chapman, 1991) and the conventional Born approximation to finite-difference modeling. In a study focusing on 3D ray-Born inversion, Thierry et al. (1999) compute synthetic traces from a migrated image and compare the results with real data. In a review paper, Moser (2012) pays special attention to the advantages of acoustic ray-Born modeling for modeling diffractions from specific structural discontinuities, such as edges and tips. Šachl (2013) investigates discretization effects of the elastic ray-Born scattering integral and suggests a cosine window to attenuate boundary diffractions. Tengesdal (2013) and Minakov et al. (2017) discuss 2D acoustic ray-Born and finite-difference modeling and show synthetic- and real-data examples as well as an application to full-waveform inversion. All of the above papers compute the ray-Born

Manuscript received by the Editor 11 October 2017; revised manuscript received 7 February 2018; published ahead of production 10 May 2018; published online 18 July 2018.
[1]Schlumberger, Risabergveien 3, 4068 Stavanger, Norway. E-mail: msarajaervi@slb.com.
[2]University of Bergen, Realfagbygget, Allégt., 5020 Bergen, Norway. E-mail: henk.keers@uib.no.

integrals, in either the time or frequency domain, by summing over individual volume or area elements (e.g., Coates and Charrette, 1993). An alternative method for the computation of ray-Born seismograms in the time domain was presented by Spencer et al. (1997) and Keers et al. (2002). That method used a specific integration over tetrahedrons.

We present a new method for 3D ray-Born modeling in the time domain using isochron surfaces. Integrals over isochrons are used to accurately compute seismograms for a band-limited signal. We compare the results with seismograms computed using frequency-domain ray-Born and finite-difference modeling. The method can be seen as an extension to 3D integrals of what was done for 1D Wentzel-Kramers-Brillouin-Jeffreys (WKBJ) seismograms by Chapman (1978), where the band limiting is done before the discretization and for 2D Kirchhoff seismograms by Haddon and Buchen (1981), where the integrals are computed over isochrons.

The results in this paper are based on the acoustic wave equation, but the concepts can be extended to the elastic case. We give a brief description of this in the "Discussion" section.

In the "Theory" section, we review the acoustic ray-Born modeling and give the theoretical background of the integration technique. Next, we discuss the numerical implementation of the ray-Born integrals. For the sake of completeness, this section also gives a brief description of the two-point ray-tracing algorithm as well as the frequency-domain ray-Born integrals and finite-difference modeling. In the "Results" section, we present seismograms computed using two types of velocity models with different scattering characteristics. The first type has smooth velocity perturbations, analogous to that used in ray-based tomographic model building (or full-waveform inversion with low-frequency velocity updates). The second type, the 3D SEG/EAGE overthrust model of Aminzadeh et al. (1997), contains rough and rapidly varying perturbations.

THEORY

For a heterogeneous velocity model $c(\mathbf{x})$, the propagation of acoustic waves can be described by a Green's function $G(\mathbf{x}, \mathbf{s}, t)$, where $\mathbf{x}$ denotes the spatial position of observation, $\mathbf{s}$ denotes the source position, and t denotes time. The term G satisfies the wave equation (e.g., Morse and Feshbach, 1953; Ikelle and Amundsen, 2005)

$$c^{-2}(\mathbf{x})G_{tt}(\mathbf{x}, \mathbf{s}, t) - \Delta G(\mathbf{x}, \mathbf{s}, t) = \delta(\mathbf{x} - \mathbf{s})\delta(t), \tag{1}$$

where the source is a Dirac delta function in time and space, which acts at time $t = 0$, $G_{tt} = \partial^2 G/\partial t^2$ and Δ is the Laplace operator. Expressions for the solution of equation 1 are often more convenient in the frequency domain than those in the time domain. For this reason, we also give the frequency-domain formulation, with ω denoting the angular frequency, and we adopt the sign convention for the Fourier-transform pair from Chapman (2004, pp. 58 and 59). The wave equation in the frequency domain becomes the Helmholtz equation:

$$\omega^2 c^{-2}(\mathbf{x})g(\mathbf{x}, \mathbf{s}, \omega) + \Delta g(\mathbf{x}, \mathbf{s}, \omega) = \delta(\mathbf{x} - \mathbf{s}), \tag{2}$$

where $g(\mathbf{x}, \mathbf{s}, \omega)$ denotes the Green's function in the frequency domain.

In forward modeling, equation 2, for a given c, needs to be solved for g. If perturbation theory is used to solve equation 2, then one decomposes c into a background velocity c_0 and perturbation term c_1:

$$c = c_0 + c_1. \tag{3}$$

Often, the velocity field c_0 is smooth, or it is only a function of depth, and c_1 is more rapidly varying. The solution to equation 2 for $c = c_0$ is denoted by g_0. The modeling problem then is to find g when g_0 is known. Using perturbation theory, an expression for g in terms of the background Green's function g_0 (e.g., Červený, 2005; Ikelle and Amundsen, 2005) can be found:

$$g(\mathbf{x}, \mathbf{s}, \omega) = g_0(\mathbf{x}, \mathbf{s}, \omega) + \omega^2 \int_D g_0(\mathbf{x}, \mathbf{x}', \omega)V(\mathbf{x}')g(\mathbf{x}', \mathbf{s}, \omega)d\mathbf{x}'. \tag{4}$$

Here, the "scattering term" V is defined as

$$V = c_0^{-2} - c^{-2}, \tag{5}$$

and D is the scattering region, i.e., the region where V is nonzero. In compact notation, if we define the operator L as

$$(Lg)(\mathbf{x}) = \omega^2 \int_D g_0(\mathbf{x}, \mathbf{x}', \omega)V(\mathbf{x}')g(\mathbf{x}', \omega)d\mathbf{x}', \tag{6}$$

then equation 4 can be rewritten as

$$g = g_0 + Lg \tag{7}$$

or

$$g = (I - L)^{-1}g_0, \tag{8}$$

where I is the identity. The operator $(I - L)^{-1}$ can be expanded, assuming that it is invertible, as a power series in terms of L (e.g., Griffel, 2002):

$$g = (I + L + L^2 + L^3 + \cdots)g_0. \tag{9}$$

Here, each term describes scattering of a certain order: L describes the first-order scattering, L^2 describes the second-order scattering, and so on. The series is convergent if $\|L\| < 1$ for some suitable norm. This is often the case, especially if V is sufficiently small and D is not too large.

In practice, the explicit expression for g as given by the scattering series 9 is approximated using only the first two terms. This gives the Born approximation to the wavefield:

$$g \approx g_0 + Lg_0 = g_0 + g_1. \tag{10}$$

In the above expression, the singly scattered waves g_1 are given as a volume integral over wavefields propagating through the background medium:

$$g_1(\mathbf{x}, \mathbf{s}, \omega) = \omega^2 \int_D g_0(\mathbf{x}, \mathbf{x}', \omega)V(\mathbf{x}')g_0(\mathbf{x}', \mathbf{s}, \omega)d\mathbf{x}'. \tag{11}$$

As it is assumed that $\|c_1\| \ll \|c_0\|$, we can rewrite V:

$$V = \frac{1}{c_0^2} - \frac{1}{(c_0 + c_1)^2} \approx 2c_1 c_0^{-3}. \tag{12}$$

If we furthermore use $\mathbf{r}$ to indicate receiver position, then the Born-scattering integral 11 becomes

$$g_1(\mathbf{r},\mathbf{s},\omega) \approx \omega^2 \int_D g_0(\mathbf{r},\mathbf{x},\omega) \frac{2c_1(\mathbf{x})}{c_0^3(\mathbf{x})} g_0(\mathbf{x},\mathbf{s},\omega) d\mathbf{x}. \quad (13)$$

A clear physical interpretation of equation 13 can be given (e.g., Snieder, 2001): The scattered wave $g_0(\mathbf{x},\mathbf{s},\omega)$ propagates in the background medium from the source located at $\mathbf{s}$ to a scatterer at point $\mathbf{x}$, interacts with the scatterer, and then propagates as $g_0(\mathbf{r},\mathbf{x},\omega)$ from the scatterer to the receiver at $\mathbf{r}$. The total scattered wavefield g_1 is then found by summation of the integrand over all scattering points $\mathbf{x}$ and multiplication by ω^2.

Background Green's function

The Born integral 13 requires knowledge of g_0. In the case of a homogeneous background medium, g_0 is known explicitly. However, this situation is geologically not realistic. If we assume a slowly varying c_0, then it is natural to use ray theory (Červený, 2005) to compute g_0. In such a case, we look for solutions of the form

$$g_0(\mathbf{x},\mathbf{s}) = A(\mathbf{x},\mathbf{s}) e^{i\omega T(\mathbf{x},\mathbf{s})}, \quad (14)$$

and we solve for traveltime T and amplitude A. Equations for T and A are found by inserting 14 into the wave equation and applying the high-frequency approximation ($\omega \to \infty$). This gives the eikonal equation for T

$$|\nabla T|^2 = c_0^{-2}, \quad (15)$$

and the transport equation for A

$$A\Delta T + 2\nabla A \cdot \nabla T = 0. \quad (16)$$

Characteristics of the eikonal equation 15 are raypaths, described by the kinematic ray equations (Červený, 2005), and they can be used to find T. A solution to the transport equation 16 can be found by solving the dynamic ray-tracing equations (Červený, 2005). Note that it is assumed that there is no multipathing.

The ray-Born approximation

If the background Green's function g_0 is obtained from ray theory, then the Born approximation is called the ray-Born approximation. In this case, the ray-Born integral (e.g., Beylkin, 1985) becomes

$$g_1(\mathbf{r},\mathbf{s},\omega) = \omega^2 \int_D A_{rs}(\mathbf{x}) e^{i\omega T_{rs}(\mathbf{x})} V(\mathbf{x}) d\mathbf{x}, \quad (17)$$

which in the time domain is

$$G_1(\mathbf{r},\mathbf{s},t) = \frac{d^2}{dt^2} \int_D A_{rs}(\mathbf{x}) \delta(t - T_{rs}(\mathbf{x})) V(\mathbf{x}) d\mathbf{x}. \quad (18)$$

Here, the traveltime function T_{rs} and amplitude function A_{rs} are defined as

$$T_{rs}(\mathbf{x}) = T(\mathbf{r},\mathbf{x}) + T(\mathbf{x},\mathbf{s}) \quad (19)$$

and

$$A_{rs}(\mathbf{x}) = A(\mathbf{r},\mathbf{x}) A(\mathbf{x},\mathbf{s}). \quad (20)$$

The integral in equation 18 can be computed using the composition of a function and the Dirac delta. This gives

$$G_1(\mathbf{r},\mathbf{s},t) = \frac{d^2}{dt^2} \int_{S_t} \frac{A_{rs}(\mathbf{x})}{|\nabla T_{rs}(\mathbf{x})|} V(\mathbf{x}) d\mathbf{x}, \quad (21)$$

with $S_t = \{\mathbf{x} | t = T_{rs}(\mathbf{x})\}$. For a slowly varying c_0, S_t is a closed 2D surface, which envelopes the source $\mathbf{s}$ and receiver $\mathbf{r}$. We call this surface an isochron. From equation 21, we see that at time t the ray-Born seismogram is determined only by the isochron S_t and the values of A_{rs}, ∇T_{rs}, and V on S_t.

Later on in this paper, we briefly discuss multiple scattering and multipathing, and it is important to distinguish the two effects. Multipathing occurs when, given a velocity model c and source $\mathbf{s}$, there are two or more rays between $\mathbf{s}$ and the receiver $\mathbf{r}$. Multiply scattered waves form that part of the wavefield that is modeled using the higher order terms ($L^2 g_0$, $L^3 g_0$, etc.) of the scattering series 9. Therefore, multipathing depends on the source $\mathbf{s}$, the velocity model c, but also the background velocity c_0. Singly scattered waves are the waves modeled by the term $g_1 = Lg_0$. Note that multipathing and multiple scattering are not mutually exclusive, so that a part of the wavefield can be interpreted in terms of multipathing and in terms of multiple scattering.

Discrete ray-Born seismograms in the time domain

We start with the expression for the ray-Born approximation

$$G_1(\mathbf{r},\mathbf{s},t) = \frac{d^2}{dt^2} \int_D a_{rs}(\mathbf{x}) \delta(t - T_{rs}(\mathbf{x})) d\mathbf{x}, \quad (22)$$

with

$$a_{rs}(\mathbf{x}) = A_{rs}(\mathbf{x}) 2c_1(\mathbf{x}) c_0^{-3}(\mathbf{x}), \quad (23)$$

and T_{rs} as given in equation 19. In practice, seismograms are only recorded and computed at discrete equally spaced time points. It is therefore natural to compute seismograms at these times only. The discretization can be done in various ways. One way is to convolve G_1 with the Boxcar function $B(t)$ given by the Heaviside step function $H(t)$ as

$$B(t) = H(t + \Delta t/2) - H(t - \Delta t/2). \quad (24)$$

This gives the band-limited integral, where the support is bounded

$$\hat{G}_1(\mathbf{r},\mathbf{s},t) = \frac{d^2}{dt^2} \int_{V_t(\mathbf{x})} a_{rs}(\mathbf{x}) d\mathbf{x}, \quad (25)$$

with

$$V_t(\mathbf{x}) = \{\mathbf{x} | t - \Delta t/2 < T_{rs}(\mathbf{x}) < t + \Delta t/2\}. \quad (26)$$

We leave the differential d^2/dt^2 to be evaluated numerically later. The integral over $V_t(\mathbf{x})$ is the volume between two isochron surfaces $S_{t-\Delta t/2}$ and $S_{t+\Delta t/2}$ with $S_{t\pm\Delta t/2} = \{\mathbf{x} | t \pm \Delta t/2 = T_{rs}(\mathbf{x})\}$ (see

Figure 1). Note also that $V_t = \cup S_\tau$ with $\tau \in [t - \Delta t/2, t + \Delta t/2]$, i.e., V_t is the union for all isochron surfaces and $S_{\tau_1} \cap S_{\tau_2} = \emptyset$ for $\tau_1 \neq \tau_2$. Moreover, because there is no multipathing, all surfaces S_τ are well-defined and smooth. For a homogeneous 3D background medium, the isochrons are ellipsoidal (see Figure 2). The ellipsoids become spheres in the special case when $\mathbf{s} = \mathbf{r}$. If the background medium is slowly varying, then the surfaces S_τ are deformed, but they are still smooth and ellipsoidal with $S_{\tau+\Delta t}$ completely enveloping S_τ.

A natural parameterization of V_t is given in terms of coordinates (ξ_1, ξ_2), which parameterize the surface S_τ and coordinate y. For fixed (ξ_1, ξ_2), the isochron raypath $\mathbf{x}(\xi_1, \xi_2, y)$ is approximated by a line perpendicular to S_τ. Because of this, the map $(x_1, x_2, x_3) \mapsto (\xi_1, \xi_2, y)$ is well-defined and smooth, and so is its Jacobian

$$J = \left| \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, y)} \right|. \tag{27}$$

Therefore, we have

$$\hat{G}_1(\mathbf{r}, \mathbf{s}, t) = \frac{d^2}{dt^2} \int_{V_t(\mathbf{x})} a_{rs}(\mathbf{x}(\xi_1, \xi_2, y)) \times \left| \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, y)} \right| d\xi_1 d\xi_2 dy. \tag{28}$$

Because

$$\left| \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, y)} \right| = \left| \frac{\partial \mathbf{x}}{\partial y} \cdot \left(\frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2} \right) \right| \tag{29}$$

and $(\partial \mathbf{x}/\partial y) \perp (\partial \mathbf{x}/\partial \xi_i)$, with $i = 1$ or 2, this can be rewritten as

$$\hat{G}_1(\mathbf{r}, \mathbf{s}, t) = \frac{d^2}{dt^2} \int_{V_t(\mathbf{x})} a_{rs}(\mathbf{x}(\xi_1, \xi_2, y)) \times \left| \frac{\partial \mathbf{x}}{\partial y} \right| \left| \frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2} \right| d\xi_1 d\xi_2 dy. \tag{30}$$

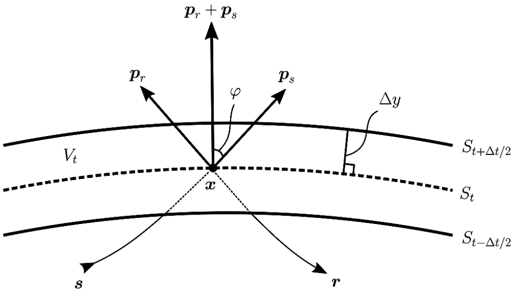


Figure 1. Illustration of a strip from an isochron at $S_{t-\Delta t/2}$ to an isochron at $S_{t+\Delta t/2}$ and the line segment Δy that is used to compute the time-domain surface integral 36. The figure is modified from Chapman (2004, figure 10.23).

