Adjoint formulation for the elastic wave equation

Gustavo Alves

ABSTRACT

I present the formulation for the forward non-linear elastic wave equation and the linearized Born modeling approximation. The methodolgy used is similar to the adjoint formulation presented for the acoustic case by Almomin (2013). I also demonstrate how a perfectly matched absorbing layer (PML) can be applied to the formulation and derive the necessary adjoints for implementing inversion methods. Finally, I show results for a synthetic model using the linear approximation and compare it to the non-linear solution for a small localized perturbation. The linear solution converges to the non-linear one when the perturbation is small in relation to the background model.

INTRODUCTION

Seismic theory tells us that waves propagating in a linear elastic medium can be uniquely described by prescribing body forces and boundary conditions of a given problem (Aki and Richards, 1980). The solution is usually a combination of body waves, both pressure and shear, and surface waves. However, most current imaging techniques use only pressure waves. This is due both to the lack of available multicomponent data and the greater computational requirements of processing multicomponent data.

In recent years, however, this scenario has started to change. New technologies, such as ocean bottom cables (OBCs) and ocean bottom nodes (OBNs), have extended the ability to record multicomponent data to offshore acquisitions. New prospecting challenges also push the limits of current imaging methods, which in turn force the need for more detailed data. Consequently, a new collection of algorithms is required to process this new multicomponent data.

In this work, I present the velocity-stress formulation for the elastic wave-equation in the framework of adjoint methods. My goal is to cast the problem as a series of adjoint operators, so that both linearized forward methods and their adjoints can be clearly constructed.

I start by constructing two possible representations to the recursive operator for the 2D elastic wave equation, together with their respective adjoints. The recursive operator and its adjoint are the core elements for most wave propagation methods, such as Born modeling, Reverse Time Migration (RTM), Tomography and Wave

Equation Migration Velocity Analysis (WEMVA). Therefore, proposing different solutions to the recursive operator helps in better understanding how to develop such methods.

I follow by extending both solutions to include an absorbing boundary condition. I propose the use of a Perfectly Matched Layer (PML), according to Collino and Tsogka (2001).

Finally, I describe the implementation of one of the proposed recursive operators for the Born operator. The Born operator is a linearized approximation of the non-linear wave equation and accurately estimates the non-linear solution for small perturbations of the background wavefield. I show synthetic examples for a point scatterer and compare the results to the non-linear case.

METHODOLOGY

The elastic wave equation for a two dimensional problem can be written in the velocity-stress formulation as a set of 5 equations,

$$\rho(\mathbf{x})\frac{\partial}{\partial t}V_x(\mathbf{x},t) - [\frac{\partial}{\partial x}\sigma_{xx}(\mathbf{x},t) + \frac{\partial}{\partial z}\sigma_{xz}(\mathbf{x},t)] = s_1(\mathbf{x},t) \quad (1)$$

$$\rho(\mathbf{x})\frac{\partial}{\partial t}V_z(\mathbf{x},t) - [\frac{\partial}{\partial x}\sigma_{xz}(\mathbf{x},t) + \frac{\partial}{\partial z}\sigma_{zz}(\mathbf{x},t)] = s_2(\mathbf{x},t) \quad (2)$$

$$\frac{\partial}{\partial t}\sigma_{xx}(\mathbf{x},t) = [\lambda(\mathbf{x}) + 2\mu(\mathbf{x})]\frac{\partial}{\partial x}V_x(\mathbf{x},t) + \lambda(\mathbf{x})\frac{\partial}{\partial z}V_z(\mathbf{x},t) + s_3(\mathbf{x},t) \quad (3)$$

$$\frac{\partial}{\partial t}\sigma_{zz}(\mathbf{x},t) = [\lambda(\mathbf{x}) + 2\mu(\mathbf{x})]\frac{\partial}{\partial z}V_z(\mathbf{x},t) + \lambda(\mathbf{x})\frac{\partial}{\partial x}V_x(\mathbf{x},t) + s_4(\mathbf{x},t) \quad (4)$$

$$\frac{\partial}{\partial t}\sigma_{xz}(\mathbf{x},t) = \mu(\mathbf{x})[\frac{\partial}{\partial z}V_x(\mathbf{x},t) + \frac{\partial}{\partial x}V_z(\mathbf{x},t)] + s_5(\mathbf{x},t), \quad (5)$$

where ρ, λ and μ are the model parameters, V_x and V_z are the particle velocities, σ_{xx}, σ_{zz} and σ_{xz} are the normal and shear stresses, respectively, s_1 and s_2 are the velocity components of the source and s_3, s_4 and s_5 are the stress components of the source.

My first goal is to write the forward non-linear modeling operator and its adjoint. To achieve that, I need to re-cast the previous set of equations as a recursive system. First, I apply a finite difference approximation to the time derivates, following the staggered-grid approach described by Virieux (1986) and Levander (1988). The equations become

$$V_x^n = V_x^{n-1} + \frac{\Delta t}{\rho}(\frac{\partial}{\partial x}\sigma_{xx}^{n-1/2} + \frac{\partial}{\partial z}\sigma_{xz}^{n-1/2} + s_1^{n-1}) \quad (6)$$

$$V_z^n = V_z^{n-1} + \frac{\Delta t}{\rho}(\frac{\partial}{\partial x}\sigma_{xz}^{n-1/2} + \frac{\partial}{\partial z}\sigma_{zz}^{n-1/2} + s_2^{n-1}) \quad (7)$$

$$\sigma_{xx}^{n+1/2} = \sigma_{xx}^{n-1/2} + \Delta t[(\lambda + 2\mu)\frac{\partial}{\partial x}V_x^n + \lambda\frac{\partial}{\partial z}V_z^n + s_3^{n-1/2}] \quad (8)$$

$$\sigma_{zz}^{n+1/2} = \sigma_{zz}^{n-1/2} + \Delta t[(\lambda + 2\mu)\frac{\partial}{\partial z}V_z^n + \lambda\frac{\partial}{\partial x}V_x^n + s_4^{n-1/2}] \quad (9)$$

$$\sigma_{xz}^{n+1/2} = \sigma_{xz}^{n-1/2} + \Delta t[\mu\frac{\partial}{\partial x}V_z^n + \mu\frac{\partial}{\partial z}V_x^n + s_5^{n-1/2}], \quad (10)$$

where $\mathbf{n}$ is the discretized time interval and $\mathbf{\Delta t}$ is the time step. For simplicity, I supress the spatial dependencies of the vectors.

The time staggering in this approach uses an alternating solution to the velocity and stress wave fields, which is concise and easy to implement. However, describing the set of equations as a single recursive relation becomes more difficult. Here, I choose two possible solutions to this problem. In the first method, which I call recursive operator by backward substitution, the stress equations are re-injected into the velocity equations, so that a single recursive equation arises. The second method, which I refer as recursive operator by time step refinement, maintains the staggered-time structure, but redefines the time stepping of each wave field so that both velocities and stresses can be defined at every time step.

