Enhanced neural network sampling for two-stage Markov chain Monte Carlo seismic inversion

Georgia K. Stuart, Susan E. Minkoff, and Felipe Pereira, The University of Texas at Dallas*

SUMMARY

Monte Carlo techniques for seismic inversion are effective at producing high-quality uncertainty information about model parameters while reducing the number of assumptions required by non-stochastic full waveform inversion algorithms. However, Monte Carlo techniques are computationally expensive since they require repeated solution of the wave equation for each proposed velocity model. One approach to reduce the computational expense is two-stage Markov chain Monte Carlo (MCMC), wherein a computationally inexpensive filter is used to quickly reject unacceptable models. This filter reduces the number of models that must be tested using the full fine-grid forward solver. Often, coarse-grid formulations of the forward problems are used in the filter stage. We instead use a neural network to replace the expensive computation of the residual directly. In addition, we train the neural network as part of the MCMC process, bypassing the need to generate separate training data outside of the MCMC process. We find that the neural network two-stage MCMC (NNMCMC) substantially reduces the time-per-trial and time-per-rejection by 65% and 84%, respectively, when compared with the one-stage MCMC. Furthermore, NNMCMC increases the acceptance rate to 86% from 29% over the one-stage MCMC.

INTRODUCTION

Markov chain Monte Carlo (MCMC) methods are employed in seismic inversion to recover distributions of models that fit observed data. Monte Carlo techniques avoid assumptions of linearity between the model and the data (Sambridge and Mosegaard, 2002) and do not impose restrictions on the shape of the posterior distribution (i.e., uncertainty information) of the model (Mosegaard and Tarantola, 1995). However, MCMC techniques require testing tens of thousands to millions of models, and each model requires solution of the forward problem to produce simulated data. Due to the large number of samples needed to construct uncertainty information, even a moderately computationally expensive forward problem, such as the 2D acoustic wave equation used in this paper, results in a computationally infeasible algorithm (Higdon et al., 2011).

In traditional Metropolis-Hastings MCMC, 40% to 90% of the proposed models are rejected and not used to characterize the posterior distribution (Rosenthal, 2011). One way to reduce computational expense is to sample the posterior distribution more efficiently, thus requiring fewer solutions of the forward problem and increasing the acceptance ratio (Frangos et al., 2011). Two-stage MCMC reduces the number of models that must be tested on the full forward problem by using a computationally inexpensive filter to reject unacceptable models (Christen and Fox, 2005; Efendiev et al., 2006). Efendiev et al. (2006) and Ginting et al. (2011) use coarse-grid models as filters for subsurface flow problems. Stuart et al. (2016, in review 2019) apply the two-stage MCMC algorithm to seismic velocity inversion using operator upscaling as the coarse-grid filter (see Vdovina et al. (2005) and Korostyshevskaya and Minkoff (2006)).

In this work, we apply the two-stage MCMC algorithm to the full waveform inversion problem with a novel filter—a neural network. A neural network is a collection of interconnected nodes with the ability to learn complex relationships between input and output (Hastie et al., 2013; Strang, 2019). Röth and Tarantola (1994) pioneered the use of neural networks in seismic inversion by training a neural network to predict simple layered Earth velocity models from shot gathers. Lewis and Vigh (2017) used a convolutional neural network to predict the position of salt bodies in order to construct a prior velocity field for full waveform inversion. Araya-Polo et al. (2018) use a deep neural network to predict velocity fields with simple structures from raw seismic data.

The papers outlined above all use neural networks to directly predict velocity fields from seismic data. However, in this work we propose to use neural networks to predict the residual necessary for the filter stage of the two-stage MCMC. By using this method, the MCMC inversion avoids one of the major pitfalls of neural networks—the ignorance of the underlying physics. In our method, any model that is accepted by the neural network filter is then evaluated using the acoustic wave equation before the model is used to characterize the posterior distribution and compute uncertainty measures. Thus, we avoid physically infeasible outcomes.

Neural network two-stage MCMC (NNMCMC) consists of three phases:

1. Generate the training set of velocity field and residual pairs with one-stage Metropolis-Hastings MCMC using operator upscaling as a fast wavesolver.
2. Train the Neural Network.
3. Proceed with two-stage MCMC using the neural network filter.