We now assume that, for fixed (ξ_1, ξ_2), a_{rs} is constant and also $|(\partial \mathbf{x}/\partial \xi_1) \times (\partial \mathbf{x}/\partial \xi_2)|$. We choose these constants, such that $a_{rs}(\mathbf{x}(\xi_1, \xi_2, y)) = a_{rs}(\mathbf{x}(\xi_1, \xi_2, \tilde{y}))$, where $\tilde{y}$ is the value for S_t, and similarly for $|(\partial \mathbf{x}/\partial \xi_1) \times (\partial \mathbf{x}/\partial \xi_2)|$. We denote these constants for y by $a_{rs}(\mathbf{x}(\xi_1, \xi_2))$ and $|(\partial \mathbf{x}/\partial \xi_1) \times (\partial \mathbf{x}/\partial \xi_2)|$. It should be stressed that a_{rs} varies over each isochron (as a function of (ξ_1, ξ_2)). In the integration over y in equation 30, it is only assumed that a_{rs} is constant as a function of y when going from one isochron to the next isochron. We now have

$$\hat{G}_1(\mathbf{r}, \mathbf{s}, t) = \frac{d^2}{dt^2} \int_{S_t} a_{rs}(\mathbf{x}(\xi_1, \xi_2)) \left| \frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2} \right| \times \left[\int \left| \frac{\partial \mathbf{x}}{\partial y} \right| dy \right] d\xi_1 d\xi_2. \tag{31}$$

The integral $\int |\partial \mathbf{x}/\partial y| dy$ is found using the length of the line segment $2\Delta y$ (see Figure 1) and from the definition of S_t and $S_{t+\Delta t/2}$. This gives

$$\Delta y = \frac{\Delta t}{2|\mathbf{p}_r + \mathbf{p}_s|}. \tag{32}$$

Here, the slowness vectors from the source and receiver at $\mathbf{x}$ are denoted by $\mathbf{p}_s = \nabla T(\mathbf{s}, \mathbf{x})$ and $\mathbf{p}_r = \nabla T(\mathbf{r}, \mathbf{x})$, respectively. Furthermore, using

$$|\mathbf{p}_r + \mathbf{p}_s| = \frac{2 \cos \varphi}{c_0}, \tag{33}$$

with φ being the half the angle between $\mathbf{p}_r$ and $\mathbf{p}_s$ (e.g., Miller et al., 1987), $2\Delta y$ is, to first order

$$2\Delta y = \frac{c_0 \Delta t}{2 \cos \varphi}. \tag{34}$$

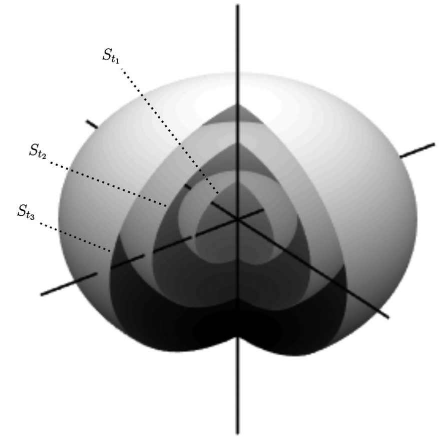


Figure 2. Sketch of isochrons for a medium with a homogeneous background.

Inserting this in equation 31 and using the definition of a_{rs} gives

$$\hat{G}_1(\mathbf{r},\mathbf{s},t)=\frac{d^2}{dt^2}\int_{S_t} A_{rs}(\mathbf{x}(\xi_1,\xi_2))\frac{\Delta t}{|\cos\varphi(\mathbf{x}(\xi_1,\xi_2))|} \times\frac{c_1(\mathbf{x}(\xi_1,\xi_2))}{c_0^2(\mathbf{x}(\xi_1,\xi_2))}\left|\frac{\partial\mathbf{x}}{\partial\xi_1}\times\frac{\partial\mathbf{x}}{\partial\xi_2}\right|d\xi_1 d\xi_2 \tag{35}$$

or

$$\hat{G}_1(\mathbf{r},\mathbf{s},t)=\frac{d^2}{dt^2}\int_{S_t} A_{rs}(\mathbf{x})\frac{\Delta t}{|\cos\,\varphi(\mathbf{x})|}\frac{c_1(\mathbf{x})}{c_0^2(\mathbf{x})}d\mathbf{x}. \tag{36}$$

It should be emphasized that equation 36 is band-limited and therefore scales when V_t changes as a function of Δt. Note that when $\varphi\approx 0$, for instance, near zero-offset, the integral can be approximated by

$$\hat{G}_1(\mathbf{r},\mathbf{s},t)=\frac{d^2}{dt^2}\int_{S_t} A_{rs}(\mathbf{x})\Delta t\frac{c_1(\mathbf{x})}{c_0^2(\mathbf{x})}d\mathbf{x}. \tag{37}$$

Similarly, for forward scattering, we have $\varphi\approx\pi$, so that equation 37 also holds in this case.

For $t=t_1$, it is not possible to integrate along an isochron because the isochron for $t=t_1-\Delta t$ does not exist. In this case, there are two different cases that must be distinguished: (1) $t_1-\Delta t<T_{rs}<t_1-\Delta t/2$ and (2) $t_1-\Delta t/2<T_{rs}<t_1$. In both cases, the ray-Born integral is given in terms of a volume integral as in equation 25. In our experience, these contributions in practice are not significant. However, this should be included here for the sake of completeness. The numerical implementation of equation 36 and the volume integrals are discussed in the next section.

NUMERICAL IMPLEMENTATION

Ray tracing

In preparation for ray-Born modeling using equation 36, the quantities T_{rs}, ∇T_{rs}, and A_{rs} are computed using ray tracing. They are found by first computing $T(\mathbf{s},\mathbf{x})$, $\nabla T(\mathbf{s},\mathbf{x})$, and $A(\mathbf{s},\mathbf{x})$ from the source $\mathbf{s}$ to all points $\mathbf{x}$ in the scattering region D. Likewise, $T(\mathbf{r},\mathbf{x})$, $\nabla T(\mathbf{r},\mathbf{x})$, and $A(\mathbf{r},\mathbf{x})$ are computed from the receiver $\mathbf{r}$ to the scattering points $\mathbf{x}$.

If the velocity model is slowly varying, raypaths, traveltimes T, and wavefront normals ∇T are found by solving the kinematic ray equations given the initial conditions (position and take-off angles). In a similar manner, the amplitude is computed by solving the dynamic ray equations. Both sets of equations are solved numerically with a fourth-order Runge-Kutta method (Press et al., 2007) with Snell's law applied when the ray encounters an interface. The result is two 3D fans of rays emanating from the source and the receiver.

For a given source $\mathbf{s}$ and scattering point $\mathbf{x}$, the two-point ray-tracing problem is to find a ray from $\mathbf{s}$ to $\mathbf{x}$. This is done in two steps. First, all points on the ray fan are used to construct a 3D Delaunay tessellation (de Berg et al., 2000), which covers the scattering region D. For each $\mathbf{x}$ in D, we then interpolate values for T, ∇T, and A using the tessellation and linear interpolation on a tetrahedron (e.g., Farin, 2002). Likewise, the same procedure is done for a receiver $\mathbf{r}$ and scattering point $\mathbf{x}$.

With the two-point ray tracing completed for the source-receiver pair, we compute the traveltime function T_{rs} using equation 19 and the amplitude function A_{rs} using equation 20.

Ray-Born seismograms: Time domain

The isochron can be computed in various ways. In a heterogeneous background with the traveltime function T_{rs} given on a discrete grid, one of several isosurface extraction techniques can be used (e.g., Sutton et al., 2000). We have computed T_{rs} using the ray-tracing procedure described above, and we have used the marching cubes algorithm (Lorensen and Cline, 1987) to extract surfaces. In this case, S_τ is defined on a triangular grid with vertex positions $\mathbf{x}_1=\mathbf{x}_1(n)$, $\mathbf{x}_2=\mathbf{x}_2(n)$, and $\mathbf{x}_3=\mathbf{x}_3(n)$, where $n=1,\ldots,n_{tr}$ and n_{tr} is the number of triangles.

The numerical integration on the triangular grid is carried out by computing an average value for the integrand at each triangle times the area of each triangle. We evaluate the integrand 36 at vertex positions $\mathbf{x}_m(n)$ with $m=1,2,3$ and compute an average value I_n as

$$I_n=\frac{\Delta t}{3}\sum_{m=1}^{3}\frac{A_{rs}(\mathbf{x}_m)}{|\cos\,\varphi(\mathbf{x}_m)|}\frac{c_1(\mathbf{x}_m)}{c_0^2(\mathbf{x}_m)}. \tag{38}$$

Furthermore, we denote the area element of a triangle ΔA_n and calculate this as half the area of the parallelogram spanned by $\mathbf{x}_2-\mathbf{x}_1$ and $\mathbf{x}_3-\mathbf{x}_1$:

$$\Delta A_n=\frac{1}{2}|(\mathbf{x}_2(\mathrm{n})-\mathbf{x}_1(\mathrm{n}))\times(\mathbf{x}_3(\mathrm{n})-\mathbf{x}_1(\mathrm{n}))|. \tag{39}$$

Having at our disposal the area ΔA_n and integrand I_n on the triangular grid, we compute the integral for $T_{rs}=t_i$ as

$$\hat{G}_1(\mathbf{r},\mathbf{s},t_i)\approx\frac{d^2}{dt^2}\sum_{n=1}^{n_{tr}}I_n\Delta A_n. \tag{40}$$

The second derivative is computed once for each source-receiver pair using numerical differentiation. In the following sections, we show seismograms computed using equation 40.

The volume integral 25 for the two special cases mentioned in the previous section can also be evaluated numerically. For this, the vertices of the first isochron as well as the points on the background ray, which is completely enclosed by the isochron, can be used. In this case, the integral 25 becomes a sum of the volume contributions of the tetrahedrons $\sum_{n=1}^{n_{\text{tet}}}\hat{I}_n\Delta V_n$, where n_{tet} is the total number of tetrahedrons, ΔV_n is the volume of tetrahedron n, and $\hat{I}_n$ is the integrand of tetrahedron n (which may be computed as the average of the integrand on the vertices of each tetrahedron).

Ray-Born seismograms: Frequency-domain comparison

The computation of the ray-Born approximation in the frequency domain is done using equation 17. The discretization of the integral is accomplished using a regular grid over the scattering region D. By approximating the integrand in each grid element by a constant, the computation of $g_1(\mathbf{r},\mathbf{s},\omega)$ becomes a sum over these integrand values, multiplied by the volume element.

The comparison with the time-domain ray-Born modeling method and finite-difference modeling is done in the time domain. Therefore, after computation of $g_1(\mathbf{r}, \mathbf{s}, \omega)$ for all frequencies, an inverse fast Fourier transform is used.

Finite-difference comparison

The finite-difference method (e.g., Etgen and O'Brien, 2007) is used to solve the wave equation 1. For our implementation, we do not include any absorbing boundary conditions. To avoid edge effects within the time window of interest, all velocity models are extended beyond the area of interest.

Table 1. Velocity information for one realization of a random Gaussian velocity model, but with different perturbation strengths.

Model name	Velocity range	Strength (ε)
Weak perturbation	$1990 - 2010$ m/s	0.5% (0.001)
Medium-strength perturbation	$1800 - 2205$ m/s	10.18% (0.02)
Strong perturbation	$1706 - 2302$ m/s	15.18% (0.03)

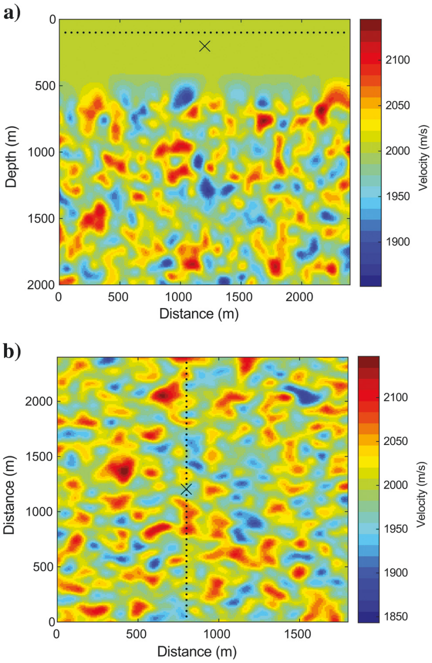


Figure 3. Random Gaussian media with medium-strength perturbation (10%) for: (a) y-direction and (b) a depth slice at 1000 m used to compute synthetics. The black dots indicate the receiver positions, and the crosses indicate the shot position.

RESULTS

In this section, seismograms computed using the three modeling methods (time-domain ray-Born using isochrons, frequency-domain ray-Born, and finite-difference) discussed in the previous section, are presented for two different velocity models. The first of these velocity models is a smooth model with random reflectivity. The second one is the 3D EAGE/SEG overthrust model (Aminzadeh et al., 1997). This is a complicated velocity model with a rapidly varying velocity. Small differences between the computed seismograms are assessed using single-trace displays. Larger scale differences, with significance for structural interpretation, are assessed using shot records and zero-offset sections.

3D random Gaussian model

First, we present the modeling results for the velocity models with random variations. These random velocities are generated with a Gaussian autocorrelation function characterized by a perturbation strength ε (root-mean-square slowness variation) and correlation distance a (Baig and Dahlen, 2004). We use this to create a random realization with weak, medium-strength, and strong perturbations (details are given in Table 1). The correlation distance in all cases is $a = 50$ m and the velocity model covers a volume of $1.8 \times 2.4 \times 2$ km. A constant background velocity of 2000 m/s is used. Figure 3 shows two cross sections of the particular realization that was used.

The model grid spacing is 10 m, and this is used for the ray-Born-modeling methods as well as the finite-difference-modeling method. This way, amplitudes, traveltimes, and velocities are all sampled on the same grid. The models are extended in all directions to avoid interference from artificial boundary reflections within the recording length of 2 s. The sampling interval used was 4 ms. Receivers are positioned at a 50 m interval from 0 to 2350 m in the y-direction and at depth of 100 m (Figure 3). Synthetics are generated for a shot located at $\mathbf{s} = (800, 1200, 200)$ m, using a source time function in the form of a causal Ricker wavelet, with a center frequency of 15 Hz.

For three discrete sample points in time, $t_i = (0.65, 0.85, 1.05)$ s and a source-receiver distance of 800 m, Figure 4 shows the corresponding isochron surfaces $T_{rs} = t_i$. The values on the surfa-

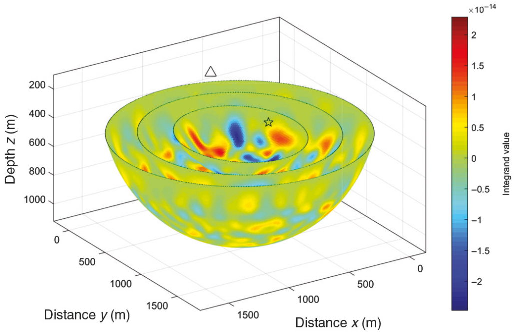


Figure 4. Isochron surfaces in a homogeneous background with integral values for the random Gaussian model at times $t_i = (0.65, 0.85, 1.05)$ s. The source and receiver positions are indicated by a star and a triangle, respectively.

ces are the values of the integrand of equation 36 for the weak perturbation model. As the background medium is homogeneous, the isochron surfaces are part of a spheroid.