Recursive operator by backward substitution

My first step in this approach is to write the velocities and stresses as two separate data vectors,

$$\mathbf{d_1^n} = \begin{pmatrix} V_x^n \\ V_z^n \end{pmatrix} \qquad \mathbf{d_2^n} = \begin{pmatrix} \sigma_{xx}^{n-\frac{1}{2}} \\ \sigma_{zz}^{n-\frac{1}{2}} \\ \sigma_{xz}^{n-\frac{1}{2}} \end{pmatrix}. \quad (11)$$

I also define a few operators to make the representation more compact,

$$A = \tfrac{\Delta t}{\rho}\tfrac{\partial}{\partial x} \qquad B = \tfrac{\Delta t}{\rho}\tfrac{\partial}{\partial z} \qquad C = \tfrac{\Delta t}{\rho}$$

$$D = \Delta t(\lambda + 2\mu)\tfrac{\partial}{\partial x} \qquad E = \Delta t\lambda\tfrac{\partial}{\partial z} \qquad F = \Delta t(\lambda + 2\mu)\tfrac{\partial}{\partial z}$$

$$G = \Delta t\lambda\tfrac{\partial}{\partial x} \qquad H = \Delta t\mu\tfrac{\partial}{\partial x} \qquad J = \Delta t\mu\tfrac{\partial}{\partial z}\ .$$

I can now write my elastic equations as two matrix operations,

$$\mathbf{d_1^n} = \mathbf{d_1^{n-1}} + \begin{pmatrix} A & 0 & B \\ 0 & B & A \end{pmatrix} \mathbf{d_2^n} + C \begin{pmatrix} s_1^{n-1} \\ s_2^{n-1} \end{pmatrix} \quad (12)$$

$$\mathbf{d_2^n} = \mathbf{d_2^{n-1}} + \begin{pmatrix} D & E \\ F & G \\ H & J \end{pmatrix} \mathbf{d_1^{n-1}} + \Delta t \begin{pmatrix} s_3^{n-\frac{1}{2}} \\ s_4^{n-\frac{1}{2}} \\ s_5^{n-\frac{1}{2}} \end{pmatrix}. \tag{13}$$

Substituting equation 13 into 12 yields,

$$\mathbf{d_1^n} = \mathbf{d_1^{n-1}} + \begin{pmatrix} A & 0 & B \\ 0 & B & A \end{pmatrix} \mathbf{d_2^{n-1}} + \begin{pmatrix} A & 0 & B \\ 0 & B & A \end{pmatrix} \begin{pmatrix} D & E \\ F & G \\ H & J \end{pmatrix} \mathbf{d_1^{n-1}} +$$
$$C \begin{pmatrix} s_1^{n-1} \\ s_2^{n-1} \end{pmatrix} + \Delta t \begin{pmatrix} A & 0 & B \\ 0 & B & A \end{pmatrix} \begin{pmatrix} s_3^{n-\frac{1}{2}} \\ s_4^{n-\frac{1}{2}} \\ s_5^{n-\frac{1}{2}} \end{pmatrix}. \tag{14}$$

I can now define a single vector $\mathbf{d^n}$

$$\mathbf{d^n} = \begin{pmatrix} V_x^n \\ V_z^n \\ \sigma_{xx}^{n-\frac{1}{2}} \\ \sigma_{zz}^{n-\frac{1}{2}} \\ \sigma_{xz}^{n-\frac{1}{2}} \end{pmatrix}, \tag{15}$$

which combines both subsets $\mathbf{d_1^n}$ and $\mathbf{d_2^n}$ into a single generalized recursive relation,

$$\mathbf{d^n} = \begin{pmatrix} I + AD + BH & AE + BJ & A & 0 & B \\ BF + AH & I + BG + AJ & 0 & B & A \\ D & E & I & 0 & 0 \\ F & G & 0 & I & 0 \\ H & J & 0 & 0 & I \end{pmatrix} \mathbf{d^{n-1}} + \begin{pmatrix} C & 0 & A & 0 & B \\ 0 & C & 0 & B & A \\ 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & \Delta t \end{pmatrix} \mathbf{S^{n-1}}, \tag{16}$$

where $\mathbf{I}$ is the identity operator and $\mathbf{S^n}$ is the source vector at time $\mathbf{n}$. For compactness, I define the matrices in 16 as $\mathbf{X}$ and $\mathbf{Y}$, so that the forward recursive relation can be written simply as

$$\mathbf{d^n} = \mathbf{X}\mathbf{d^{n-1}} + \mathbf{Y}\mathbf{S^{n-1}}. \tag{17}$$

While equation 17 correctly describes the forward elastic wave propagation, the adjoint of this equation cannot be taken directly since operators $\mathbf{X}$ and $\mathbf{Y}$ don't commute. The last step is to re-define the source term. The vector $\mathbf{q^n}$ is defined as the source term $\mathbf{S^n}$ after applying the operator $\mathbf{Y}$. Equation 18 is the final forward recursive operator

$$\mathbf{d^n} = \mathbf{X}\mathbf{d^{n-1}} + \mathbf{q^{n-1}}, \tag{18}$$

whose adjoint is described by

$$\mathbf{q^n} = \mathbf{X}'\mathbf{q^{n+1}} + \mathbf{d^{n+1}}, \tag{19}$$

where $\mathbf{X}'$ is the adjoint of $\mathbf{X}$.

Recursive operator by time step refinement

In this approach, instead of re-injecting the stress fields into the velocities to get them at integer time steps, I take the opposite approach. In other words, this method tries to obtain all wavefields at both integer and half-integer time steps, effectively refining the time stepping in the recursive relation.