To further reduce the computational cost of NNMCMC, we use operator upscaling for the acoustic wave equation in place of the full fine grid wave equation during the training set collection phase. Operator upscaling is a physics-aware, highly accurate surrogate for the acoustic wave equation which does not require averaging the velocity field and can resolve subwavelength heterogeneities (Vdovina et al., 2005; Korostyshevskaya and Minkoff, 2006; Vdovina and Minkoff, 2008).

We find that the NNMCMC algorithm results in a 65% reduction in time-per-trial and an 84% reduction in time-per-rejection over one-stage MCMC, including the time required to generate the training set. In addition, the acceptance rate in the NNMCMC increases to 86% from 29% in one-stage MCMC. This substantial decrease in time required and increase

10.1190/segam2019-3215402.1

in acceptance rate indicates that neural networks are an effective filter in two-stage MCMC for velocity inversion.

THEORY

Bayesian Inversion

One way to enable uncertainty quantification in velocity inversion is to approach the problem from a Bayesian perspective (Tarantola, 2005). We invoke Bayes' rule, which relates the posterior distribution we are trying to determine to the likelihood function (a measure of how well the simulated data given a model matches the true data) multiplied by the prior distribution of the model (what we think about the uncertainty of the model parameters before simulation). Bayes' rule states

$$P(\theta|d) \propto P(d|\theta)P(\theta) \tag{1}$$

where $P(\theta|d)$ is the posterior distribution of the model, $P(d|\theta)$ is the likelihood function, and $P(\theta)$ is the prior distribution of the model (Gelman et al., 2013).

In this work we assume a Gaussian likelihood function and a uniform prior distribution. The likelihood function is

$$P(d|\theta) = \exp\left(\frac{\|\mathscr{F}(\theta) - d\|}{-2\sigma^2\|d\|}\right) \tag{2}$$

where $\mathscr{F}(\theta)$ the the simulated data, d is the observed data, σ is a user-prescribed precision parameter related to the noise level in the data, and $\|\cdot\|$ is the L^2 norm.

Two Stage Markov Chain Monte Carlo

While computationally inexpensive deterministic solutions to Bayesian-formulated inverse problems exist, they require significant assumptions on the shape of the posterior distribution (e.g., the posterior distribution is Gaussian) (Tarantola, 2005). To avoid this assumption, we employ stochastic MCMC methods. MCMC techniques are designed to sample from a posterior distribution by constructing a Markov chain with the posterior distribution as the chain's equilibrium distribution (Brooks et al., 2011).

However, this process requires testing tens or hundreds of thousands of velocity fields to estimate the posterior distribution. For every proposed velocity field, we must compute the likelihood function (Equation (2)) which requires solution of the wave equation, $\mathscr{F}$. Unfortunately, the majority of velocity models are rejected after data simulation and not included in the estimation of the posterior distribution. Two-stage MCMC reduces the computational expense of MCMC by employing a computationally inexpensive filter to quickly reject unacceptable velocity models (Christen and Fox, (2005); Efendiev et al., (2006); Stuart et al. (2016); Stuart et al. (2019, in review)).

The **two-stage MCMC algorithm**:

1. **Model Generation:** Generate a velocity field proposal θ' from the proposal density $q(\theta'|\theta_n)$ where θ_n is the previously accepted velocity model.
2. **Filter:** Determine the velocity field proposal for the fine grid step, θ, where

$$\theta = \begin{cases} \theta' & \text{with probability } \rho(\theta_n, \theta') \\ \theta_n & \text{with probability } 1 - \rho(\theta_n, \theta') \end{cases} \tag{3}$$

and

$$\rho(\theta_n, \theta') = \frac{P_F(\theta'|d)q(\theta_n|\theta')}{P_F(\theta_n|d)q(\theta'|\theta_n)} \tag{4}$$

where $P_F(\theta_n|d)$ is the posterior distribution using the filter likelihood function.