Figure 5a (0.5% perturbation) shows a trace offset by 800 m relative to the source position (see the dashed blue line in Figure 6 for the location of the receiver) for the weak perturbation model (Table 1). We show the finite-difference seismogram and ray-Born seismograms computed in the time and frequency domains. Next, we show traces for the same random realization, but with medium-strength and strong perturbation models in Figure 5b (10% perturbation) and 5c (15% perturbation), respectively. For the models with 0.5%, 10%, and 15% perturbation, the ray-Born seismograms all appear with the same phase, but with a relative difference in the amplitudes. The results computed in the time and frequency domain are basically identical, as expected. In the comparison with finite-difference modeling, the seismograms appear with changes in the waveforms for the medium-strength (Figure 5b) and more so for the strong perturbation model (Figure 5c). This is because finite-difference modeling accounts for complicated wave-propagation effects such as multiple scattering and multipathing that become more predominant for stronger perturbations.

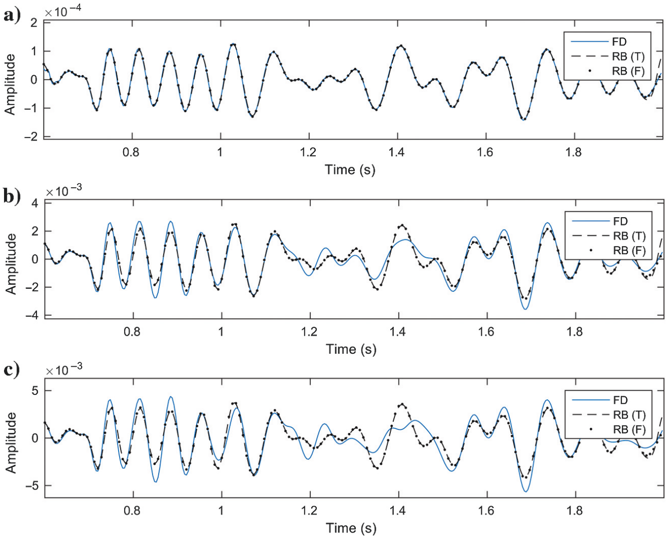


Figure 5. Comparison of traces computed using finite-difference and ray-Born modeling in 3D random Gaussian media for: (a) the weak perturbation (0.5%), (b) the medium-strong perturbation (10%), and (c) the strong perturbation (15%). Finite difference (solid blue) is compared with time-domain ray-Born (dashed black) and frequency-domain ray-Born (black dots).

Figure 6 shows synthetic shot gathers for the medium-strength perturbation model. The synthetics generated by the ray-Born method and finite differences are shown in Figure 6a and 6b, respectively. Across the offset positions, we can recognize events from the singly scattered wavefield in both seismograms.

3D SEG/EAGE overthrust model

In this subsection, time-domain ray-Born, frequency-domain ray-Born, and finite-difference synthetics are computed using the acoustic 3D SEG/EAGE overthrust model (Aminzadeh et al., 1997; Operto et al., 2003). The overthrust model represents a succession of sedimentary deposits in a compressional tectonic setting, resulting in strong vertical and lateral velocity variations. It was created to study 3D wave propagation and to test seismic processing algorithms.

Two minor modifications are made to the original model: Edges are extrapolated to avoid artificial boundary reflections, and a seafloor interface is added at a depth of 500 m using a water layer velocity of 1480 m/s. A subvolume of the larger model is displayed as a vertical cross section in Figure 7a and at a depth of 2 km in Figure 7b.

We create the background model by 3D boxcar averaging of the model without the water layer. The background model is shown in Figure 7c and 7d. The difference $c - c_0$ then gives the perturbation model, which, in Figure 7e and 7f, is displayed in terms of percent relative to the background. Perturbation strengths have a maximum of 41.1% and a minimum of −21.7%. On average, the positive velocity perturbation is 6.8% and the average value of the negative velocity perturbation is −5.1%. In contrast to the velocity models of

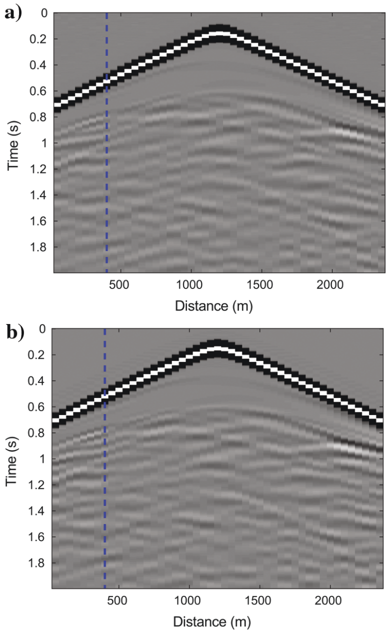


Figure 6. Modeling results in 3D random Gaussian media with 10% maximum perturbation for (a) ray-Born and (b) finite-difference method. Individual traces, for different perturbation strengths, are selected for the location indicated by the dashed blue line.

the previous section, the perturbation model contains sharp velocity variations, and, therefore, it is quite rough.

The grid spacing is 12.5 m along all spatial axes, and it is the same for frequency-domain ray-Born-modeling and finite differences. To achieve an exact match between time-domain ray-Born modeling and frequency-domain ray-Born modeling, we reduced the vertical grid spacing to a 6.25 m space for the time-domain implementation. This allows for a more precise isochron surface extraction. The source-

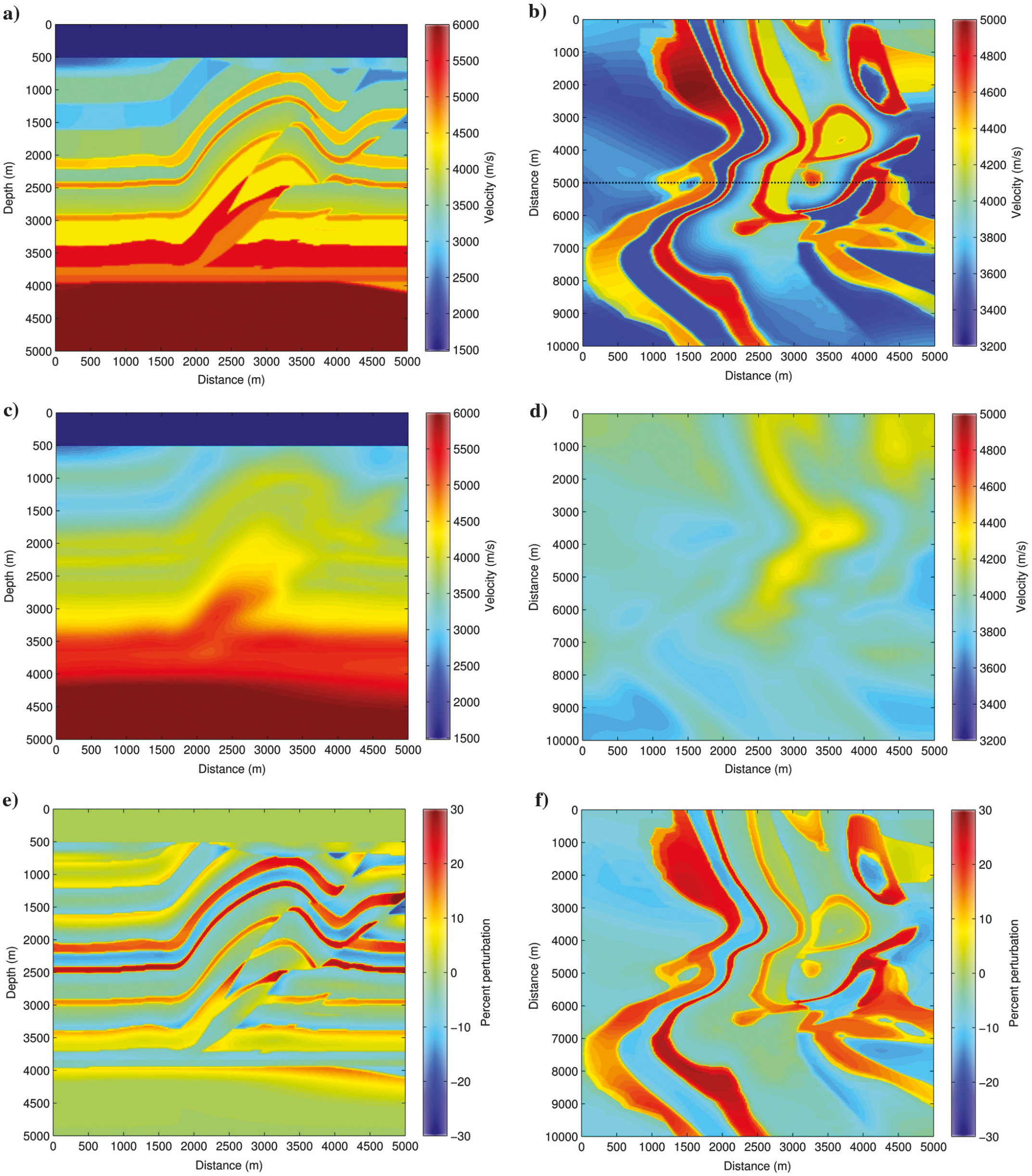


Figure 7. Cross sections and depth slices for the decomposed 3D overthrust model for (a and b) the complete model, (c and d) the smooth model, and (e and f) the perturbation model. The position of the cross section is indicated by the dotted black lines in (b).

time function used for the computation of seismograms is defined by a causal Ricker wavelet with a center frequency of 10 Hz. In the case of the ray-Born modeling methods, we include the direct wave traveling in the water layer, but we do not calculate the seafloor reflection.

With a water layer, take-off angles for ray tracing are limited by the critical angle at the water-sediment interface. Therefore, we do not have any real valued raypaths beyond these angles. This limited ray coverage creates an artificial boundary in the scattering volume, with an associated end-point signal (Bender and Orszag, 1999). To avoid these artifacts, traveltime and amplitude values are interpolated by including points just above the seafloor from ray tracing in the water layer, and adding these to the collection of spatial locations from one-point ray tracing before the Delaunay tessellation process. Figure 8 shows traveltimes and amplitudes for a source at $\mathbf{s} = (2500, 5000, 200)$m after interpolation on the uniform grid. Figure 8 also shows the interpolated ray shadow zone, the triangular wedges between the dashed line and the seafloor (the solid line).

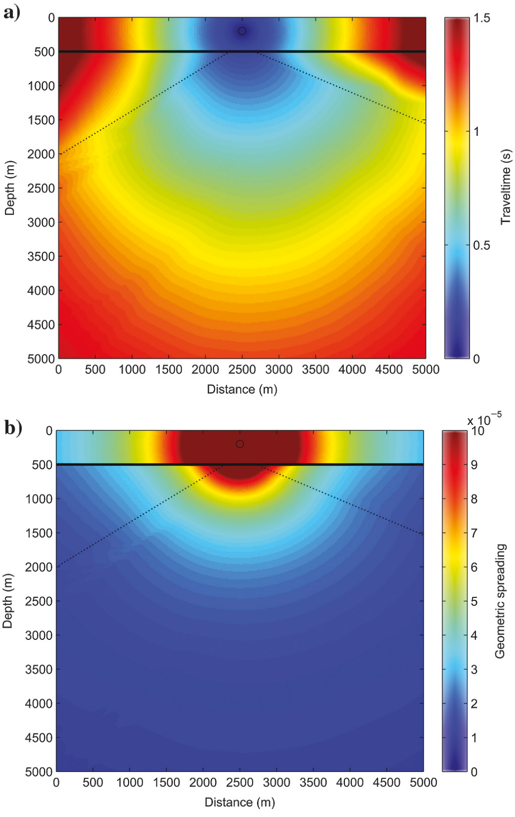


Figure 8. Cross section for two-point (3D) ray tracing in the overthrust model for (a) traveltimes and (b) amplitudes. The region between the water-sediment interface (solid black lines) and the dotted black lines indicate the ray shadow zones, in which the traveltimes and amplitudes are interpolated.

For a source positioned at $\mathbf{s} = (2500, 5000, 200)$ m and a receiver positioned at $\mathbf{r} = (2500, 5000, 100)$ m, Figure 9 shows isochron surfaces in the overthrust background model for times $T_{rs} = 1.8$ s (Figure 9a) and $T_{rs} = 2.36$ s (Figure 9b). Values on the surfaces are computed with the integrand in equation 36. Because the background model is heterogeneous, the isochron surfaces look like deformed spheroids as expected.

Ray-Born and finite-difference zero-offset sections

We show an example of zero-offset synthetics computed across a region with complicated faulting as well as a structurally simpler area with near-horizontal layers. Sources (at a depth of 100 m) and receivers (at a depth of 200 m) are positioned from 500 to 5000 m in the x-direction and at 5000 m in the y-direction. This is indicated by the dotted black line in Figure 7b. With an x, y aperture of 4000 m, the wavefield is computed using the two ray-Born-modeling methods and the finite-difference-modeling method.

Seismograms, computed using finite-difference and ray-Born modeling, are shown in Figure 10, for three positions in the overthrust model. The three positions are at 1000, 2500, and 4000 m, as indicated with the dashed blue lines in Figure 11. The seismograms

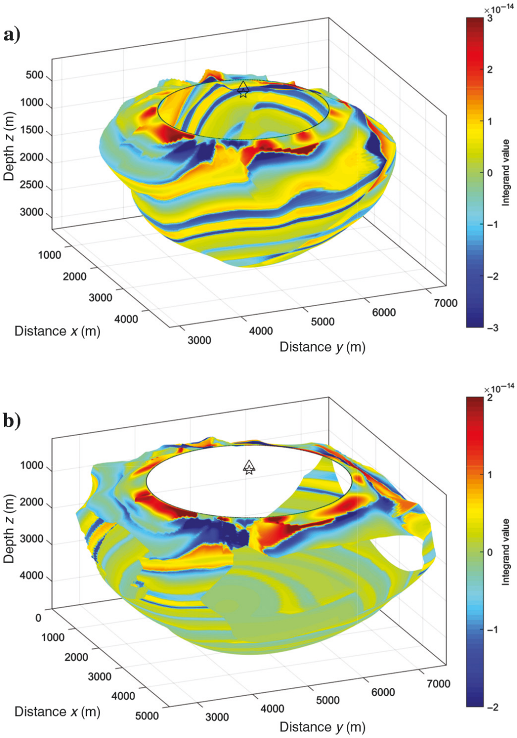


Figure 9. Isochron surfaces in a heterogeneous background, with integrand values for the overthrust model at times (a) $t = 1.8$ and (b) $t = 2.36$ s. For the later arrival times in (b), the bounding box of the total integration area intersects the surface. The source and receiver positions are indicated by a star and triangle, respectively.

in Figure 10a (1000 m) are extracted from a region with less complicated wave-propagation effects (the velocity model has horizontal and dipping layers). Here, the comparison with the finite differences shows that the phase and amplitude appear to be well-matched, indicating the relative importance of singly scattered waves. We also compare the seismograms shown in Figure 10b (2500 m). In this case, the source and receiver are located in a region where the velocity model has steeply dipping layers and faulted structures. These structures cause complicated wave-propagation effects (such as multiply scattered events) that appear as differences in the modeling results after traveltimes of approximately 1.4 s. In Figure 10c (4000 m), the early-time waveform mismatch in the time interval 1−1.4 s is caused by a triplication in the wavefield that is formed by the shallow syncline. Perturbations at the syncline are relatively strong (up to 30%) and are therefore not accurately captured by the ray-Born approximation.