Again, I start by definning a data vector $\mathbf{D^n}$, which represents the velocities and stresses at time $\mathbf{n}$. Notice that, unlike in the previous case, the stresses are not shifted in time in respect to the velocities.

$$\mathbf{d^n} = \begin{pmatrix} V_x^n \\ V_z^n \\ \sigma_{xx}^n \\ \sigma_{zz}^n \\ \sigma_{xz}^n \end{pmatrix}. \tag{20}$$

Next, I define a set of operator matrices that represent the original time staggered equations,

$$\mathbf{d^n} = \begin{pmatrix} 0 & 0 & A & 0 & B \\ 0 & 0 & 0 & B & A \\ D & E & 0 & 0 & 0 \\ F & G & 0 & 0 & 0 \\ H & J & 0 & 0 & 0 \end{pmatrix} \mathbf{d^{n-\frac{1}{2}}} + \mathbf{I} \cdot \mathbf{d^{n-1}} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & \Delta t \end{pmatrix} \mathbf{S^{n-\frac{1}{2}}} +$$

$$\begin{pmatrix} C & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{S^{n-1}}, \tag{21}$$

where $\mathbf{I}$ is the identity matrix and operators $\mathbf{A}$ through $\mathbf{J}$ are the same as those defined in the previous section. The forward recursive relation can be represented as

$$\mathbf{d^n} = \mathbf{X d^{n-\frac{1}{2}}} + \mathbf{I} \cdot \mathbf{d^{n-1}} + \mathbf{Y_1 S^{n-\frac{1}{2}}} + \mathbf{Y_2 S^{n-1}}, \tag{22}$$

and its adjoint as

$$\mathbf{q^n} = \mathbf{X' q^{n+\frac{1}{2}}} + \mathbf{I} \cdot \mathbf{q^{n+1}} + \mathbf{Y_1' d^{n+\frac{1}{2}}} + \mathbf{Y_2' d^{n+1}}. \tag{23}$$

While the recursive relations described here and in the previous section are at the core of most forward and inverse methodologies, it is important to take into account other important effects that appear when trying to numerically solve the elastic wave equation. In the next section, I re-derive the two previous solutions, taking into account border and interpolation effects.

PERFECTLY MATCHED LAYER IMPLEMENTATION

A perfectly matched layer (PML) is a type of absorbing boundary condition used in numerical modeling to avoid reflections of propagating waves off the corners of a finite numerical problem. It is more efficient than decaying exponential methods (Cerjan et al., 1985), because it splits the wavefront into its directional components, applying an absorbing factor in each direction instead of a single absorption normal to the model boundary.

In order to apply PML to my problem, I start by splitting the previous set of equations into its spatial derivatives. Since this is a 2D problem, the original equations are split into their **x** and **z** derivatives, namely: V_x^x, V_x^z, V_z^x and V_z^z for the particle velocities and σ_{xx}^x, σ_{xx}^z, σ_{zz}^x, σ_{zz}^z, σ_{xz}^x and σ_{xz}^z for the stresses. Also, I use the subscripts i, j to represent the grid position to which the variables belong. I need to include this because I implement the equations using the staggered-grid approach, similarly to what was done in the time domain in the previous section. The decomposed set of 10 equations is

$$\frac{\partial}{\partial t}(V_x^x)_{i,j} = \frac{1}{\rho}[\frac{\partial}{\partial x}(\sigma_{xx})_{i+1/2,j}] + s_x^x \tag{24}$$

$$\frac{\partial}{\partial t}(V_x^z)_{i,j} = \frac{1}{\rho}[\frac{\partial}{\partial z}(\sigma_{xz})_{i,j+1/2}] + s_x^z \tag{25}$$

$$\frac{\partial}{\partial t}(V_z^x)_{i+1/2,j+1/2} = \frac{1}{\rho}[\frac{\partial}{\partial x}(\sigma_{xz})_{i,j+1/2}] + s_z^x \tag{26}$$

$$\frac{\partial}{\partial t}(V_z^z)_{i+1/2,j+1/2} = \frac{1}{\rho}[\frac{\partial}{\partial z}(\sigma_{zz})_{i+1/2,j}] + s_z^z \tag{27}$$

$$\frac{\partial}{\partial t}(\sigma_{xx}^x)_{i+1/2,j} = (\lambda + 2\mu)\frac{\partial}{\partial x}(V_x)_{i,j} + s_{xx}^x \tag{28}$$

$$\frac{\partial}{\partial t}(\sigma_{xx}^z)_{i+1/2,j} = \lambda\frac{\partial}{\partial z}(V_z)_{i+1/2,j+1/2} + s_{xx}^z \tag{29}$$

$$\frac{\partial}{\partial t}(\sigma_{zz}^x)_{i+1/2,j} = \lambda\frac{\partial}{\partial x}(V_x)_{i,j} + s_{zz}^x \tag{30}$$

$$\frac{\partial}{\partial t}(\sigma_{zz}^z)_{i+1/2,j} = (\lambda + 2\mu)\frac{\partial}{\partial z}(V_z)_{i+1/2,j+1/2} + s_{zz}^z \tag{31}$$

$$\frac{\partial}{\partial t}(\sigma_{xz}^x)_{i,j+1/2} = \mu\frac{\partial}{\partial x}(V_z)_{i+1/2,j+1/2} + s_{xz}^x \tag{32}$$

$$\frac{\partial}{\partial t}(\sigma_{xz}^z)_{i,j+1/2} = \mu\frac{\partial}{\partial z}(V_x)_{i,j} + s_{xz}^z, \tag{33}$$

where $V_x = V_x^x + V_x^z$, and so on. It is important to note that, following the method by Virieux (1986), the elastic properties of the model are defined at separate grid points. Therefore, they must be averaged to correspond to the equivalent values at the points being calculated. Here, I follow the method of harmonic averages described by Moczo et al. (2002).

The next step is to include the PML parameters. To do so, I follow the work of Collino and Tsogka (2001). Taking equation 24 as an example and representing the time derivative as a second order finite difference approximation, the elastic equation with PML becomes

$$\frac{(V_x^x)_{i,j}^{n+1} - (V_x^x)_{i,j}^{n}}{\Delta t} + d_i^x \frac{(V_x^x)_{i,j}^{n+1} + (V_x^x)_{i,j}^{n}}{2} = \frac{1}{\rho}[\frac{\partial}{\partial x}(\sigma_{xx})_{i+1/2,j}] + s_x^x, \quad (34)$$

where the superscript n refers to the discrete time in the finite differences method and d_i^x is given by

$$d_i^x = log(\frac{1}{R})(\frac{3V_P}{2\delta})(\frac{x}{\delta})^2, \quad (35)$$

where the index i refers to the grid position and the x represents the direction of absorption. R is an arbitrary parameter that is associated with the desired reflectivity at the outer boundary and δ is the boundary thickness. Typical values for these parameters are 0.001 and 10 grid points, respectively.