3. **Fine Grid:** Update the next step in the Markov chain, θ_{n+1} with

$$\theta_{n+1} = \begin{cases} \theta & \text{with probability } \rho(\theta_n, \theta) \\ \theta_n & \text{with probability } 1 - \rho(\theta_n, \theta) \end{cases} \tag{5}$$

where

$$\rho(\theta_n, \theta) = \frac{P(\theta|d)P_F(\theta_n|d)}{P(\theta_n|d)P_F(\theta|d)}. \tag{6}$$

However, if $\theta' = \theta_n$ in the filter step, then $\theta_{n+1} = \theta_n$ with probability 1 in the fine grid step. Thus, the fine grid step can be omitted when a new model is rejected on the filter. This results in significant computational cost savings.

Operator Upscaling

To further reduce the computational expense of NNMCMC, during the training set construction phase we replace the fine grid computation with operator upscaling for the acoustic wave equation. Operator upscaling is an accurate approximation of the acoustic wave equation that can capture fine-scale heterogeneities in the velocity field (Vdovina et al., 2005). Due to significantly decreased communication between processors, operator upscaling reduces the computation time of solving the wave equation by approximately 45% (Stuart et al. (2019, in review)). In operator upscaling, the domain is decomposed into coarse grid and a fine grid within each coarse block.

The operator upscaling algorithm proceeds in four steps:

1. Solve for fine-grid unknowns in each coarse grid block. No communication between processes occurs at this stage.
2. Using the fine-grid solutions, solve for the coarse-grid unknowns in each coarse block.
3. Communicate only the coarse-grid unknowns to each adjacent process.
4. Repeat for the next time step.

See Vdovina et al. (2005), Korostyshevskaya and Minkoff (2006), Vdovina and Minkoff (2008), and Stuart et al. (2019, in review) for details on operator upscaling derivation, implementation, convergence, and timings.

Neural Network Two-Stage MCMC

A neural network consists of layers of interconnected nodes containing a weighting value and a nonlinear activation function. Each layer takes input from the preceding layer and produces output for the next layer by taking linear combinations

10.1190/segam2019-3215402.1

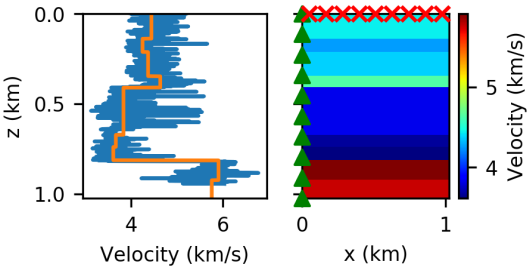

Figure 1: Left: The recorded well log p-wave velocity (blue) and nine-layer blocking (orange). The well log is courtesy of Pioneer Natural Resources. Right: The reference velocity generated from the well log. The red x's mark the line of 20 sources. The green triangles show the vertical receiver array.

of the weighted activation functions (Hastie et al., 2013). The training process updates the weights of each hidden node by minimizing the misfit between the training outcome associated with a given model and the outcome predicted by the neural network.

In this work, we use a neural network to compute the likelihood function in the filter posterior distribution. In the filter posterior distribution, the filter likelihood function is then

$$P_F(d|\theta) = \exp\left(-\frac{\mathscr{N}(\theta)}{2\sigma_F^2}\right) \tag{7}$$

where $\mathscr{N}(\theta)$ is the neural network prediction of the residual norm (see Equation (2)).

We modify the two-stage MCMC algorithm to include generating the training set and training the neural network. The **two-stage neural network MCMC algorithm (NNMCMC)**:

1. While $n < N$, where n is the number of velocity fields tested and N is the desired number for the training set, perform the *Metropolis-Hastings MCMC algorithm* with operator upscaling as the forward solver. The tested velocity models and residuals computed via operator upscaling form the training set.
2. Train the neural network using the training set generated in Step (1).
3. Perform the *two-stage MCMC algorithm with neural network filter*.