A zero-offset section for the ray-Born synthetics is shown in Figure 11a and for finite differences in Figure 11b. Strong reflections from the horizontal and dipping layers are indicated in the blue rectangle labeled by I, confirming what was discussed above for Figure 10a. The wavefield becomes more complicated below the anticline, indicated by the blue rectangle labeled II. In this area, ray-Born synthetics exhibit stronger reflections than the finite-difference synthetics. This is likely an effect of multiple scattering and multipathing: Internal reflections in the finite-difference synthetic interfere with the main reflections, and these internal reflections do not exist in the case of ray-Born modeling. In the blue rectangle labeled III, we have indicated the triplication effect that causes the mismatch at early times in Figure 10c. This particular effect is present in the finite-difference synthetic, but poorly modeled in the ray-Born synthetic.

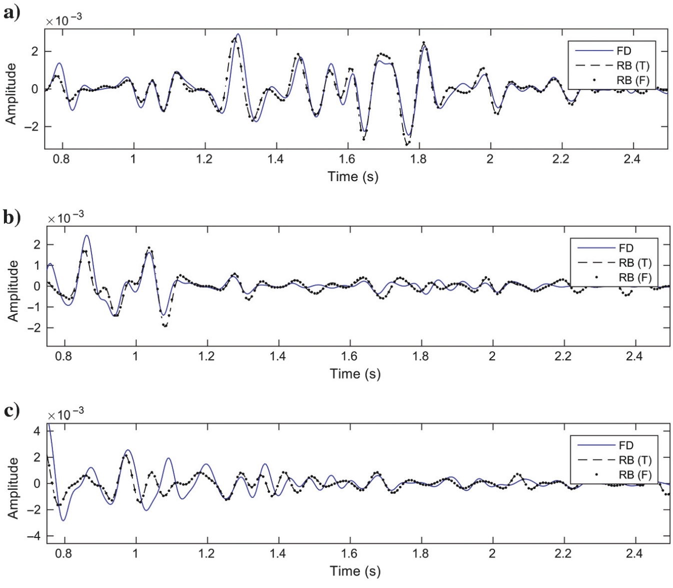


Figure 10. Zero-offset synthetics for the overthrust model at distances: (a) 1000, (b) 2500, and (c) 4000 m.

DISCUSSION

The main results of this paper are the derivation and computation of ray-Born integrals as integrals over isochron surfaces. First, we give the continuous ray-Born integral 21 and then derive the band-limited ray-Born integral 36, which is used in the numerical implementation. In the derivation of the band-limited integral, we use the normal distance from the isochron S_t to $S_{t+\Delta t/2}$ to reduce a volume integral to a surface integral. Then, we described a numerical integration over this surface in equation 40, where the integration is carried out over triangles.

The concepts discussed above are derived for the acoustic case, but they can be applied to the elastic case and also to the generalized ray-Born integrals (Chapman, 2004). For an elastic isotropic model, a natural starting point is to carry out the derivations from equations 21–36 using the elastic ray-Born integral (e.g., Dahlen et al. [2000], equation 56) as a starting point.

Waveforms computed using this new method are compared with ray-Born integrals computed on a regular grid in the frequency domain. This latter method is accurate as long as sampling intervals for the integration grid are kept small; however, relative to the new time domain method, the frequency-domain method is not as computationally efficient. This is because a summation is carried out over all volume elements in the scattering region and the summation is repeated for every frequency. The numerical efficiency of the time-domain isochron ray-Born method largely depends on the efficiency of the method used to compute the isochrons. If the isochrons are readily available, as they could be in a 1D background model, the integration over triangles is very cheap and the computational cost is insignificant compared with the frequency-domain method.

In the ray-Born integration method by Spencer et al. (1997) and Keers et al. (2002) (see also Chapman, 2004), the computations are made accurate in the time domain by having an integration grid that closely follows isochrons. This is important for modeling at higher frequencies because the isochron surface tends to have irregular curvature. However, compared with the ray-Born integrals using the isochrons explicitly, as discussed in this paper, the integration for the tetrahedron method is carried out over volumes and with a more involved implementation.

It should be noted that isochrons play an important role in many seismic methods because they associate the traveltime of a point in the subsurface with a source-receiver pair. Some examples are Kirchhoff imaging (Bleistein, 1987), the computation of Fresnel volumes (Červený and Soares, 1992), and finite-frequency kernels (Dahlen et al., 2000). Isochrons have also been studied by Iversen (2004), who presents a system of differential equations to compute isochron rays in a similar manner to how conventional rays are computed using the ray equations.

We also compare the new ray-Born modeling method with finite-difference modeling to assess the singly scattered wavefield as an approximation to the complete wavefield, where multiple scattering and multipathing are present. We broadly conclude that, for weak perturbations, or for regions

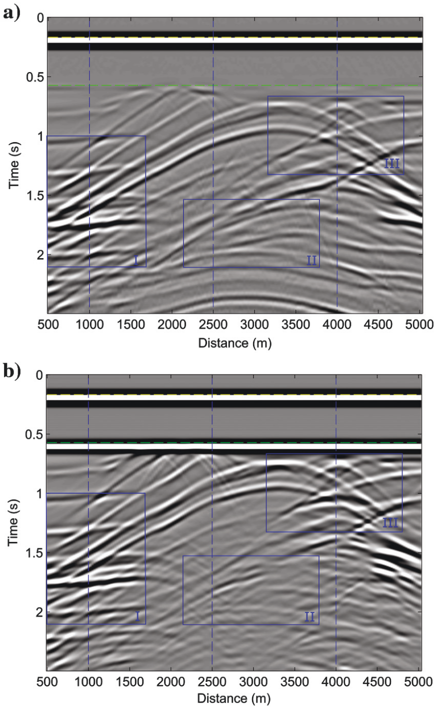


Figure 11. The 3D overthrust model synthetics for (a) ray-Born and (b) finite-difference modeling. The dashed blue lines indicate positions where individual traces are extracted for comparison. In the simple region (box I), singly scattered waves are important, but in the other two regions (boxes II and III), the wave-propagation effects are more complicated.

where the wavefield to a large extent consists of singly scattered waves, the ray-Born approximation is the most successful. Based on its ability to produce multiple-free data, it is natural to consider the ray-Born approximation as a modeling method for full-waveform inversion (e.g., during early iterations when the background model is slowly varying) or for the purpose of testing various inversion and imaging algorithms. Furthermore, there are applications in target-oriented studies, for example, in the modeling of the seismic responses caused by small and localized scatterers.

CONCLUSION

We have presented a physically intuitive method for computing 3D ray-Born integrals over isochrons in the time domain. The integrals are simplified because volume integrals are split in a natural way to several surface integrals. Each surface integral gives the waveform in the ray-Born integral for a certain time. The integration technique was used to compute ray-Born seismograms for two velocity models with different scattering characteristics: slowly and rapidly varying velocities. For both model types, the seismograms computed over surface integrals in the time domain are identical to ray-Born seismograms computed over volumes in the frequency domain.

ACKNOWLEDGMENTS

We thank our colleagues at Schlumberger for their constructive comments that led to improvements of the manuscript. In particular, we thank J. Ø. H. Bakke, M. Nickel, and L. Sønneland. The work was funded by the Norwegian Research Council through the Petromaks2 program, project number #245587. This research was partly done while H. Keers was on sabbatical at the University of Kiel and the University of Washington, and support for this work (by the L. Meltzer Høyskolefond, the Faculty of Mathematics and Natural Sciences of the University of Bergen, the Institute of Geoscience at the Christian Albrechts University of Kiel, the Applied Physics Laboratory at the University of Washington, and especially hosts T. Meier [CAU] and B. Odom [UW]) is gratefully acknowledged.

REFERENCES

Aminzadeh, F., B. Jean, and T. Kunz, 1997, 3-D salt and overthrust models: SEG.

Baig, A. M., and F. A. Dahlen, 2004, Traveltime biases in random media and the S-wave discrepancy: Geophysical Journal International, **158**, 922–938, doi: 10.1111/j.1365-246X.2004.02341.x.

Bender, C. M., and S. A. Orszag, 1999, Advanced mathematical methods for scientists and engineers I: Springer.

Beylkin, G., 1985, Imaging of discontinuities in the inverse scattering problem by inversion of a causal generalized radon transform: Journal of Mathematical Physics, **26**, 99–108, doi: 10.1063/1.526755.

Bleistein, N., 1987, On the imaging of reflectors in the earth: Geophysics, **52**, 931–942, doi: 10.1190/1.1442363.

Carcione, J. M., G. C. Herman, and A. P. E. ten Kroode, 2002, Seismic modeling: Geophysics, **67**, 1304–1325, doi: 10.1190/1.1500393.

Červený, V., 2005, Seismic ray theory: Cambridge University Press.

Červený, V., and D. Coppoli, 1992, Ray-Born synthetic seismograms for complex structures containing scatterers: Journal of Seismic Exploration, **1**, 191–206.

Červený, V., and J. E. P. Soares, 1992, Fresnel volume ray tracing: Geophysics, **57**, 902–915, doi: 10.1190/1.1443303.

Chapman, C., 2004, Fundamentals of seismic wave propagation: Cambridge University Press.

Chapman, C. H., 1978, A new method for computing synthetic seismograms: Geophysical Journal of the Royal Astronomical Society, **54**, 481–518, doi: 10.1111/j.1365-246X.1978.tb05491.x.

Coates, R. T., and C. Chapman, 1991, Generalized Born scattering of elastic waves in 3-D media: Geophysical Journal International, **107**, 231–263, doi: 10.1111/j.1365-246X.1991.tb00823.x.

Coates, R. T., and E. E. Charrette, 1993, A comparison of single scattering and finite difference synthetic seismograms in realizations of 2-D elastic random media: Geophysical Journal International, **113**, 463–482, doi: 10.1111/j.1365-246X.1993.tb00900.x.

Dahlen, F. A., S.-H. Hung, and G. Nolet, 2000, Fréchet kernels for finite-frequency traveltimes. I: Theory: Geophysics Journal International, **141**, 157–174, doi: 10.1046/j.1365-246X.2000.00070.x.

de Berg, M., M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, 2000, Computational geometry: Springer.

Etgen, J. T., and M. J. O'Brien, 2007, Computational methods for large-scale 3D acoustic finite-difference modeling: A tutorial: Geophysics, **72**, no. 2, SM223–SM230, doi: 10.1190/1.2753753.

Farin, G. E., 2002, Curves and surfaces for CAGD: A practical guide: Morgan Kaufmann.

Gibson, R. L., M. N. Toksöz, and F. Batini, 1993, Ray-Born modeling of fracture-zone reflections in the Larderello geothermal field: Geophysical Journal International, **114**, 81–90, doi: 10.1111/j.1365-246X.1993.tb01468.x.

Griffel, D. H., 2002, Applied functional analysis: Dover Publications.

Haddon, R., and P. Buchen, 1981, Use of Kirchhoff's formula for body wave calculations in the earth: Geophysical Journal International, **67**, 587–598, doi: 10.1111/j.1365-246X.1981.tb06939.x.

Hudson, J., 1977, Scattered waves in the coda of P: Journal of Geophysics, **43**, 359–374.

Hudson, J. A., and J. R. Heritage, 1981, The use of the Born approximation in seismic scattering problems: Geophysical Journal International, **66**, 221–240, doi: 10.1111/j.1365-246X.1981.tb05954.x.

Ikelle, L. T., and L. Amundsen, 2005, Introduction to petroleum seismology: SEG.

Iversen, E., 2004, The isochron ray in seismic modeling and imaging: Geophysics, **69**, 1053–1070, doi: 10.1190/1.1778248.

Keers, H., C. Chapman, and D. Nichols, 2002, A fast integration technique for the generation of ray-Born synthetics: 64th Annual International Conference and Exhibition, EAGE, Extended Abstracts, P108.

Knopoff, L., and J. Hudson, 1964, Scattering of elastic waves by small inhomogeneities: The Journal of the Acoustical Society of America, **36**, 338–343, doi: 10.1121/1.1918957.

Lorensen, W. E., and H. E. Cline, 1987, Marching cubes: A high resolution 3D surface construction algorithm: ACM Siggraph Computer Graphics, **21**, 163–169, doi: 10.1145/37402.

Miller, D., M. Oristaglio, and G. Beylkin, 1987, A new slant on seismic imaging: Migration and integral geometry: Geophysics, **52**, 943–964, doi: 10.1190/1.1442364.

Minakov, A., H. Keers, D. Kolyukhin, and H. C. Tengesdal, 2017, Acoustic waveform inversion for ocean turbulence: Journal of Physical Oceanography, **47**, 1473–1491, doi: 10.1175/JPO-D-16-0236.1.

Morse, P. M., and H. Feshbach, 1953, Methods of theoretical physics: McGraw-Hill.

Moser, T. J., 2012, Review of ray-Born forward modeling for migration and diffraction analysis: Studia Geophysica et Geodaetica, **56**, 411–432, doi: 10.1007/s11200-011-9046-0.

Operto, S., G. Lambaré, P. Podvin, and P. Thierry, 2003, 3D ray+Born migration/inversion. Part 2: Application to the SEG/EAGE overthrust experiment: Geophysics, **68**, 1357–1370, doi: 10.1190/1.1598129.

Press, W. H., S. A. Teukolsky, and W. T. Vetterling, 2007, Numerical recipes: The art of scientific computing, 3rd ed.: Cambridge University Press.

Šachl, L., 2013, Validation of 3D synthetic seismograms based on the ray-Born approximation: Studia Geophysica et Geodaetica, **57**, 84–102, doi: 10.1007/s11200-012-0722-5.

Snieder, R., 2001, General theory of elastic wave scattering, *in* R. Pike and P. Sabatier, eds., Scattering and inverse scattering in pure and applied science: Academic Press Inc., 528–541.

Spencer, C., C. H. Chapman, and J. E. Kragh, 1997, A fast, accurate integration method for Kirchhoff, Born and Maslov synthetic seismogram generation: 67th Annual International Meeting, SEG, Expanded Abstracts, 1838–1841.

Sutton, P. M., C. D. Hansen, H.-W. Shen, and D. Schikore, 2000, A case study of isosurface extraction algorithm performance, *in* W. C. de Leeuw and R. van Liere, eds., Data visualization 2000: Springer, 259–268.

Tengesdal, H. C. B., 2013, Ray-Born modeling and full waveform inversion: Master thesis, The University of Bergen.

Thierry, P., G. Lambaré, P. Podvin, and M. S. Noble, 1999, 3-D preserved amplitude prestack depth migration on a workstation: Geophysics, **64**, 222–229, doi: 10.1190/1.1444518.

Virieux, J., and G. Lambaré, 2015, Theory and Observations: Body Waves, Ray Methods and Finite — Frequency Effects,, *in* Treatise on geophysics: Elsevier.

Paper II

A copy of paper II is included in this chapter.

GEOPHYSICS, VOL. 84, NO. 5 (SEPTEMBER-OCTOBER 2019); P. S425–S436, 10 FIGS., 4 TABLES.
10.1190/GEO2018-0510.1

Ray-based modeling and imaging in viscoelastic media using graphics processing units

Martin Sarajaervi[1] and Henk Keers[2]

ABSTRACT

In seismic data processing, the amplitude loss caused by attenuation should be taken into account. The basis for this is provided by a 3D attenuation model described by the quality factor Q, which is used in viscoelastic modeling and imaging. We have accomplished viscoelastic modeling and imaging using ray theory and the ray-Born approximation. This makes it possible to take Q into account using complex-valued and frequency-dependent traveltimes. We have developed a unified parallel implementation for modeling and imaging in the frequency domain and carried out the numerical integration on a graphics processing unit. A central part of the implementation is an efficient technique for computing large integrals. We applied the integration method to the 3D SEG/EAGE overthrust model to generate synthetic seismograms and imaging results. The attenuation effects are accurately modeled in the seismograms and compensated for in the imaging algorithm. The results indicate a significant improvement in computational efficiency compared to a parallel central processing unit baseline.