Rearranging the terms in equation 34, I get

$$(V_x^x)_{i,j}^{n+1} = (1 + \frac{\Delta t}{2} d_i^x)^{-1}[(1 - \frac{\Delta t}{2} d_i^x)(V_x^x)_{i,j}^{n} + \frac{\Delta t}{\rho}\frac{\partial}{\partial x}(\sigma_{xx})_{i+1/2,j}^{n+1/2}], \quad (36)$$

and similarly for the z component of the V_x velocity I get

$$(V_x^z)_{i,j}^{n+1} = (1 + \frac{\Delta t}{2} d_j^z)^{-1}[(1 - \frac{\Delta t}{2} d_j^z)(V_x^z)_{i,j}^{n} + \frac{\Delta t}{\rho}\frac{\partial}{\partial z}(\sigma_{xz})_{i,j+1/2}^{n+1/2}]. \quad (37)$$

Re-writing the set of equations as a chain of operators, I get

$$(V_x^x)_{i,j}^{n} = KL(V_x^x)_{i,j}^{n-1} + KA(\sigma_{xx})_{i+1/2,j}^{n-1/2} + KC(s_x^x)^{n-1} \quad (38)$$

$$(V_x^z)_{i,j}^{n} = MN(V_x^z)_{i,j}^{n-1} + MB(\sigma_{xz})_{i,j+1/2}^{n-1/2} + MC(s_x^z)^{n-1} \quad (39)$$

$$(V_z^x)_{i+1/2,j+1/2}^{n} = OP(V_x^x)_{i,j}^{n-1} + OA(\sigma_{xz})_{i,j+1/2}^{n-1/2} + OC(s_z^x)^{n-1} \quad (40)$$

$$(V_z^z)_{i+1/2,j+1/2}^{n} = QR(V_x^z)_{i,j}^{n-1} + QB(\sigma_{xx})_{i+1/2,j}^{n-1/2} + QC(s_z^z)^{n-1} \quad (41)$$

$$(\sigma_{xx}^x)_{i+1/2,j}^{n+1/2} = OP(\sigma_{xx}^x)_{i+1/2,j}^{n-1/2} + OD(V_x)_{i,j}^{n} + O\Delta t(s_{xx}^x)^{n-1/2} \quad (42)$$

$$(\sigma_{xx}^z)_{i+1/2,j}^{n+1/2} = QR(\sigma_{xx}^z)_{i+1/2,j}^{n-1/2} + QE(V_z)_{i+1/2,j+1/2}^{n} + Q\Delta t(s_{xx}^z)^{n-1/2} \quad (43)$$

$$(\sigma_{zz}^x)_{i+1/2,j}^{n+1/2} = OP(\sigma_{xx}^x)_{i+1/2,j}^{n-1/2} + OF(V_x)_{i,j}^{n} + O\Delta t(s_{zz}^x)^{n-1/2} \quad (44)$$

$$(\sigma_{zz}^z)_{i+1/2,j}^{n+1/2} = QR(\sigma_{xx}^z)_{i+1/2,j}^{n-1/2} + QG(V_z)_{i+1/2,j+1/2}^{n} + Q\Delta t(s_{zz}^z)^{n-1/2} \quad (45)$$

$$(\sigma_{xz}^x)_{i,j+1/2}^{n+1/2} = KL(\sigma_{xz}^x)_{i,j+1/2}^{n-1/2} + KH(V_x)_{i,j}^{n} + K\Delta t(s_{xz}^x)^{n-1/2} \quad (46)$$

$$(\sigma_{xz}^z)_{i,j+1/2}^{n+1/2} = MN(\sigma_{xz}^z)_{i,j+1/2}^{n-1/2} + MJ(V_z)_{i+1/2,j+1/2}^{n} + M\Delta t(s_{xz}^z)^{n-1/2}, \quad (47)$$

where the operators **A** through **J** have been defined previously and the PML operators are given by

$$
\begin{array}{lll}
K = (1 + \frac{\Delta t}{2} d_i^x)^{-1} & L = (1 - \frac{\Delta t}{2} d_i^x) & M = (1 + \frac{\Delta t}{2} d_j^z)^{-1} \\
N = (1 - \frac{\Delta t}{2} d_j^z) & O = (1 + \frac{\Delta t}{2} d_{i+1/2}^x)^{-1} & P = (1 - \frac{\Delta t}{2} d_{i+1/2}^x) \\
Q = (1 + \frac{\Delta t}{2} d_{j+1/2}^z)^{-1} & R = (1 - \frac{\Delta t}{2} d_{j+1/2}^z)\ . &
\end{array}
$$

Now that I have represented each equation as an idependent recursive relation, I need to describe the full data set as one recursive relation. Again, I will derive the necessary equations for both methods described earlier.

PML for the recursive operator by backward substitution

Similarly to the case without PML, I start by writing my wave fields as two data vectors,

$$
\mathbf{d_1^n} = \begin{pmatrix} (V_x^x)^n \\ (V_x^z)^n \\ (V_z^x)^n \\ (V_z^z)^n \end{pmatrix} \qquad \mathbf{d_2^n} = \begin{pmatrix} (\sigma_{xx}^x)^{n-1/2} \\ (\sigma_{xx}^z)^{n-1/2} \\ (\sigma_{zz}^x)^{n-1/2} \\ (\sigma_{zz}^z)^{n-1/2} \\ (\sigma_{xz}^x)^{n-1/2} \\ (\sigma_{xz}^z)^{n-1/2}. \end{pmatrix} \tag{48}
$$

The recursive relations then become

$$
\mathbf{d_1^n} = \begin{pmatrix} KL & 0 & 0 & 0 \\ 0 & MN & 0 & 0 \\ 0 & 0 & OP & 0 \\ 0 & 0 & 0 & QR \end{pmatrix} \mathbf{d_1^{n-1}} + \begin{pmatrix} KA & KA & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & MB & MB \\ 0 & 0 & 0 & 0 & OA & OA \\ 0 & 0 & QB & QB & 0 & 0 \end{pmatrix} \mathbf{d_2^n} +
$$
$$
\begin{pmatrix} KC & 0 & 0 & 0 \\ 0 & MC & 0 & 0 \\ 0 & 0 & OC & 0 \\ 0 & 0 & 0 & QC \end{pmatrix} \mathbf{S_1^{n-1}} \tag{49}
$$

$$\mathbf{d_2^n} = \begin{pmatrix} OD & OD & 0 & 0 \\ 0 & 0 & QE & QE \\ OF & OF & 0 & 0 \\ 0 & 0 & QG & QG \\ KH & KH & 0 & 0 \\ 0 & 0 & MJ & MJ \end{pmatrix} \mathbf{d_1^{n-1}} + \begin{pmatrix} OP & 0 & 0 & 0 & 0 & 0 \\ 0 & QR & 0 & 0 & 0 & 0 \\ 0 & 0 & OP & 0 & 0 & 0 \\ 0 & 0 & 0 & QR & 0 & 0 \\ 0 & 0 & 0 & 0 & KL & 0 \\ 0 & 0 & 0 & 0 & 0 & MN \end{pmatrix} \mathbf{d_2^{n-1}} +$$

$$\begin{pmatrix} O\Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & Q\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & O\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & Q\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & K\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & M\Delta t \end{pmatrix} \mathbf{S_2^{n-1}}. \tag{50}$$