NUMERICAL EXPERIMENT

We demonstrate NNMCMC on a flat-layered model derived from a real well log from the Midland Basin. We block the well log into nine layers (see Figure 1) and assume the interface positions are known (i.e., we will invert for the velocity values within the layers only). This simple experiment provides compelling evidence that NNMCMC is worth further investigating on more complex models. The computational domain is 1024 m in x and 1024 m in z, with 1 m grid spacing and a 160 m absorbing boundary on all sides. The acquisition geometry contains a line of 20 sources along the top of

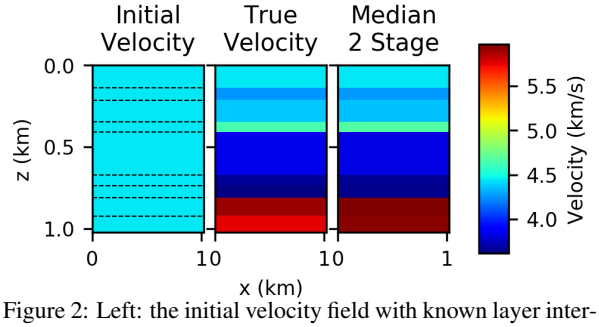

Figure 2: Left: the initial velocity field with known layer interfaces marked with black dashed lines. Center: The true velocity field. Right: the velocity field constructed from the median of the posterior distribution.

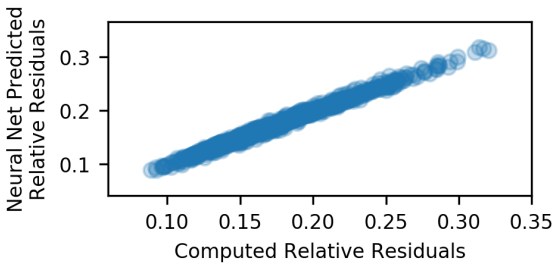

Figure 3: Relative residuals for data computed via operator upscaling for 1000 velocity fields, $\|\mathscr{F}_{OU}(m) - d\| / \|d\|$, vs. the relative residual predicted by the neural network.

the computational domain (red x's in Figure 1) and a vertical array of 512 receivers in a well at the left side of the computational domain (green triangles in Figure 1). Each source is a 20 Hz Ricker wavelet in time and a point source in space.

For the two-stage MCMC algorithm, we set both the filter and fine grid likelihood precision parameters (σ_F and σ in Equations (7) and (2), respectively) to 0.05. To construct the posterior distribution, we tested 37,000 velocity fields in one Markov chain and removed the first half of the chain as burn-in to reduce the impact of the starting velocity model on the posterior distribution (Gelman et al., 2013). We observe that the chain has converged to a steady state due to the stability of the posterior distribution estimates (not pictured). We assume our initial velocity field is constant-velocity with known layer interface positions (see Figure 2, left).

Neural Network Construction

We train the neural network on the first 10,000 velocity field samples of the Markov chain. For the 10,000 initial samples, we use operator upscaling in lieu of the fine grid wavesolver. To construct the neural net, we use the Tensorflow Python library (Abadi et al., 2016). We use a feed-forward neural network with three layers of 32 nodes each with Rectified Linear Units (ReLU) activation functions. We used 20% of the data for validation and trained the neural network using Adam optimization (Kingma and Ba, 2014). Figure 3 (right) demonstrates that the norm of the relative residual predicted by the neural network is very well correlated with the norm of the relative residual computed using operator upscaling, indicating that the neural network is a good filter for the two-stage

10.1190/segam2019-3215402.1

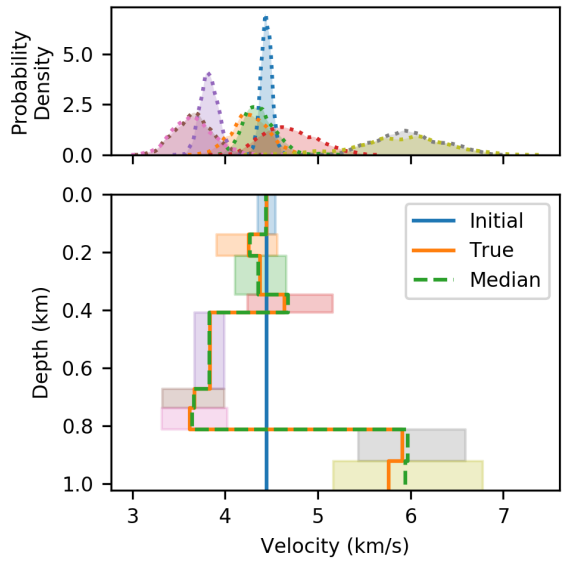

Figure 4: Top: the posterior distribution kernel density estimates for each layer velocity. Bottom: a one-dimensional slice of the initial velocity field (blue solid line), true velocity field (orange solid line), and median velocity field (green dashed line). The shaded regions in the bottom plot show the 90% HPD intervals for each layer. The color of each HPD interval on the bottom plot corresponds with the same colored posterior distribution on the top plot.