INTRODUCTION

Amplitude loss in seismic data caused by attenuation should be taken into account in processing algorithms such as seismic modeling and imaging. The basis for this is provided by a 3D attenuation model described by the quality factor Q (Aki and Richards, 2009). The building and use of 3D Q models (Ribodetti and Virieux, 1998; Cavalca et al., 2011) in seismic algorithms have proven to be important to correct for the distortion of seismic waveforms caused by attenuation. For example, below gas or mud regions (e.g., Traynin et al., 2008), the distortion, due to attenuation, can be significant.

A general account of viscoelastic wave propagation is given by Carcione (2015). This book contains a comprehensive review for isotropic and anisotropic media. We accomplish viscoelastic waveform modeling using ray tracing (Červený, 2005) in combination with the viscoelastic ray-Born approximation (e.g., Dahlen et al., 2000). For viscoelastic ray tracing, the theory has been developed for complex rays and real rays (Kravtsov et al., 1999; Hanyga and Seredyńska, 2000; Vavryčuk, 2012), with further developments for real rays in anisotropic media (Vavryčuk, 2008, 2010). Complex and real viscoelastic ray tracing in attenuating media are important developments in ray theory and useful in practice for regions with attenuation effects. One mainly uses real rays because this is computationally more efficient. This is valid only in weakly attenuating media.

The ray-Born integrals used for viscoelastic modeling are also a fundamental starting point for deriving an imaging condition for viscoelastic media. In the perfectly elastic case, imaging conditions based on the Born and ray-Born integrals were derived by Tarantola (1986) and Beylkin and Burridge (1990), respectively. The ray-Born integrals are closely related to the Kirchhoff integrals (Jaramillo and Bleistein, 1999) used for imaging by Bleistein (1987). For inversion in attenuating media, a comprehensive theoretical background was given by Tarantola (1988). In the more recent literature, attenuation was included using Kirchhoff integrals by Keers et al. (2001), Traynin et al. (2008), Xie et al. (2009), and Wu et al. (2017). In these studies, complex-valued and frequency-dependent traveltimes are obtained from real ray tracing and, for this reason, the integration was carried out in the frequency domain. We formulate attenuation using the same principle. However because we describe a modeling and an imaging algorithm, it is natural to use ray-Born integrals rather than Kirchhoff integrals, so that the scatterers are volume elements.

Manuscript received by the Editor 3 August 2018; revised manuscript received 19 March 2019; published ahead of production 03 June 2019; published online 24 August 2019.

[1]Schlumberger, Risabergveien 3, Stavanger 4068, Norway. E-mail: msarajaervi@slb.com.
[2]University of Bergen, Realfagbygget, Allégt., Bergen 5020, Norway. E-mail: henk.keers@uib.no.

Ray-based methods are efficient when compared with fully numerical methods such as finite differences. Unfortunately, for modeling and imaging in large 3D models with multiple source-receiver pairs, even when ray-based methods are used, the computations may still run for days or weeks. This is particularly true in the case of viscoelastic media in which the velocity model is frequency dependent. Therefore, it is important to be able to speed up the computations. This can be done using parallel computing hardware such as the graphics processing unit (GPU). To achieve this, software must be written that can efficiently run sequential operations in parallel.

Several authors have studied the parallel implementation of seismic processing algorithms. For example, imaging using the acoustic Kirchhoff integral was studied by Panetta et al. (2009), Brouwer et al. (2011), and Shi et al. (2011). These authors present methods to compute images on the GPU and use a trace-based parallel strategy. This resulted in a significant increase in computational efficiency for acoustic Kirchhoff imaging. The compensation of attenuation effects using viscoacoustic ray theory and Kirchhoff imaging was done by Han and Sun (2015). They compute individual image points on the GPU and use a so-called image-domain parallel strategy for a multi-GPU system. Also in this case, the authors reported a significant increase in efficiency compared to a central processing unit (CPU) baseline. A fast method for solving the eikonal and transport equations on a GPU was presented by Noack and Gillberg (2015). Several other authors have also present parallel methods for finite-difference-based modeling (Weiss and Shragge, 2013), imaging (Wang et al., 2019), and waveform inversion (Yang et al., 2015).

We present the use of GPU acceleration for numerical evaluation of large integrals that appear in two important seismic applications. The first application is viscoelastic ray-Born modeling, and the second application is the computation of seismic images. In both instances, we also describe a queueing system on the CPU for passing the ray data efficiently to the GPU.

We first give a brief account for the quantities needed to compute ray-Born integrals and derive an expression for the viscoelastic imaging condition. Then, we describe the numerical and the parallel implementations in a unified way. The modeling and imaging, with and without attenuation, are then applied to the 3D SEG/EAGE overthrust model (Aminzadeh et al., 1997), and the computational performance is assessed.

THEORY

The viscoelastic modeling and viscoelastic imaging can be done efficiently using viscoelastic ray theory in a 3D heterogeneous background model (e.g., Hanyga and Seredyńska, 2000; Chapman, 2004; Červený, 2005; Vavryčuk, 2008). In general, the background model must be slowly varying, but a few interfaces can be added to improve accuracy in the case of sharp velocity contrasts. Commonly, the background model is based on seismic interval velocities that have been calibrated using well information with subsequent reflection tomography updates (e.g., Woodward et al., 2008). In the background model $\{c_0, Q\}$ with wave velocity c_0 and quality factor $Q \gtrsim 40$, raypaths are characteristics of the eikonal equation and are computed by solving the kinematic ray equations (e.g., Červený, 2005). Solutions to the eikonal equation are complex-valued in attenuating media (Vavryčuk, 2012), but we ignore this effect for the ray geometry and include only the complex-valued traveltimes. Along individual raypaths parameterized by arc length l, we compute the traveltime $T' = \int (1/c_0)\mathrm{d}l$ and the attenuation time $T^* = \int (1/c_0 Q)\mathrm{d}l$ by integration along the raypath. Here, we denote the traveltime from a source position $\mathbf{s}$ to a scattering point $\mathbf{x}$ as $T'(\mathbf{s},\mathbf{x})$ and from a receiver position $\mathbf{r}$ to a scattering point $\mathbf{x}$ as $T'(\mathbf{r},\mathbf{x})$. Likewise for the attenuation time T^*. The polarization vectors $\mathbf{p}$ are also obtained along raypaths and, in this instance, we use a subscript notation $\mathbf{p}_s$ and $\mathbf{p}_r$ to indicate vectors leaving from the source $\mathbf{s}$ and arriving at the receiver $\mathbf{r}$, respectively. Note that we have separate polarization vectors for the SH, SV, and P-waves, and we assume that the P- and S-waves are linearly polarized, rather than elliptically polarized (Carcione, 2006). The polarization vectors are also assumed to be real valued; therefore, we incorporate attenuation effects in the traveltimes only and not directly in the amplitudes. This assumption is practical for numerical computations, but it is not a theoretical limitation for ray-Born integrals. Furthermore, using the dynamic ray equations (e.g., Červený, 2005), we find the geometric spreading A, denoted in a similar manner as the traveltimes by $A(\mathbf{r},\mathbf{x})$ and $A(\mathbf{s},\mathbf{x})$. The real raypaths are computed separately for P- and S-waves by using the substitution $c_0 \rightarrow \alpha_0$ and $Q \rightarrow Q_p$ for P-waves, and likewise for S-waves using $c_0 \rightarrow \beta_0$ and $Q \rightarrow Q_s$.

Ray-Born modeling

Once the ray quantities described in the previous section are available, the scattered wavefield $\mathbf{u}_1$ is computed using ray-Born integrals. The explicit expression for the viscoelastic ray-Born integral (Dahlen et al., 2000) is

$$\mathbf{u}_1(\mathbf{r},\mathbf{s},\omega) = \omega^2 f(\omega) \sum_{\mathrm{P,S}} \int [\mathbf{M} : \mathbf{k}_s(\mathbf{x})\mathbf{p}_s(\mathbf{x}) A_{rs}(\mathbf{x}) \\ \times [\mathbf{\Omega}(\mathbf{x}) \cdot \mathbf{m}_1(\mathbf{x})] e^{-i\omega T_{rs}(\mathbf{x},\omega)} \mathbf{p}_r(\mathbf{x})]\mathrm{d}\mathbf{x}, \qquad (1)$$

with angular frequency ω, source-time function f, moment tensor $\mathbf{M}$, and unit polarization vector $\mathbf{k}_s = \mathbf{p}_s/|\mathbf{p}_s|$. The amplitude factor A_{rs} is

$$A_{rs}(\mathbf{x}) = A(\mathbf{r},\mathbf{x})A(\mathbf{s},\mathbf{x}) \qquad (2)$$

and T_{rs} is the complex-valued traveltime function

$$T_{rs}(\mathbf{x},\omega) = T(\mathbf{r},\mathbf{x},\omega) + T(\mathbf{s},\mathbf{x},\omega). \qquad (3)$$

For a reference frequency ω_0, the complex-valued traveltime from $\mathbf{r}$ to $\mathbf{x}$ is given by

$$T(\mathbf{r},\mathbf{x},\omega) = T'(\mathbf{r},\mathbf{x}) - \frac{1}{2} i T^*(\mathbf{r},\mathbf{x}) - \frac{1}{\pi} T^*(\mathbf{r},\mathbf{x}) \ln(\omega/\omega_0) \qquad (4)$$

with the traveltime T' and the attenuation time T^* as described above. The model parameter vector $\mathbf{m}_1 = (\alpha_1, \beta_1, \rho_1)$ in equation 1 contains the model perturbations for the P-wave velocity α_1, S-wave velocity β_1, and density ρ_1. Combined with the background medium α_0, the actual medium is $\alpha = \alpha_0 + \alpha_1$ in the case of the P-wave velocity and similarly for the S-velocity and density. The Rayleigh factors $\mathbf{\Omega}$ are given by Dahlen et al. (2000, Table 1), where the different components of $\mathbf{\Omega}$ are described by the polarization vectors for the incoming and outgoing waves at the scattering positions.

Imaging

For viscoelastic imaging, we follow the method outlined by Tarantola (1984, 1986, 1988) and seek a solution for minimizing the objective function M with respect to the model perturbation $\mathbf{m}_1$. Here, M is the difference between the recorded 3C waveform $\mathbf{d}$ and the corresponding synthetics $\mathbf{u}_1$. The difference is taken for all sources, receivers, and angular frequencies in the least-squares norm:

$$M(\mathbf{m}_1) = \int |\mathbf{d}(\mathbf{r},\mathbf{s},\omega) - \mathbf{u}_1(\mathbf{r},\mathbf{s},\omega,\mathbf{m}_1)|^2 \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{s}\mathrm{d}\omega. \quad (5)$$

The observed waveform $\mathbf{d}$ is here assumed to be preconditioned in the sense that direct waves, refracted waves, and multiples were removed so that only the singly scattered wavefield remains.

Now, we derive the imaging condition. For the purpose of brevity, we rewrite the ray-Born integral from equation 1 using only the perturbation $\mathbf{m}_1 = \alpha_1$ and we consider PP scattered waves only. Furthermore, we use a more compact notation and write the ray-Born integral 1 as

$$\mathbf{u}_1(\mathbf{a}) = \int \mathbf{v}(\mathbf{a},\mathbf{x})\alpha_1(\mathbf{x})\mathrm{d}\mathbf{x}, \quad (6)$$

where $\mathbf{a} = (\mathbf{r},\mathbf{s},\omega)$ is the acquisition vector and

$$\mathbf{v}(\mathbf{a},\mathbf{x}) = \omega^2[f(\omega)\mathbf{M}(\omega):\mathbf{k}_s(\mathbf{x})\mathbf{p}_s(\mathbf{x})] \times [\alpha_0^{-1}(\mathbf{x})A_{rs}(\mathbf{x})e^{-i\omega T_{rs}(\mathbf{x})}\mathbf{p}_r(\mathbf{x})]. \quad (7)$$

Here, the P-wave background model α_0 is present because we use the explicit Rayleigh factor for PP scattering (Dahlen et al., 2000) rather than $\mathbf{\Omega}$. If we insert $\mathbf{u}_1$, as defined in equation 6, in equation 5, and expand the square of the objective function (for details, see Appendix A), then we obtain the normal equations in integral form

$$\int \mathbf{d}(\mathbf{a})\overline{\mathbf{v}(\mathbf{a},\mathbf{x})}\mathrm{d}\mathbf{a} = \int \overline{\mathbf{v}(\mathbf{a},\mathbf{x})}\mathbf{v}(\mathbf{a},\mathbf{y})\alpha_1(\mathbf{y})\mathrm{d}\mathbf{a}\mathrm{d}\mathbf{y} \quad (8)$$

with a new variable $\mathbf{y}$ denoting the spatial scattering positions. A solution to equation 8 in terms of α_1 can be obtained if we use a numerical inversion algorithm such as LSQR (Paige and Saunders, 1982). This is often not practical or necessary if we want to compute structural images of the subsurface. Instead, we use a diagonal approximation of the Hessian. This is accomplished by inserting a delta function $\delta(\mathbf{x}-\mathbf{y})$ on the right side of equation 8 and results in an explicit expression for α_1 (the so-called imaging condition):

$$\alpha_1(\mathbf{x}) \approx \left[\int \mathbf{d}(\mathbf{a})\overline{\mathbf{v}(\mathbf{a},\mathbf{x})}\mathrm{d}\mathbf{a}\right]\left[\int |\mathbf{v}(\mathbf{a},\mathbf{x})|^2\mathrm{d}\mathbf{a}\right]^{-1}. \quad (9)$$

Compensation for attenuation effects (Q) in the imaging process is accounted for by the conjugate of $\mathbf{v}$ as shown on the left side in equations 8 and 9.

UNIFIED IMPLEMENTATION

Viscoelastic modeling using equation 1 and viscoelastic imaging using equation 9 require evaluating multidimensional integrals using precomputed ray data. We present a straightforward implementation for both types of integrals, in three steps, which can be applied to both situations: (1) ray tracing for all sources and receivers in the background model, (2) initialization of the ray data on the CPU, and (3) computation of the integrals using a GPU.

Ray tracing

Amplitudes, traveltimes, and polarization vectors are computed in the viscoelastic background model along real raypaths by solving the real ray equations, given the initial conditions such as the source position and take-off angles. For a given source $\mathbf{s}$ and receiver $\mathbf{r}$, this is the two-point real ray-tracing problem of finding a real ray from $\mathbf{s}$ to $\mathbf{r}$. In 3D media, it is common to use triangulation and an interpolation scheme on a bundle of rays to solve the problem (e.g., Vinje et al., 1993; Kraaijpoel, 2003), but a variety of other techniques have been published (Červený, 2005, section 3.8). In some situations, the real two-point ray tracing is simpler, for example, in a homogeneous or a 1D medium. For the modeling and imaging examples, discussed below, we have solved the two-point ray-tracing problem analytically in a homogeneous medium. However, rather than computing the quantities directly, the values were stored as ray tables, as would be the case if the ray tracing had been done in heterogeneous media.

In the following sections, we assume that the real ray tables are readily available on a regular grid, but with a different sorting order, depending on the application. If the application is modeling, then the ray data should first be sorted by source-receiver pairs, and then by scattering points. If the application is imaging, then the ray data are first sorted by scattering points and then by source-receiver pairs. This is not a requirement, but it ensures that we read ray data efficiently for each application.