Substituting 13 into 12, I get

$$\mathbf{d_1^n} = \begin{pmatrix} KL + KAOD & KAOD & KAQE & KAQE \\ MBKH & MN + MBKH & MBMJ & MBMJ \\ OAKH & OAKH & OP + OAMJ & OAMJ \\ QBOF & QBOF & QBQG & QR + QBQG \end{pmatrix} \mathbf{d_1^{n-1}} +$$

$$\begin{pmatrix} KAOP & KAQR & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & MBKL & MBMN \\ 0 & 0 & 0 & 0 & OAKL & OAMN \\ 0 & 0 & QBOP & QBQR & 0 & 0 \end{pmatrix} \mathbf{d_2^{n-1}} +$$

$$\begin{pmatrix} KC & 0 & 0 & 0 \\ 0 & MC & 0 & 0 \\ 0 & 0 & OC & 0 \\ 0 & 0 & 0 & QC \end{pmatrix} \mathbf{S_1^{n-1}} +$$

$$\begin{pmatrix} KAO\Delta t & KAQ\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & MBK\Delta t & MBM\Delta t \\ 0 & 0 & 0 & 0 & OAK\Delta t & OAM\Delta t \\ 0 & 0 & QBO\Delta t & QBQ\Delta t & 0 & 0 \end{pmatrix} \mathbf{S_2^{n-1}}. \tag{51}$$

Finally, I can construct the recursive operator **W**

$$
\mathbf{d^n} = \begin{pmatrix} d_1^n \\ d_2^n \end{pmatrix} = \begin{pmatrix}
KL+KAOD & KAOD & KAQE & KAQE & KAOP & KAQR & 0 & 0 & 0 & 0 \\
MBKH & MN+MBKH & MBMJ & MBMJ & 0 & 0 & 0 & 0 & MBKL & MBMN \\
OAKH & OAKH & OP+OAMJ & OAMJ & 0 & 0 & 0 & 0 & OAKL & OAMN \\
QBOF & QBOF & QBQG & QR+QBQG & 0 & 0 & QBOP & QBQR & 0 & 0 \\
OD & OD & 0 & 0 & OP & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & QE & QE & 0 & QR & 0 & 0 & 0 & 0 \\
OF & OF & 0 & 0 & 0 & 0 & OP & 0 & 0 & 0 \\
0 & 0 & QG & QG & 0 & 0 & 0 & QR & 0 & 0 \\
KH & KH & 0 & 0 & 0 & 0 & 0 & 0 & KL & 0 \\
0 & 0 & MJ & MJ & 0 & 0 & 0 & 0 & 0 & MN
\end{pmatrix} \mathbf{d^{n-1}} +
$$

$$
\begin{pmatrix}
KC & 0 & 0 & 0 & KAO\Delta t & KAQ\Delta t & 0 & 0 & 0 & 0 \\
0 & MC & 0 & 0 & 0 & 0 & 0 & 0 & MBK\Delta t & MBM\Delta t \\
0 & 0 & OC & 0 & 0 & 0 & 0 & 0 & OAK\Delta t & OAM\Delta t \\
0 & 0 & 0 & QC & 0 & 0 & QBO\Delta t & QBQ\Delta t & 0 & 0 \\
0 & 0 & 0 & 0 & O\Delta t & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & Q\Delta t & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & O\Delta t & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & Q\Delta t & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K\Delta t & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M\Delta t
\end{pmatrix} \begin{pmatrix} S_1^n \\ S_2^n \end{pmatrix} \tag{52}
$$

PML for the recursive operator by time step refinement

Lastly, the PML implementation in the case of the refined time stepping method is

$$
\begin{pmatrix} (V_x^x)^n \\ (V_x^z)^n \\ (V_z^x)^n \\ (V_z^z)^n \\ (\sigma_{xx}^x)^n \\ (\sigma_{xx}^z)^n \\ (\sigma_{zz}^x)^n \\ (\sigma_{zz}^z)^n \\ (\sigma_{xz}^x)^n \\ (\sigma_{xz}^z)^n \end{pmatrix} =
\begin{pmatrix}
0 & 0 & 0 & 0 & KA & KA & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & MB & MB \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & OA & OA \\
0 & 0 & 0 & 0 & 0 & 0 & QB & QB & 0 & 0 \\
OD & OD & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & QE & QE & 0 & 0 & 0 & 0 & 0 & 0 \\
OF & OF & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & QG & QG & 0 & 0 & 0 & 0 & 0 & 0 \\
KH & KH & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & MJ & MJ & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \mathbf{d^{n-\frac{1}{2}}} +
$$

$$
\begin{pmatrix}
KL & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & MN & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & OP & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & QR & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & OP & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & QR & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & OP & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & QR & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & KL & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & MN
\end{pmatrix} \mathbf{d^{n-1}} +
$$

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & O\Delta t & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & Q\Delta t & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & O\Delta t & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & Q\Delta t & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K\Delta t & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M\Delta t
\end{pmatrix} \mathbf{S^{n-\frac{1}{2}}} +
$$

$$
\begin{pmatrix}
KC & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & MC & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & OC & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & QC & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \mathbf{S^{n-1}}, \tag{53}
$$

which can be compactly described by the recursive relation

$$\mathbf{d^n} = \mathbf{X d^{n-\frac{1}{2}}} + \mathbf{Z d^{n-1}} + \mathbf{Y_1 S^{n-\frac{1}{2}}} + \mathbf{Y_2 S^{n-1}}. \tag{54}$$

NONLINEAR MODELING

Theoretically, both recursive operators defined in the previous sections could be used to construct a linear operator with respect to the source function. However, the first method yields a more cumbersome system, which would only grow in complexity as more elements are added. Therefore, in the next sections I focus only on the recursive operator by time stepping refinement. I start with a general linear operator summarized by

$$\mathbf{d} = \mathbf{F s}, \tag{55}$$

where $\mathbf{d}$ is the output data, $\mathbf{F}$ is the forward modeling operator and $\mathbf{s}$ is the source.