MCMC process and captures the physics well.

Inversion Results

Figure 2 compares the velocity field generated by the median of the posterior distribution (right) with the initial velocity field (left) and true velocity field (center). We see that the median velocity field matches the true velocity field very well. Figure 4 (bottom) shows a one dimensional slice of the median velocity field (green dashed line), true velocity field (orange solid line), and initial velocity field (blue solid line). In addition, we display the uncertainty information for each layer using a 90% highest posterior density (HPD) interval with shaded regions of various colors. Figure 4, top, shows the corresponding posterior distributions for each layer. The deepest layers (yellow and grey HPD intervals) and the narrowest layers (orange, red, brown, and pink HPD intervals) have the largest uncertainties and the broadest posterior distributions, whereas the shallowest layer (blue HPD interval) and the largest layer (purple HPD interval) have the narrowest HPD intervals and posterior distributions.

Timings

To determine the computational efficiency of the neural network two-stage MCMC algorithm, we will compare the time per velocity field tried and time per velocity field rejected for the neural network two-stage MCMC to the standard one-stage MCMC. The timings reported in Table 1 include generating the training set and training the neural network for the NNMCMC. In Table 1, we find the neural network two-stage MCMC reduces the time per velocity field tried by 65%, from 10 s to 3.5 s, and reduces the time per velocity field rejected by 84%, from 10 s to 1.6 s due to the negligible cost of evaluating a model on the neural network. The time per velocity field rejected includes both models rejected on the filter and models rejected on the fine grid.

	Standard Metropolis Hastings	NNMCMC	Percent Reduction
Seconds per Trial	10	3.52	65
Seconds per Rejection	10	1.58	84
Acceptance Ratio	0.29	0.86	N/A

Table 1: A comparison of the time per trial, time per rejection, and acceptance rate for the NNMCMC vs the one stage MCMC. All times for the NNMCMC include generating the training set and training the model.

We define the two-stage acceptance ratio as the proportion of velocity models accepted on the fine grid (Step (3) in the two-stage MCMC algorithm) to the number tested on the fine grid (i.e., accepted in the filter stage) (Christen and Fox, 2005; Efendiev et al., 2006). The acceptance rate increases from 29% for one-stage MCMC to 86% for the neural network two-stage MCMC.

CONCLUSION

In this paper, we introduced a novel sampling strategy, and demonstrated the efficacy of the NNMCMC algorithm for velocity full waveform inversion on a flat-layered velocity model derived from a real well log. In NNMCMC, we used a neural network trained to predict residuals from velocity fields as an inexpensive filter. The neural network is trained on velocity field and residual pairs generated with a physics-based operator upscaling solver. We find that the algorithm gives the expected uncertainty information for a substantially reduced computational cost. The NNMCMC algorithm reduced the time needed to try or reject each velocity model by 65% and 84%, respectively, while simultaneously increasing the acceptance ratio from 29% to 86%. Therefore, the NNMCMC algorithm is an effective and computationally efficient replacement for standard Metropolis-Hastings MCMC.

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation's Enriched Doctoral Training Program, DMS grant #1514808. This work has utilized the Cyber-Infrastructure Research Services computing resources at University of Texas at Dallas Office of Information Technology and the Seismology HPC cluster. F. Pereira was also funded in part by a Science Without Borders/CNPq-Brazil grant #400169/2014-2. We thank Tim Ullrich and Dennis Kubes for their guidance on neural networks.

10.1190/segam2019-3215402.1

REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, 2016, Tensorflow: A system for large-scale machine learning: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283.

Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep-learning tomography: The Leading Edge, **37**, 58–66, doi: https://doi.org/10.1190/tle37010058.1.

Brooks, S., A. Gelman, G. L. Jones, and X.-L. Meng, 2011, Handbook of Markov Chain Monte Carlo: Chapman and Hall/CRC.

Christen, J. A., and C. Fox, 2005, Markov Chain Monte Carlo using an approximation: Journal of Computational and Graphical Statistics, **14**, 795–810, doi: https://doi.org/10.1198/106186005X76983.

Efendiev, Y., T. Hou, and W. Luo, 2006, Preconditioning Markov Chain Monte Carlo simulations using coarse-scale models: SIAM Journal on Scientific Computing, **28**, 776–803, doi: https://doi.org/10.1137/050628568.

Frangos, M., Y. Marzouk, K. Willcox, and B. van Bloemen Waanders, 2011, Surrogate and reduced-order modeling: A comparison of approaches for large-scale statistical inverse problems, *in* L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. van Bloemen Waanders, and K. Willcox, eds., Large-scale inverse problems and quantification of uncertainty: John Wiley & Sons, 123–150.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, 2013, Bayesian data analysis, 3rd ed.: Chapman and Hall/CRC.

Ginting, V., F. Pereira, M. Presho, and S. Wo, 2011, Application of the two-stage Markov Chain Monte Carlo method for characterization of fractured reservoirs using a surrogate flow model: Computational Geosciences, **15**, 691–707, doi: https://doi.org/10.1007/s10596-011-9236-4.

Hastie, T., R. Tibshirani, and J. Friedman, 2013, The elements of statistical learning: Data mining, inference, and prediction: Springer Science & Business Media.

Higdon, D., C. S. Reese, J. D. Moulton, J. A. Vrugt, and C. Fox, 2011, Posterior exploration for computationally intensive forward models, *in* S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds., Handbook of Markov Chain Monte Carlo: Chapman and Hall/CRC, 401–418.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv:1412.6980.

Korostyshevskaya, O., and S. E. Minkoff, 2006, A matrix analysis of operator-based upscaling for the wave equation: SIAM Journal on Numerical Analysis, **44**, 586–612, doi: https://doi.org/10.1137/050625369.

Lewis, W., and D. Vigh, 2017, Deep learning prior models from seismic images for full-waveform inversion: 87th Annual International Meeting, SEG, Expanded Abstracts, 1512–1517.

Mosegaard, K., and A. Tarantola, 1995, Monte Carlo sampling of solutions to inverse problems: Journal of Geophysical Research, **100**, 12431–12447, doi: https://doi.org/10.1029/94JB03097.

Rosenthal, J. S., 2011, Optimal proposal distributions and adaptive MCMC, *in* S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, eds., Handbook of Markov Chain Monte Carlo: CRC Press, 93–111.

Roth, G., and A. Tarantola, 1994, Neural networks and inversion of seismic data: Journal of Geophysical Research, **99**, 6753–6768, doi: https://doi.org/10.1029/93JB01563.

Sambridge, M., and K. Mosegaard, 2002, Monte Carlo methods in geophysical inverse problems: Reviews of Geophysics, **40**, 1009, doi: https://doi.org/10.1029/2000RG000089.

Strang, G., 2019, Linear algebra and learning from data: Wellesley-Cambridge Press.

Stuart, G. K., W. Yang, S. Minkoff, and F. Pereira, 2016, A two-stage Markov Chain Monte Carlo method for velocity estimation and uncertainty quantification: 86th Annual International Meeting, SEG, Expanded Abstracts, 3682–3687.

Tarantola, A., 2005, Inverse problem theory and methods for model parameter estimation: SIAM.

Vdovina, T., and S. Minkoff, 2008, An a priori error analysis of operator upscaling for the acoustic wave equation: International Journal of Numerical Analysis and Modeling, **5**, 543–569.

Vdovina, T., S. E. Minkoff, and O. Korostyshevskaya, 2005, Operator upscaling for the acoustic wave equation: Multiscale Modeling & Simulation, **4**, 1305–1338, doi: https://doi.org/10.1137/050622146.