Parallel computations using CPU and GPU

Parallel strategies for assigning computational tasks across multiple CPU or GPU cores should be used for the efficient implementation of seismic algorithms. Fundamental to each such strategy is the use of threads to run programmed instructions in parallel. Moreover, we need to be able to start threads programmatically, keep track of them, and eventually retrieve the computational results (Williams, 2012). This is done differently for a CPU than for a

Table 1. A comparison of the computation time for viscoelastic ray-Born modeling for the GPU method with the computation time of the multicore CPU baseline.

	8 Haswell cores	64 Haswell cores
	Integration grid: $20 \times 20 \times 10$ m	
Tesla K80	22.9	3.2
Tesla P100	39.5	5.5
	Integration grid: $20 \times 20 \times 5$ m	
Tesla K80	27.3	3.8
Tesla P100	54.5	7.5

Note: Values larger than one favor the GPU implementation. The results are computed in the overthrust model for two GPU configurations and two CPU configurations.

GPU, and it also depends on programming language-specific methods or specialized programming interfaces.

GPU hardware was originally designed for fast rendering of computer graphics, but it later found use in general purpose computing as well as in scientific computing. Its advantage over a CPU lies in its ability to run a large number of threads in parallel, using a large number of cores. This differs from a CPU, which has fewer, but more powerful cores, and therefore runs a much smaller number of threads in parallel.

For the management of threads on a CPU, there are various options. These options depend on the programming language that is used. Some programming languages have built-in support for threads, whereas other programming languages may require the use of additional programming libraries. We have used the C++ language in which thread creation and manipulation is part of the standard library since the 2011 standard (Stroustrup, 2013). However, it is still up to the programmer to write the code that keeps track of the various threads that are running as well as their outcome. We create threads from a parent process and let each thread run until its computational task is finished. The outcome of the computations is then communicated to the parent. To do this, we make use of a data type that is "thread safe," which means that its content can be shared between different threads. In our implementation, we have used a thread-safe queue (Williams, 2012), which is a so-called container type in which data can be inserted or removed.

For the management of threads on a GPU, we use the CUDA application programming interface (e.g., Nickolls et al., 2008). This programming interface can be used in combination with various programming languages such as C or C++, and it is also used for GPU memory management. In a typical CUDA-based program, the data are first moved from the CPU memory to the GPU global memory. Once the data reside in GPU memory, programmed instructions can be executed across a large number of GPU threads. Each thread can process part of the data and store the outcome of the computation in the GPUs memory. Finally, after completion of all threads, the data are copied from the GPU and its memory, back to the CPU and its memory.

The computational procedure presented in this paper is outlined using a flow diagram (see Figure 1) that consists of two main processes: *A* and *B*. Process *A* describes the parallel initialization, and process *B* describes the GPU-accelerated integral evaluation. In the case of modeling (equation 1), we compute integrals over the spatial coordinates, for each source-receiver-frequency point. In the case of imaging (equation 9), we compute integrals over sources, receivers, and frequencies, for each image point.

We discuss our use of GPUs in more detail in the next two subsections. In the first subsection, we discuss how the data are prepared for the GPU. In the second subsection, we discuss how these data are used to efficiently compute the integral equation 1 (modeling) or 9 (imaging).

Parallel initialization

This section concerns the part of the flow diagram that is called process *A* in Figure 1. Process *A* consists of five steps with each represented by a numbered box in the flow diagram in Figure 1.

In step 1, we create a queue with no members but with room for member data types. Each member contains a set of vectors with data for the numerical integration. At the beginning, the queue is empty because integral data have not yet been prepared.

In step 2, we start the main loop by getting a new integral from a list of integrals to be processed. We then move on to step 3. If the queue length is smaller than a maximum queue length, then we move on to step 4. Otherwise, we wait for the queue length to be reduced. Reduction of the queue length is discussed in the next section on GPU acceleration.

In step 4, we perform the parallel initialization across the CPU threads, as indicated by multiple arrows for the data that flow through CPU thread 1 to CPU thread *N*. Here, *N* is the maximum number of threads. In each thread, we create a new member data type, as discussed above, and we fill the member vectors with data. Various tasks are performed in each of the threads. The most important ones for the seismic applications are the actual reading of integrand tables from disk (e.g., ray tables), interpolation onto a suitable grid (if needed), and contraction of quantities that do not depend on the integration domain. Once a thread has finished, the member and its data are moved to the queue in step 5. At this step, the data are ready to be used in step 7 of process *B* in which the numerical integration is done. This is discussed in more detail in the next section.

The maximum queue size depends on the size of the available CPU memory. For example, if one member type requires 10% of the memory, then we can safely use a maximum queue size of nine whereby we would use, at a maximum, 90% of the available memory. In this case, up to nine members with vector data are pending to be passed on to the GPU memory.

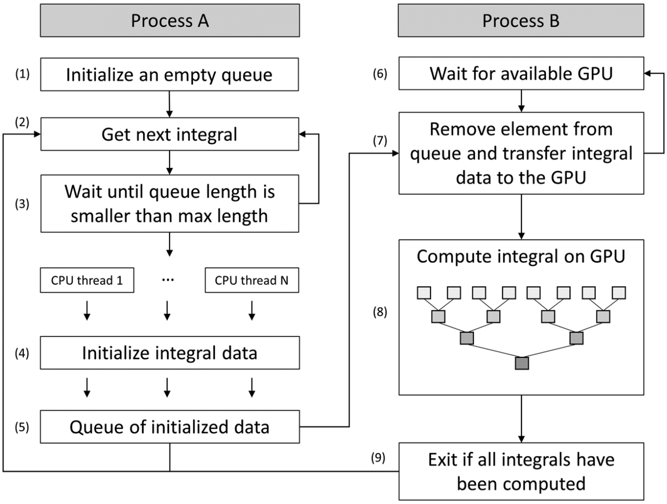

Process A
Process B
(1)
Initialize an empty queue
(2)
Get next integral
(3)
Wait until queue length is smaller than max length
CPU thread 1
···
CPU thread N
(4)
Initialize integral data
(5)
Queue of initialized data
(6)
Wait for available GPU
(7)
Remove element from queue and transfer integral data to the GPU
(8)
Compute integral on GPU
(9)
Exit if all integrals have been computed

Figure 1. Flowchart describing the parallel initialization of integrals using CPU threads and the GPU-accelerated numerical integration shown as a reduction tree in step 8.

GPU accelerated integral evaluation

Process B in Figure 1 shows the part of the diagram in which the evaluation of integrals in equation 1 or 9 is done. For this, one or multiple GPUs are used in collaboration with the data initialization process A. We start process B in step 6 by waiting for an available GPU. Once a GPU is available, we move to step 7 in which we get the integrand from members of the queue in step 5 (process A). If a member exists in the queue, we get its vectors with the integrand data and remove the member from the queue. The data are then moved from the CPU and its memory to the GPU and its memory. In this step, we free up space in the queue and memory used by the vectors and also the GPU receives the data needed for the numerical integration.

In step 8, the numerical integration is carried out as a summation over a list of elements. The list size is determined by the size of the integration domain, and the list elements are either spatial volume elements (in the case of modeling) or contributions from different sources, receivers, and frequencies (in the case of imaging).

Because summation is associative $(a + (b + c) = (a + b) + c$, for real numbers a, b, c) and commutative $(a + b = b + a$, for real numbers a, b), it is natural to split the integration into many partial sums so that these partial sums can be computed in parallel. In a specific computer implementation, an algorithm based on a tree structure can be used for this purpose. Using a tree, we start at the leaf nodes of the tree and compute partial sums for two nodes at the same time (these are the white squares in Figure 1, step 8). When all partial sums for these nodes are completed, we pass the results down to the parent nodes in which new partial sums are computed. This continues until the root of the tree is reached, and the final result is available (the black squares in Figure 1, step 8). This procedure is similar to a sum reduction, but with one modification. We evaluate the integrands of equation 1 or 9 already on the leaf nodes of the tree before passing the results downward rather than just summing all nodes. This has the advantage that the integrands are evaluated in a highly parallel manner.

In practice, we have a so-called computation grid that is made up of multiple thread blocks, in which each thread block is a group of GPU threads. Across the computation grid, we distribute the calculation of equation 1 or 9 as mentioned above. Because many blocks and threads run at the same time on the GPU, the integral evaluations become very efficient.

Once the integrand evaluations at the leaf nodes of the tree are done, the reduction is performed in each thread block to compute partial sums. To maximize efficiency of the reduction, low-level techniques can be used here to efficiently exchange the partial sums between threads (or tree nodes). For CUDA, Harris (2007) suggest strategies that are efficient at reducing large data sets, which is important for seismic applications. The tests presented by Harris (2007) show that by carefully improving the CUDA kernels so that partial results are communicated optimally between threads, we can achieve significant benefits in efficiency. These results were later improved upon by Luitjens (2014) who takes advantage of new hardware instructions for efficient communication of partial sums between GPU threads. We adopt this low-level algorithm for the reduction. Finally, after all the GPU threads in each of the thread blocks are completed, we compute the final sum over only the blocks. This completes the numerical integration for one integral.

In step 9, we perform a check to see if all integrals have been computed. If there are no new integrals to compute in step 2 and the queue in step 5 is empty, then the check holds true and we exit. Otherwise, the computational loops continue. In practical applications in which we compute a large number of integrals, steps 6–8 will be done many times and continue as long as the queue in step 5 contains data.

Handing complex numbers on the GPU

The real and imaginary parts of the integrands are evaluated separately on the GPU using equation 3, and by rewriting the complex exponential in equations 1 and 7 as

$$e^{-i\omega T_{rs}(\mathbf{x})} = e^{-\frac{1}{2}\omega[T^*(\mathbf{r},\mathbf{x})+T^*(\mathbf{s},\mathbf{x})]}(\cos[\omega\tau(\mathbf{x})] + i\ \sin[\omega\tau(\mathbf{x})]). \tag{10}$$

Here, T' is the previously described traveltime, T^* is the attenuation time, and τ is defined as

$$\tau(\mathbf{x}) = \pi^{-1}\ln(\omega/\omega_0)[T^*(\mathbf{r},\mathbf{x}) + T^*(\mathbf{s},\mathbf{x})] - T'(\mathbf{r},\mathbf{x}) - T'(\mathbf{s},\mathbf{x}). \tag{11}$$

This is a practical choice which, compared to using other data types that handle complex numbers, leads to increased computational performance.

Parallel CPU baseline

To compare the GPU implementation with an efficient implementation using CPU only, we implemented the modeling and imaging algorithms on multicore CPU. For the modeling, we compute sets of frequencies in parallel (using CPU threads); for imaging, we compute multiple image points in parallel.

The numerical integration in this case is also carried out in an optimized manner using the Intel SSE3 instructions (e.g., Hassaballah et al., 2008) for all mathematical operations that are performed multiple times (e.g., the integral kernels) and with the sum reduction done using the horizontal addition instruction (e.g., function *_mm_hadd_ps*).

RESULTS

Modeling

We demonstrate the validity of the waveforms computed with the viscoelastic ray-Born integrals in three ways. First, by comparing the computed seismograms with a closed-form solution for the far field in homogeneous viscoelastic media (Vavryčuk, 2007). Second, by comparing ray-Born seismograms in a 1D layered medium with viscoelastic seismograms computed using finite differences (Graves, 1996). Finally, we compute synthetic seismograms in the overthrust model (Aminzadeh et al., 1997). The seismograms computed in the overthrust model are input data for the imaging integrals. We also assess the performance of the GPU method in the overthrust model.

Ray-based seismograms compared with the viscoeleastic closed-form solution

Computation of waveforms using the real-valued ray geometry, discussed in the "Theory" section above, is compared with the exact viscoelastic Green's function (Vavryčuk, 2007; equation A-1) for the direct S-wave at 4.5 km (Figure 2). We consider only the far-field term of the exact solution in Vavryčuk (2007) because only the far-field is modeled in ray theory. For a very high attenuation with $Q = 10$, Figure 2a shows the real ray tracing (the solid) and the exact solution (the dashed) for Ricker wavelets with center frequencies of 10, 30, and 60 Hz. Synthetics corresponding to the higher Q-values, 20 and 40, are shown in Figure 2b and 2c, respectively. These numerical modeling results show that the real-valued ray approximation is accurate in weakly attenuating media ($Q \gtrsim 40$) for the propagation distances under consideration. For very low values of Q, the approximation is less accurate.

Ray-Born seismograms compared with viscoelastic finite differences

For the comparison with finite-difference modeling, we compute seismograms for a 3D medium with (1D) layered perturbations (Figure 3). The perturbations are small relative to the background model. This ensures that the effects of multiple scattering and multipathing are not very strong (for a more in-depth discussion and examples, see Sarajaervi and Keers, 2018). Figure 4 shows that, under these assumptions, there is a close agreement between the ray-Born and finite-difference seismograms. The values for Q in the background model are, respectively, 80, 40, and 20. The perturbations remain fixed. We compare the viscoelastic ray-based waveforms with a finite-difference implementation presented by Graves

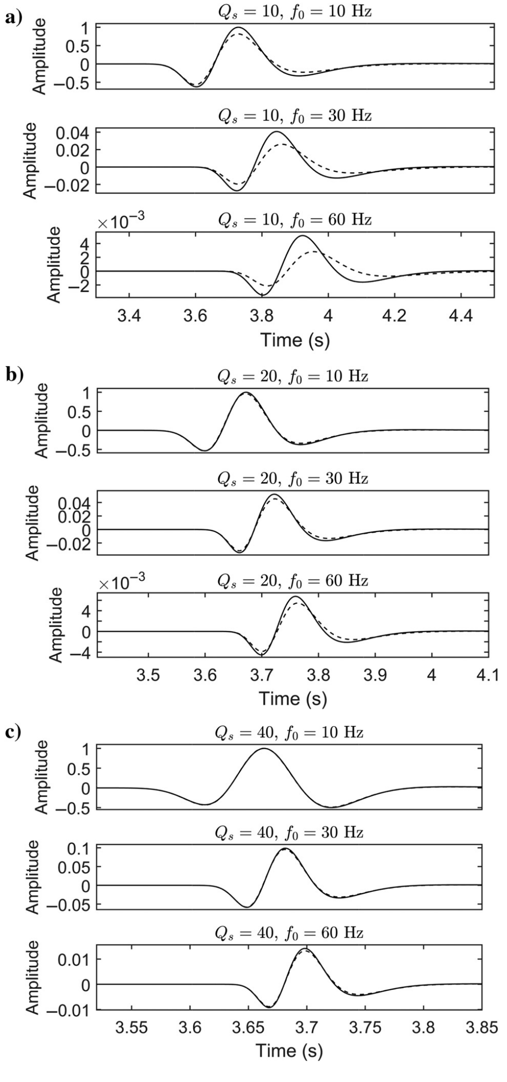


Figure 2. Seismogram for an S-wave at a distance of 4.25 km from the source position. Real ray tracing seismogram (solid) and the closed-form solution, given by Vavryčuk (2007; far-field term in equation A-1) (dashed) are shown for various frequencies and Q values. The amplitudes are normalized based on the seismogram for $f_0 = 10$ Hz in each of (a-c).

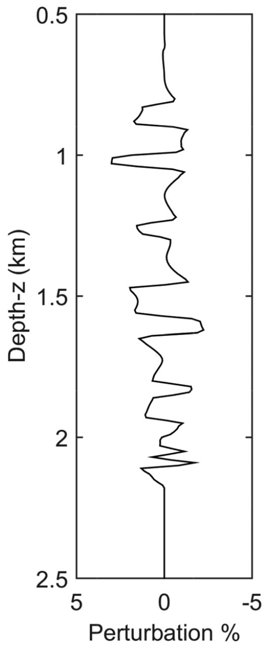


Figure 3. Perturbation strength for P- and S-wave scattering as a percentage, relative to the elastic background models. For these weak perturbations, the wave-propagation effects attributed to multipathing and multiple scattering are small, and the effects of attenuation in the ray-Born and finite-difference methods can be compared.