Next, I include the necessary interpolation and padding operators, similarl to the work in Almomin (2013),

$$\mathbf{d} = \mathbf{K_r' L_r' W Y L_s K_s s}, \tag{56}$$

where $\mathbf{K_r}$ and $\mathbf{L_s}$ are the spatial padding and time interpolation operator, respectively. $\mathbf{W}$ is the recursive operator described by equation 54. Similarly, the adjoint modeling operator is given by

$$\mathbf{s} = \mathbf{K_s' L_s' Y' W' L_r K_r d}, \tag{57}$$

where $\mathbf{W'}$ is the recursive relation described by

$$\mathbf{q^n} = \mathbf{X' q^{n+\frac{1}{2}}} + \mathbf{Z' q^{n+1}} + \mathbf{d^{n+1}}, \tag{58}$$

where $\mathbf{q^n}$ is the scaled source

$$\mathbf{q^n} = \mathbf{Y_1 s^{n+\frac{1}{2}}} + \mathbf{Y_2 s^n}. \tag{59}$$

The adjoint of matrix $\mathbf{X}$ in the adjoint recursive relation $\mathbf{W'}$ is

$$\mathbf{X'} = \begin{pmatrix} 0 & 0 & 0 & 0 & D'O' & 0 & F'O' & 0 & H'K' & 0 \\ 0 & 0 & 0 & 0 & D'O' & 0 & F'O' & 0 & H'K' & 0 \\ 0 & 0 & 0 & 0 & 0 & E'Q' & 0 & G'Q' & 0 & J'M' \\ 0 & 0 & 0 & 0 & 0 & E'Q' & 0 & G'Q' & 0 & J'M' \\ A'K' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A'K' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B'Q' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B'Q' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B'M' & A'O' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B'M' & A'O' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{60}$$

and the adjoints of matrices $\mathbf{Z}$, $\mathbf{Y_1}$ and $\mathbf{Y_2}$ are

$$\mathbf{Z}' = \begin{pmatrix} L'K' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N'M' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P'O' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & R'Q' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P'O' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R'Q' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & P'O' & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R'Q' & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L'K' & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & N'M' \end{pmatrix}, \quad (61)$$

$$\mathbf{Y_1}' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta t O' & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta t Q' & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Delta t O' & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t Q' & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t K' & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t M' \end{pmatrix}, \quad (62)$$

$$\mathbf{Y_2}' = \begin{pmatrix} C'K' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C'M' & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C'O' & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C'Q' & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (63)$$

The individual adjoint operators are

$$A' = \frac{\partial}{\partial x}\frac{\Delta t}{\rho} \qquad B' = \frac{\partial}{\partial z}\frac{\Delta t}{\rho} \qquad C' = \frac{\Delta t}{\rho}$$

$$D' = \frac{\partial}{\partial x}\Delta t(\lambda + 2\mu) \qquad E' = \frac{\partial}{\partial z}\Delta t\lambda \qquad F' = \frac{\partial}{\partial z}\Delta t(\lambda + 2\mu)$$

$$G' = \frac{\partial}{\partial x}\Delta t\lambda \qquad H' = \frac{\partial}{\partial x}\Delta t\mu \qquad J' = \frac{\partial}{\partial z}\Delta t\mu$$

$$K' = (1 + \frac{\Delta t}{2}d_i^x)^{-1} \qquad L' = (1 - \frac{\Delta t}{2}d_i^x) \qquad M' = (1 + \frac{\Delta t}{2}d_j^z)^{-1}$$

$$N' = (1 - \frac{\Delta t}{2}d_j^z) \qquad O' = (1 + \frac{\Delta t}{2}d_{i+1/2}^x)^{-1} \qquad P' = (1 - \frac{\Delta t}{2}d_{i+1/2}^x)$$

$$Q' = (1 + \frac{\Delta t}{2}d_{j+1/2}^z)^{-1} \qquad R' = (1 - \frac{\Delta t}{2}d_{j+1/2}^z)\ .$$

I apply the dot product test to validate that the adjoint of the recursive relation is correct. I test the case where the PML coefficient is equal to one (rigid boundary condition) and calculate the forward and adjoints for 5,000 time steps. For a 2D grid with 200 by 200 samples, the relative error in the dot product test is on the order of 10^{-12}. I run the tests in double precision, with random inputs for the forward and adjoint solutions. This method follows the one described in Claerbout (2010).

BORN OPERATOR

The Born method is a linear approximation to the non-linear wave propagation problem. Essentially, it aims to describe the non-linear problem in terms of a perturbation problem, where the solution is a combination of a non-scattering background wavefield and a scattering term. Such an approximation is valid for small perturbations, where secondary scattering events are small compared to the background and first order scattering terms. To derive the Born approximation of the elastic wave equation, I start by writing the initial set of equations for a perturbed properties model,

$$(\rho + \Delta\rho)\frac{\partial}{\partial t}(V_x + \Delta V_x) - [\frac{\partial}{\partial x}(\sigma_{xx} + \Delta\sigma_{xx}) + \frac{\partial}{\partial z}(\sigma_{xz} + \Delta\sigma_{xz})] = s_1 \tag{64}$$

$$(\rho + \Delta\rho)\frac{\partial}{\partial t}(V_z + \Delta V_z) - [\frac{\partial}{\partial x}(\sigma_{xz} + \Delta\sigma_{xz}) + \frac{\partial}{\partial z}(\sigma_{zz} + \Delta\sigma_{zz})] = s_2 \tag{65}$$

$$\frac{\partial}{\partial t}(\sigma_{xx} + \Delta\sigma_{xx}) = [(\lambda + \Delta\lambda) + 2(\mu + \Delta\mu)]\frac{\partial}{\partial x}(V_x + \Delta V_x) + (\lambda + \Delta\lambda)\frac{\partial}{\partial z}(V_z + \Delta V_z) + s_3 \tag{66}$$

$$\frac{\partial}{\partial t}(\sigma_{zz} + \Delta\sigma_{zz}) = [(\lambda + \Delta\lambda) + 2(\mu + \Delta\mu)]\frac{\partial}{\partial z}(V_z + \Delta V_z) + (\lambda + \Delta\lambda)\frac{\partial}{\partial x}(V_x + \Delta V_x) + s_4 \tag{67}$$

$$\frac{\partial}{\partial t}(\sigma_{xz} + \Delta\sigma_{xz}) = (\mu + \Delta\mu)(\frac{\partial}{\partial x}(V_z + \Delta V_z) + \frac{\partial}{\partial z}(V_x + \Delta V_x)) + s_5. \tag{68}$$

Rearranging equation 64, I get

$$\rho\frac{\partial}{\partial t}V_x - [\frac{\partial}{\partial x}\sigma_{xx} + \frac{\partial}{\partial z}\sigma_{xz}] + \rho\frac{\partial}{\partial t}\Delta V_x + \Delta\rho\frac{\partial}{\partial t}V_x + \Delta\rho\frac{\partial}{\partial t}\Delta V_x - [\frac{\partial}{\partial x}\Delta\sigma_{xx} + \frac{\partial}{\partial z}\Delta\sigma_{xz}] = s_1, \quad (69)$$

which is essentially equation 1 plus perturbation terms. If I ignore higher order perturbation terms and substract the original wave equation, equation 69 gives me the Born scattering term

$$\rho\frac{\partial}{\partial t}\Delta V_x - (\frac{\partial}{\partial x}\Delta\sigma_{xx} + \frac{\partial}{\partial z}\Delta\sigma_{xz}] = -\Delta\rho\frac{\partial}{\partial t}V_x. \quad (70)$$

Similarly, for the other equations I get

$$\rho\frac{\partial}{\partial t}\Delta V_z - (\frac{\partial}{\partial x}\Delta\sigma_{xz} + \frac{\partial}{\partial z}\Delta\sigma_{zz}] = -\Delta\rho\frac{\partial}{\partial t}V_z \quad (71)$$

$$\frac{\partial}{\partial t}\Delta\sigma_{xx} - [(\lambda+2\mu)\frac{\partial}{\partial x}\Delta V_x + \lambda\frac{\partial}{\partial z}\Delta V_z] = (\Delta\lambda + 2\Delta\mu)\frac{\partial}{\partial x}V_x + \Delta\lambda\frac{\partial}{\partial z}V_z \quad (72)$$

$$\frac{\partial}{\partial t}\Delta\sigma_{zz} - [(\lambda+2\mu)\frac{\partial}{\partial z}\Delta V_z + \lambda\frac{\partial}{\partial x}\Delta V_x] = (\Delta\lambda + 2\Delta\mu)\frac{\partial}{\partial z}V_z + \Delta\lambda\frac{\partial}{\partial x}V_x \quad (73)$$

$$\frac{\partial}{\partial t}\Delta\sigma_{xz} - [\mu(\frac{\partial}{\partial x}\Delta V_z + \frac{\partial}{\partial z}\Delta V_x)] = \Delta\mu(\frac{\partial}{\partial x}V_z + \frac{\partial}{\partial z}V_x). \quad (74)$$

The numerical implementation of the forward Born modeling can be broken into two steps. First, a virtual source is constructed by the forward propagation of the elastic wave fields in a background (unperturbed) properties model. I describe this background model by $\rho_{\mathbf{0}}$, $\lambda_{\mathbf{0}}$ and $\mu_{\mathbf{0}}$. This background wave field is constructed similarly to equation 56, but without the need to truncate the output to the data spatial sampling and with its time sampling equal to that of the source,

$$\mathbf{d_0} = \mathbf{L_s'WYL_sK_sf}. \quad (75)$$

After generating the background wave field, I use it as a virtual source to generate a new data set. This new data set, which I call the scattered data ($\mathbf{\Delta d}$), is given by

$$\mathbf{\Delta d} = \mathbf{K_r'L_r'WYL_sY\Theta d_0 r}, \quad (76)$$

where $\mathbf{\Theta}$ is a derivative operator that needs to be applied to the background wave field $\mathbf{d_0}$ and $\mathbf{r}$ is the perturbed properties model. The matrices that represent these operators are

$$\boldsymbol{\Theta} = \begin{pmatrix} \frac{\partial}{\partial t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial t} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial t} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{77}$$

$$\mathbf{r} = \begin{pmatrix} -\Delta\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\Delta\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\Delta\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\Delta\rho & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta\lambda + 2\Delta\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \Delta\lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Delta\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta\lambda + 2\Delta\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta\mu \end{pmatrix}. \tag{78}$$

A few observations should be made about equations 77 and 78. First, the operator $\boldsymbol{\Theta}$ has non-zero terms only for the particle velocity components. This means that the virtual sources for velocities and stresses, at least for this particular formulation, require only the velocity components from the background wavefield, which are then scaled correctly by the appropriate derivative. Second, $\mathbf{r}$ is a diagonal operator that has terms that depend not only on the perturbed density and each Lamé parameter individually, but also on a combination of the two Lamé parameters. If used in an iterative inversion method, this mixed term could lead to strong crosstalk between the gradient updates, in addition to the expected crosstalk from the equations coupling. However, this topic still needs further study.

RESULTS

I implement the Born forward modeling using a time domain, finite difference algorithm. The algorithm uses a second order approximation to the time derivatives and a 10^{th} order operator for the spatial derivatives, as described in Alves and Biondi (2014).

The model is a constant background with a gaussian perturbation in all model properties (λ, μ and ρ), as can be seen in figure 1(a).

Figure 2(a) shows a snapshot of the pressure wave field for the nonlinear modeling. In other words, this figure is generated by propagating the wave field in the correct gaussian model. For the Born modeling, I smooth the inital model (1(b)) and subtract it from the correct model (1(c)). This generates two wave fields: a background propagated one, shown in figure 2(b) and a scattered wavefield, shown in figure 2(c). All wave fields are scaled equally in order to compare the Born linear approximation and the non-linear modeling.

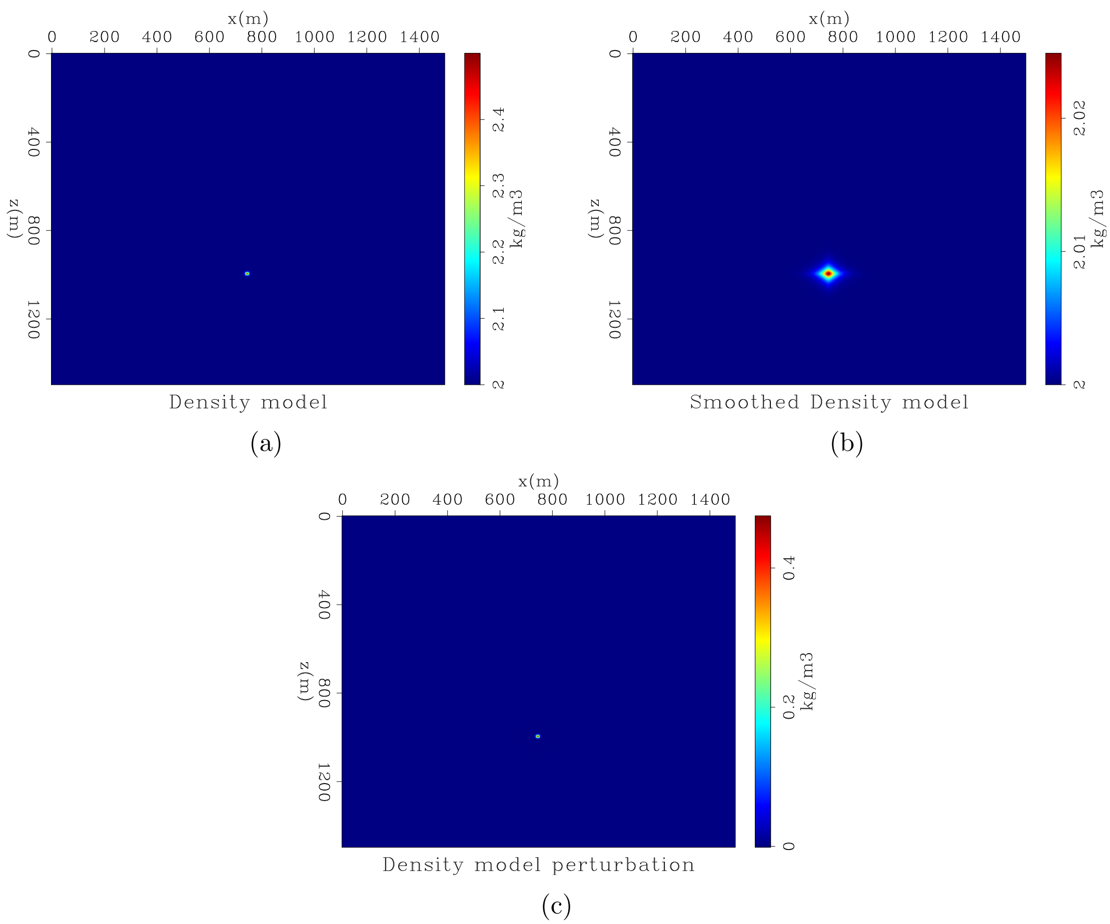

Figure 1: (a) True density model with a gaussian perturbation, (b) smoothed density model and (c) model difference between true and smoothed model. The other parameters display similar perturbations. [**ER**]