(1996). In this implementation, Q is assumed to be independent of frequency whereby the convolutional relations between the stresses and strains are avoided (for details, see Graves, 1996, equations 19–21). We choose the center frequency of the Ricker wavelet as the reference frequency. Figure 4 shows that the differences between ray-Born seismograms and the finite-difference seismograms increase with increasing attenuation (i.e., decreasing values of Q). As in the case of the closed-form comparison, these results also show that the real ray approximation is useful for weakly attenuating media ($Q \gtrsim 40$). We note that the finite-difference implementation slightly over-attenuates the lower frequencies for our choice of reference frequency, and that various other methods have been presented for viscoelastic finite-difference modeling in which a system of memory variables is used (e.g., Carcione, 1993; Moczo et al., 2014). However, these methods come at a higher computational cost.

Overthrust model

We apply the GPU acceleration method to the computation of synthetic seismograms for the 3D SEG/EAGE overthrust model (Aminzadeh et al., 1997). Precomputed ray tables for a constant density background with velocities $\alpha_0 = 3$ km/s and $\beta_0 = 2$ km/s and quality factors $Q_p = Q_s = 50$ are used. The overthrust model (Figure 5) is modified along the edges using a cosine taper to avoid end-point signals. The model extends from 0 to 9 km in the x- and y-directions and has a depth of 1.8 km. Sources are positioned along the x-direction, $s_x \in [0.8, 8.2]$ km at a 50 m interval, and at $s_y = 4.5$ km (149 sources in total). All sources are modeled as explosions (so that the only nonzero components of the moment tensor $\mathbf{M}$ are its diagonal elements: $M_{xx} = M_{yy} = M_{zz} = 1$) and a Ricker wavelet with a peak frequency of 20 Hz is used. The receivers are positioned in the x-direction at $r_x = s_x$km with a 50 m interval and in the y-direction at $r_y \in [4.4, 4.6]$ km with a 100 m interval. In Figure 5, the receiver positions (447 receivers in total) are indicated using black dots.

Modeling results for the overthrust model are shown in Figure 6 for a source positioned at $s_x = s_y = 4.5$ km using an integration grid of $451 \times 451 \times 181$ points based on sampling intervals of $20 \times 20 \times 10$ m. For PP scattering, the shot record is shown for waveforms modeled without attenuation in Figure 6a and with attenuation using $Q_p = 50$ in Figure 6b. Figure 6c shows all modes (PP, PS, SP, and SS) without attenuation and, likewise, in Figure 6d, with attenuation using $Q_p = Q_s = 50$. In this example, all converted modes originate from a P-wave because an isotropic moment tensor was used, as mentioned above.

Figure 7 shows trace-by-trace comparisons of the modeling results at $(r_x, r_y) = (4.5, 3.2)$ km. In Figure 7a, we show PP scattering without attenuation (the solid black line) with attenuation $Q_p = 100$ (the dashed blue line) and with attenuation $Q_p = 50$ (the dashed red line). Figure 7b shows all wave modes (PP, PS, SP, and SS) without attenuation (the solid black line), with $Q_p = Q_s = 100$ (the dashed blue line) and $Q_p = Q_s = 100$ (the dashed red line). Finally, Figure 7c shows a comparison of only PP scattering and scattering from all four waves modes, both with attenuation.

Imaging

For the imaging application, we first computed ray-Born seismograms for all source-receiver pairs as described above. We created a

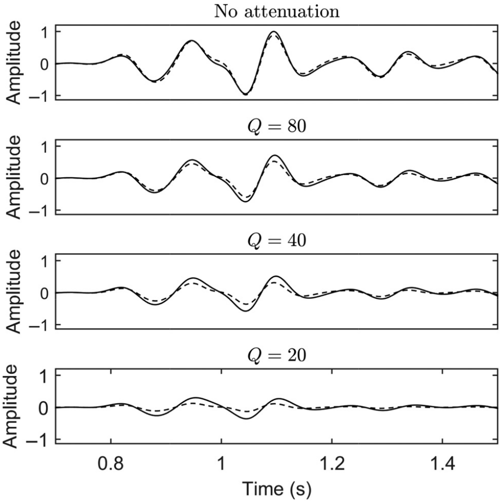


Figure 4. Ray-Born seismograms (solid) are compared with seismograms computed using finite-differences (dashed) for the layered model in Figure 3. The seismograms are computed without attenuation and for $Q = \{80, 40, 20\}$ from, respectively, the top figure to the bottom figure. The amplitudes are normalized based on the seismogram without attenuation.

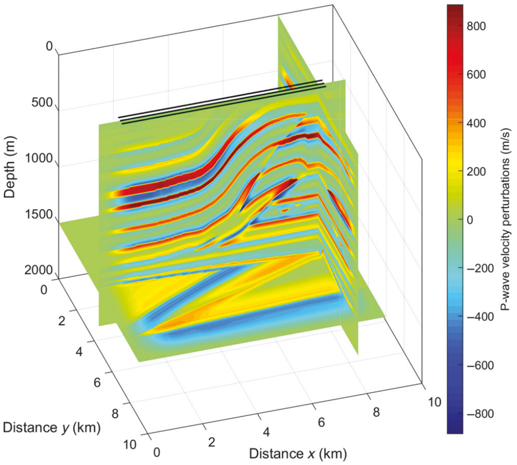


Figure 5. Cross sections and depth slices for perturbations α_1 in the 3D SEG/EAGE overthrust model. The black dots indicate receiver positions that are used in the numerical examples.

data set to be used as input for the imaging using only PP scattered waves. Therefore, only images were computed for PP-waves. The modeling used in this case is done without attenuation and with attenuation (using $Q_p = 50$).

Figure 8 shows image sections across a faulted anticline in the overthrust model. First, for comparison, Figure 8a shows the result when the PP modeling and imaging are done without attenuation. Then, Figure 8b shows the result when the PP modeling and imaging are done with attenuation using $Q_p = 50$. These two images (Figure 8a and 8b) are almost identical. The modeled data with attenuation are accurately corrected by the imaging condition with attenuation, as expected. In Figure 8c, the modeling is done using $Q_p = 50$, and the imaging is done without including attenuation. This result shows that the amplitude loss caused by attenuation has not been compensated because attenuation was not included in the imaging. As a result, the quality of the image deteriorates.

A detailed trace-by-trace comparison is shown in Figure 9 for receivers at $r_x = 4.5$ km with $r_y = 2.0$ (Figure 9a), $r_y = 4.8$ (Figure 9b), and $r_y = 6.8$ km (Figure 9c). The viscoelastic imaging result (the solid black) is compared with the imaging result without attenuation (the dashed black) and, as a reference, we show results when the modeling and the imaging are done without attenuation (the black dots). The latter comparison shows that, using the imaging condition in equation 9, we correctly compensate for the amplitude loss. As in the case of the results discussed in the previous paragraph, these results show in more detail that attenuation should be included in the imaging process. If that is not the case, then the amplitudes will incorrectly decay with increasing depth and this will negatively affect the quality of the images.

Performance comparison

In this subsection, we compare the performance of the GPU-accelerated algorithm with the multicore CPU baseline. We measured

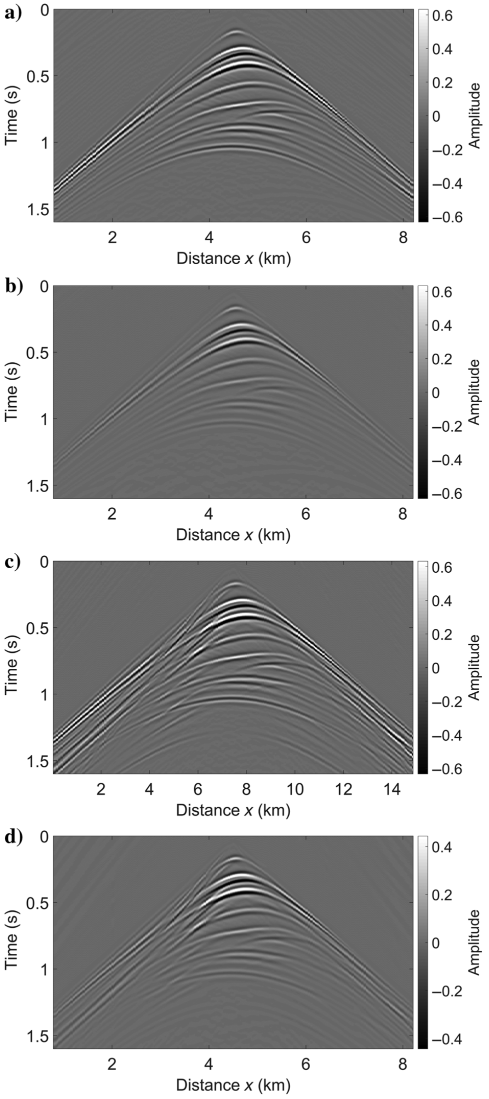


Figure 6. Modeling results in the 3D overthrust model for waveforms with (a) no attenuation with only PP scattering, (b) attenuation using $Q_p = 50$ with only PP scattering, (c) no attenuation with all scattering modes (PP, PS, SP, and SS) included, and (d) attenuation using $Q_p = Q_s = 50$ with all of the scattering modes (PP, PS, SP, and SS) included.

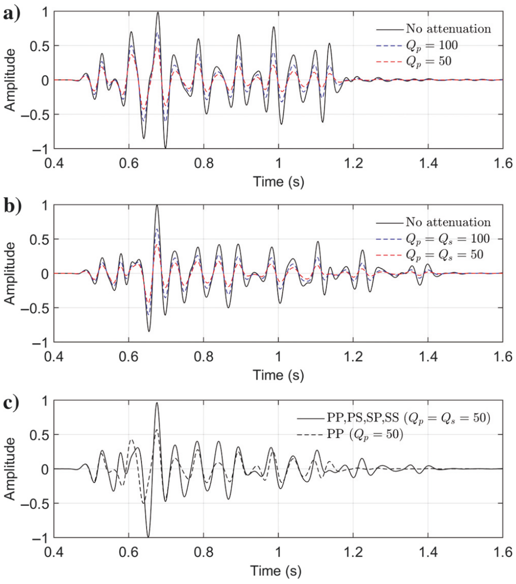


Figure 7. Single-trace comparison for synthetic seismograms in the 3D SEG/EAGE overthrust model shown at the distance $x = 3.2$ km (see the shot records in Figure 6): (a) only PP scattering, with and without attenuation, (b) all scattering modes (PP, PS, SP, and SS), with and without attenuation, and (c) all scattering modes (PP, PS, SP, and SS) and only PP scattering, both with attenuation.

the computation time using virtual machines that are widely available to the scientific community and commercial actors on cloud infrastructures. The tests for the GPU method ran on a virtual machine with either an NVIDIA Tesla K80 GPU (less powerful) or an NVIDIA Tesla P100 GPU (more powerful). These results are compared to the baseline implementation running on virtual machines with either 8 or 64 virtual Haswell cores at 2.30 GHz.

For the modeling comparison, we measured the average runtime for computing one source-receiver pair using the CPU divided by the runtime for the same computation done using the GPU. Thus, a ratio greater than one means that the GPU computations are more efficient. The results are summarized in Table 1 for an integration grid of 20 × 20 × 10 m that was used to create the synthetic seismograms for our numerical examples. This is compared to results computed on a denser grid of 20 × 20 × 5 m (needed, for example, for modeling higher frequencies). The results show that the GPUs are significantly faster for modeling, even when we use a powerful node with up to 64 CPU cores.

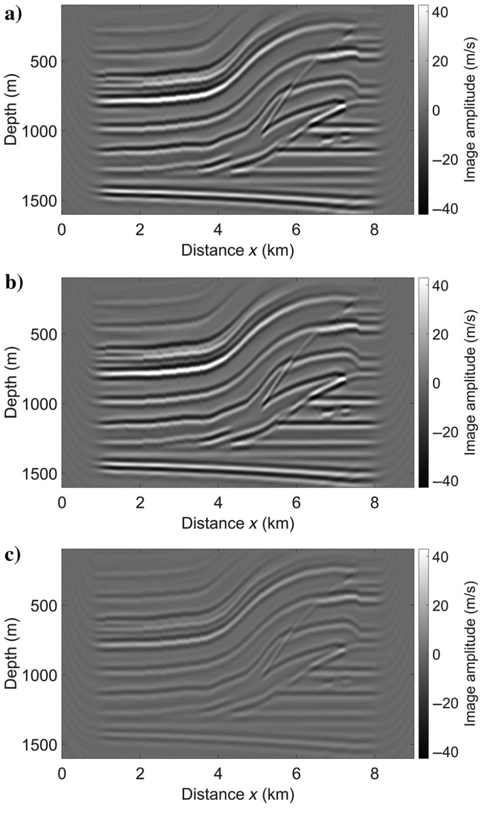


Figure 8. Imaging results in the 3D SEG/EAGE overthrust model shown for a vertical cross section across a faulted anticline: (a) modeling and imaging done without attenuation, (b) modeling and imaging done using $Q_p = 50$, and (c) modeling with attenuation using $Q_p = 50$, but imaging done without attenuation.

To assess the computational efficiency, without the initialization step, we run a second set of performance tests for the modeling and evaluate only the sum reductions. For the tests, we vary the spatial sampling of the integration domain on the assumption that denser grids represent domains with larger spatial extents. The reason for this is that the larger domains have a higher computational burden than the smaller domains. In Figure 10, the *x*-axis denotes the number of integration points. Sampling intervals of 20 × 20 × 10 m are used for the smallest domain (the same grid as used in the "Results" section for modeling), and the sampling intervals of the largest domain are 10 × 10 × 10 m. In Figure 10a, the *y*-axis denotes the total computation time and the results are shown for the baseline (red), the K80 GPU (black), and the P100 GPU (blue) runs. The values are calculated as the total time used to carry out the integrations for all frequencies. The same performance comparison is shown in Figure 10b, but here as a speed-up factor for the computation times for GPUs relative to the CPU baseline. The results show that, regardless of the domain size, the K80 and P100 GPUs are, respectively, 6 and 15 times more efficient than the baseline when computing the multidimensional integrals.

Table 2 shows the computational performance results for the imaging. In this case, we measured the total runtime required to compute the imaging result shown in Figure 8b. We performed the imaging using 240 frequencies (up to 60 Hz) in one case and using 120 frequencies (up to 30 Hz) in another case. As in the modeling example, the results show that the GPU reduction scheme is significantly more efficient than the CPU baseline.

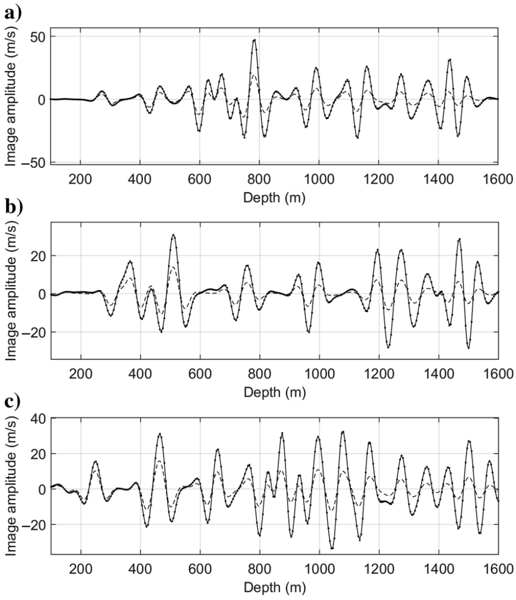


Figure 9. Imaging results shown for single traces in the black dots for modeling and imaging done without attenuation, in the solid black for modeling and imaging done with $Q_p = 50$ and in the dashed black for modeling with $Q_p = 50$, and imaging without attenuation. Traces are selected at the *x*-locations along the imaged vertical cross section at (a) 2 km, (b) 4.8 km, and (c) 6.8 km.

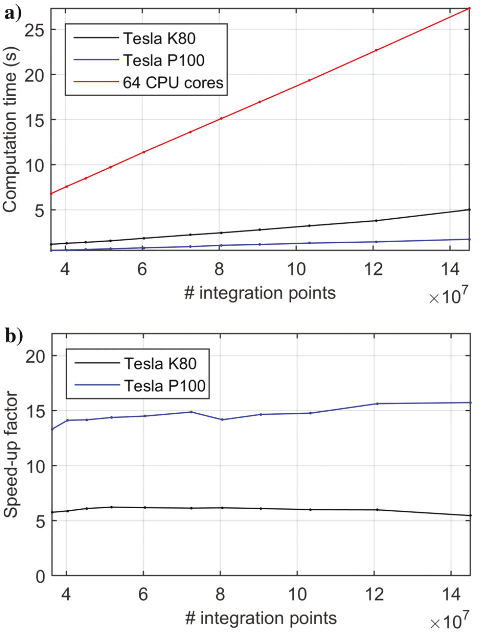


Figure 10. Performance comparison for the computation of ray-Born integrals for various sizes of the integration domain: (a) the time in seconds for computing one source-receiver pair and (b) the speed-up achieved for the GPU architectures as compared with the 64 CPU core baseline.

DISCUSSION

Increasing the computational efficiency is especially important for the processing of large seismic surveys. For example, a marine seismic acquisition may contain hundreds of thousands of shots and may take weeks to process on a cluster of nodes using conventional CPU-based techniques. For practical applications of this size, we note that our strategy is perfectly parallel and can also be implemented for multiple nodes or GPUs without difficulty. In the case of the modeling application, one can process source-receiver pairs separately on different nodes. Similarly, one can process image points separately for the imaging application. If the integration domain is very large and does not fit in the GPU memory, the domain can be decomposed into smaller domains that are passed to multiple GPUs in parallel or to one GPU sequentially.

In our numerical results, the performance comparisons in the 3D SEG/EAGE overthrust model suggest that a node with 64 virtual CPU cores cannot outperform a single GPU. Therefore, given the hardware configurations discussed in this paper, using a GPU is more cost effective. We also observe that if we increase the computation domain, then the speed-up factor remains the same and we do not saturate the GPU capacity.

The integration method used is based on straightforward summation. It could be thought that using more sophisticated integration algorithms that use irregular grids or higher order integration methods, such as Gaussian quadrature techniques (Stroud and Secrest 1966), would also lead to faster ray-Born modeling algorithms. However, initial tests indicate that this is, in general, not the case. The reason for this is that the integrand in the ray-Born integral varies quite randomly, except near the source and receiver, as the perturbation often has a “random” component (e.g., when it is derived from tomography). This confirms the usefulness of more hardware-oriented methods to speed up ray-Born modeling.

Table 2. A comparison of the computation time for viscoelastic imaging for the GPU method with the computation time of the multicore CPU baseline.

	8 Haswell cores	64 Haswell cores
Sources × receivers × frequencies: 149 × 447 × 120		
Tesla K80	12.2	2.3
Tesla P100	29.0	5.5
Sources × receivers × frequencies: 149 × 447 × 240		
Tesla K80	19.7	4.0
Tesla P100	48.4	9.7

Note: Values larger than one favor the GPU implementation.

CONCLUSION

We presented a unified implementation strategy for viscoelastic modeling and imaging using GPUs. Fundamental to the implementation is efficient numerical integration using a sum reduction method. The reduction is done in a tree-like manner across the GPU cores, using a general purpose reduction algorithm. This has the advantage that further developments of reduction algorithms, for example, with developments of new hardware, can easily be picked up from the field of high-performance computing for use in geophysical algorithms.

Comparisons to the closed-form solutions and the finite-difference method show that real ray tracing is a useful approximation in weakly attenuating media. Furthermore, we show the significance of amplitude decay in the modeling of synthetic seismograms, and we show how attenuation effects are compensated for in the imaging algorithm. Finally, the performance comparison shows a significant improvement in computational efficiency as opposed to a parallel CPU baseline.

ACKNOWLEDGMENTS

We would like to thank our colleagues at Schlumberger for comments that led to improvements of the manuscript. The work is funded by the Norwegian Research Council through the Petromaks2 program, project number #245587. We are thankful to the following participants of the project: ConocoPhilips, NORSAR, Schlumberger, and the University of Bergen. This research was partly done while H. Keers was on sabbatical at the University of Cambridge. Support for this by the L. Meltzer Høyskolefond, the Faculty of Mathematics and Natural Sciences of the University of Bergen, and hospitality by the Department of Earth Sciences of the University of Cambridge is much appreciated.

DATA AND MATERIALS AVAILABILITY

The fundamental aspects of the algorithms we present are available in the public domain. References are included.

APPENDIX A

MINIMIZATION OF THE OBJECTIVE FUNCTION FOR IMAGING

The square of the objective function 5 is defined as

$$M(\alpha_1) = M_0 + \langle M_1, \alpha_1 \rangle + \langle H(\alpha_1), \alpha_1 \rangle, \tag{A-1}$$

when the complex inner product is defined by $\langle f, g \rangle = \int \overline{f(\mathbf{x})} g(\mathbf{x}) \mathrm{d}\mathbf{x}$, $\overline{f(\mathbf{x})}$ is the complex conjugate of $f(x)$ and

$$M_0 = \int \overline{\mathbf{d}(\mathbf{a})} \mathbf{d}(\mathbf{a}) \mathrm{d}\mathbf{a}, \tag{A-2}$$

$$M_1(\mathbf{x}) = -2 \int \overline{\mathbf{v}(\mathbf{a}, \mathbf{x})} \mathbf{d}(\mathbf{a}) \mathrm{d}\mathbf{a}, \tag{A-3}$$

$$H(\mathbf{x}, \alpha_1) = \int \overline{\mathbf{v}(\mathbf{a}, \mathbf{x})} \mathbf{v}(\mathbf{a}, \mathbf{y}) \alpha_1(\mathbf{y}) \mathrm{d}\mathbf{a}\mathrm{d}\mathbf{y}. \tag{A-4}$$

Equation A-4 defines the Hessian H with the new variable $\mathbf{y}$ denoting the spatial scattering points. The least-squares objective function A-1 can be differentiated with respect to the perturbation model $\mathrm{d}M/\mathrm{d}\alpha_1 = 0$, and this gives

$$M_1(\mathbf{x}) + 2H(\mathbf{x}, \alpha_1) = 0. \tag{A-5}$$

Inserting equations A-3 and A-4 into A-5 gives the normal equation 8 in integral form.

REFERENCES

Aki, K., and P. G. Richards, 2009, Quantitative seismology, 2nd ed.: University Science Books.

Aminzadeh, F., B. Jean, and T. Kunz, 1997, 3-D salt and overthrust models: SEG, SEG/EAGE 3-D Modeling Series 1.

Beylkin, G., and R. Burridge, 1990, Linearized inverse scattering problems in acoustics and elasticity: Wave Motion, **12**, 15–52, doi: 10.1016/0165-2125(90)90017-X.

Bleistein, N., 1987, On the imaging of reflectors in the earth: Geophysics, **52**, 931–942, doi: 10.1190/1.1442363.

Brouwer, W., V. Natoli, and M. Lamont, 2011, A novel GPGPU approach to Kirchhoff time migration: 81st Annual International Meeting, SEG, Expanded Abstracts, 3465–3469, doi: 10.1190/1.3627918.

Carcione, J. M., 1993, Seismic modeling in viscoelastic media: Geophysics, **58**, 110–120, doi: 10.1190/1.1443340.

Carcione, J. M., 2006, Vector attenuation: Elliptical polarization, raypaths and the Rayleigh-window effect: Geophysical Prospecting, **54**, 399–407, doi: 10.1111/j.1365-2478.2006.00548.x.

Carcione, J. M., 2015, Wave fields in real media: Wave propagation in anisotropic, anelastic, porous and electromagnetic media: Elsevier.

Cavalca, M., I. Moore, L. Zhang, S. L. Ng, R. Fletcher, and M. Bayly, 2011, Ray-based tomography for Q estimation and Q compensation in complex media: 81st Annual International Meeting, SEG, Expanded Abstracts, 3989–3993, doi: 10.1190/1.3628039.

Červený, V., 2005, Seismic ray theory: Cambridge University Press.

Chapman, C., 2004, Fundamentals of seismic wave propagation: Cambridge University Press.

Dahlen, F. A., S.-H. Hung, and G. Nolet, 2000, Frechet kernels for finite-frequency traveltimes — 1. Theory: Geophysics Journal International, **141**, 157–174, doi: 10.1046/j.1365-246X.2000.00070.x.

Graves, R. W., 1996, Simulating seismic wave propagation in 3D elastic media using staggered-grid finite differences: Bulletin of the Seismological Society of America, **86**, 1091–1106.

Han, F., and S. Z. Sun, 2015, GPU acceleration of amplitude-preserved Q compensation prestack time migration: Computers and Geosciences, **82**, 214–224, doi: 10.1016/j.cageo.2015.06.018.

Hanyga, A., and M. Seredyńska, 2000, Ray tracing in elastic and viscoelastic media: Pure and Applied Geophysics, **157**, 679–717, doi: 10.1007/PL00001114.

Harris, M., 2007, Optimizing parallel reduction in CUDA: NVIDIA Developer Technology, **2**, 70.

Hassaballah, M., S. Omran, and Y. B. Mahdy, 2008, A review of SIMD multimedia extensions and their usage in scientific and engineering applications: The Computer Journal, **51**, 630–649, doi: 10.1093/comjnl/bxm099.

Jaramillo, H. H., and N. Bleistein, 1999, The link of Kirchhoff migration and demigration to Kirchhoff and Born modeling: Geophysics, **64**, 1793–1805, doi: 10.1190/1.1444685.

Keers, H., D. W. Vasco, and L. R. Johnson, 2001, Viscoacoustic crosswell imaging using asymptotic waveforms: Geophysics, **66**, 861–870, doi: 10.1190/1.1444975.

Kraaijpoel, D. A., 2003, Seismic ray fields and ray field maps: Theory and algorithms: Ph.D. thesis, Utrecht University.

Kravtsov, Y. A., G. Forbes, and A. Asatryan, 1999, I Theory and applications of complex rays, *in* E. Wolf, ed., Progress in optics: Elsevier 39, 1–62.

Luitjens, J., 2014, Faster parallel reductions on Kepler, https://devblogs.nvidia.com/parallelforall/faster-parallel-reductions-kepler, accessed 9 January 2018.

Moczo, P., J. Kristek, and M. Gális, 2014, The finite-difference modelling of earthquake motions: Waves and ruptures: Cambridge University Press.

Nickolls, J., I. Buck, M. Garland, and K. Skadron, 2008, Scalable parallel programming with CUDA: Queue, **6**, 40–53, doi: 10.1145/1365490.

Noack, M., and T. Gillberg, 2015, Fast computation of Eikonal and transport equations on graphics processing units computer architectures: Geophysics, **80**, no. 5, T183–T191, doi: 10.1190/geo2014-0556.1.

Paige, C. C., and M. A. Saunders, 1982, LSQR: An algorithm for sparse linear equations and sparse least squares: ACM Transactions on Mathematical Software, **8**, 43–71, doi: 10.1145/355984.355989.

Panetta, J., T. Teixeira, P. R. de Souza Filho, C. A. da Cunha Finho, D. Sotelo, F. M. R. da Motta, S. S. Pinheiro, I. P. Junior, A. L. R. Rosa, and L. R. Monnerat, 2009, Accelerating Kirchhoff migration by CPU and GPU cooperation: Computer architecture and high performance computing: SBAC-PAD'09. 21st International Symposium on IEEE, 26–32.

Ribodetti, A., and J. Virieux, 1998, Asymptotic theory for imaging the attenuation factor Q: Geophysics, **63**, 1767–1778, doi: 10.1190/1.1444471.

Sarajaervi, M., and H. Keers, 2018, Computation of ray-Born seismograms using isochrons: Geophysics, **83**, no. 5, T245–T256, doi: 10.1190/geo2017-0669.1.

Shi, X., C. Li, S. Wang, and X. Wang, 2011, Computing prestack Kirchhoff time migration on general purpose GPU: Computers and Geosciences, **37**, 1702–1710, doi: 10.1016/j.cageo.2010.10.014.

Stroud, A. H., and D. Secrest, 1966, Gaussian quadrature formulas: Prentice-Hall.

Stroustrup, B., 2013, The C++ programming language: Pearson Education.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266, doi: 10.1190/1.1441754.

Tarantola, A., 1986, A strategy for nonlinear elastic inversion of seismic reflection data: Geophysics, **51**, 1893–1903, doi: 10.1190/1.1442046.

Tarantola, A., 1988, Theoretical background for the inversion of seismic waveforms including elasticity and attenuation: Pure and Applied Geophysics, **128**, 365–399, doi: 10.1007/BF01772605.

Traynin, P., J. Liu, and J. Reilly, 2008, Amplitude and bandwidth recovery beneath gas zones using Kirchhoff prestack depth Q-migration: 78th Annual International Meeting, SEG, Expanded Abstracts, 2412–2416, doi: 10.1190/1.3059363.

Vavryčuk, V., 2007, Asymptotic Green's function in homogeneous anisotropic viscoelastic media: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, **463**, 2689–2707.

Vavryčuk, V., 2008, Real ray tracing in anisotropic viscoelastic media: Geophysical Journal International, **175**, 617–626, doi: 10.1111/j.1365-246x.2008.03898.x.

Vavryčuk, V., 2010, Behaviour of rays at interfaces in anisotropic viscoelastic media: Geophysical Journal International, **181**, 1665–1677.

Vavryčuk, V., 2012, On numerically solving the complex Eikonal equation using real ray-tracing methods: A comparison with the exact analytical solution: Geophysics, **77**, no. 4, T109–T116, doi: 10.1190/geo2011-0431.1.

Vinje, V., E. Iversen, and H. Gjøystdal, 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166, doi: 10.1190/1.1443499.

Wang, Y., H. Zhou, X. Zhao, Q. Zhang, P. Zhao, X. Yu, and Y. Chen, 2019, Cu Q-RTM: A CUDA-based code package for stable and efficient Q-compensated RTM: Geophysics, **84**, no. 1, F1–F15, doi: 10.1190/geo2017-0624.1.

Weiss, R. M., and J. Shragge, 2013, Solving 3D anisotropic elastic wave equations on parallel GPU devices: Geophysics, **78**, no. 2, F7–F15, doi: 10.1190/geo2012-0063.1.

Williams, A., 2012, C++ concurrency in action: Manning Publications.

Woodward, M. J., D. Nichols, O. Zdraveva, P. Whitfield, and T. Johns, 2008, A decade of tomography: Geophysics, **73**, no. 5, VE5–VE11, doi: 10.1190/1.2969907.

Wu, X., Y. Wang, Y. Xie, J. Zhou, D. Lin, and L. Casasanta, 2017, Least square Q-Kirchhoff migration — Implementation and application: 79th Annual International Conference and Exhibition, EAGE, Extended Abstracts, doi: 10.3997/2214-4609.201700804.

Xie, Y., K. Xin, J. Sun, C. Notfors, A. K. Biswal, and M. Balasubramaniam, 2009, 3D prestack depth migration with compensation for frequency dependent absorption and dispersion: 79th Annual International Meeting, SEG, Expanded Abstracts, 2919–2923, doi: 10.1190/1.3255457.

Yang, P., J. Gao, and B. Wang, 2015, A graphics processing unit implementation of time-domain full-waveform inversion: Geophysics, **80**, no. 3, F31–F39, doi: 10.1190/geo2014-0283.1.

NORDIC ECOLABEL
241 731
PRINTED MATTER

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

UNIVERSITAS BERGENSIS

uib.no

ISBN: 9788230846957 (print)
9788230844267 (PDF)