As can be seen from the examples, the Born modeling approximates the non-linear propagation not only qualitatively, but also quantitatively. It is important to point out that, while it is more customary to work with the pressure and shear velocities as model parameters for imaging, the formulation presented here is solved in terms of Lamè parameters.

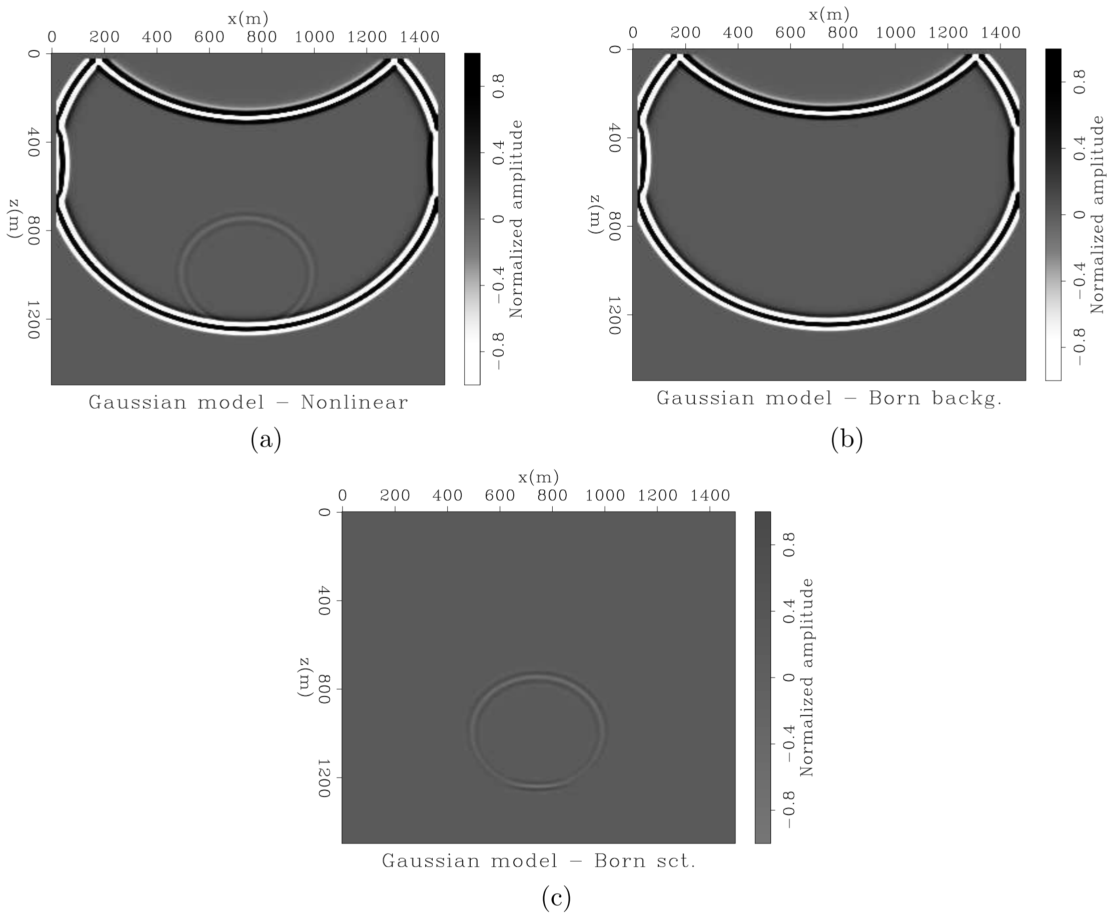

Figure 2: Snapshots of the pressure wavefield for (a) the non-linear modeling, (b) background modeling with the Born approximation and (c) scattering modeling with the Born approximation. [**ER**]

CONCLUSIONS

The recursive operator by backward substitution might be computationally more efficient, due to the smaller number of time steps required, when compared to the recursive operator by time step refinement. However, the long chains of operators required in the former make its implementation more difficult.

The recursive operator by time refinement is simpler to implement and is shown to be stable both in its forward and adjoint formulations.

The PML implementation is efficient, but requires twice as many wavefield computations due to the required component splitting. When applying the proposed scheme to problems where computational efficiency is required, the PML method should be limited to the absorbing boundaries and not to the whole model domain.

The formulation presented here is the first step in describing elastic inversion schemes in the framework of the adjoint state method. While this is not the only possible approach to such schemes, this formulation is closer to those developed in the Stanford Exploration Project for acoustic methods. Such similarity allows for an easier comparison between acoustic and elastic methods.

REFERENCES

Aki, K. and P. G. Richards, 1980, Quantitative seismology: Theory and methods, 1: I: WH Freeman and Co.

Almomin, A., 2013, Accurate implmentation of two-way wave-equation operators: Stanford Exploration Project Report, **149**, 279–286.

Alves, G. and B. Biondi, 2014, High-order elastic finite-difference modeling: Stanford Exploration Project Report, **152**, 327–340.

Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: Geophysics, **50**, 705–708.

Claerbout, J. F., 2010, Basic earth imaging: Retrieved from http://sepwww.stanford.edu/sep/prof/bei11.2010.pdf.

Collino, F. and C. Tsogka, 2001, Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media: Geophysics, **66**, 294–307.

Levander, A. R., 1988, Fourth-order finite-difference P-SV seismograms: Geophysics, **53**, 1425–1436.

Moczo, P., J. Kristek, V. Vavryčuk, R. J. Archuleta, and L. Halada, 2002, 3D heterogeneous staggered-grid finite-difference modeling of seismic motion with volume harmonic and arithmetic averaging of elastic moduli and densities: Bulletin of the Seismological Society of America, **92**, 3042–3066.

Virieux, J., 1986, P-SV wave propagation in heterogeneous media: Velocity-stress finite-difference method: Geophysics, **51**, 889–901.