Combining global optimization and boundary integral methods to robustly estimate subsurface velocity models
Gregory Ely, Massachusetts Institute of Technology, Alison Malcolm, Memorial University of Newfoundland, David Nicholls, University of Illinois at Chicago

SUMMARY
In this paper, we combine a fast wave equation solver using boundary integral methods with a global optimization method, namely Particle Swarm Optimization (PSO), to estimate an initial velocity model. Unlike finite difference methods that discretize the model space into pixels or voxels, our forward solver achieves significant computational savings by constraining the model space to a layered model with perturbations. The speed and reduced model space of the forward solve allows us to use global optimization methods that typically require numerous evaluations and few unknown variables. Our technique does not require an initial guess of a velocity model and is robust to local minima, unlike gradient descent frequently used in methods for both initial velocity model estimation and full waveform inversion. We apply our inversion algorithm to several synthetic data sets and demonstrate how prior information can be used to greatly improve the inversion.

INTRODUCTION
Most seismic processing techniques rely on an accurate velocity model to obtain meaningful results. Incorrect velocity models can hamper processing and lead to erroneous interpretation of seismic data. In particular, inversion techniques such as Full Waveform Inversion (FWI) are highly sensitive to the initial velocity. Without a good initial velocity estimate, FWI will converge to local minima with artifacts (Virieux and Operto, 2009).

Unlike global seismology, exploration and regional scale velocity models can be poorly constrained. Initial velocity models can be built using a variety of techniques such as travel time tomography, NMO semblance analysis, and even full waveform inversion at very low frequencies (Woodward et al., 2008). Most methods of constructing an initial velocity model rely on expensive travel-time or wave equation solvers and use gradient based approaches that are susceptible to local minima. Furthermore, travel time tomography does not use the full wavefield and requires the picking of arrivals as well as long offsets. Frequently noise levels are too high for full waveform inversion to be performed at frequencies where the problem is sufficiently convex and thus it can be plagued by local minima Pratt (1999); Sorgue (2006).

To combat the problem of local minima in geophysical inverse problems, researchers have applied global optimization techniques that are less susceptible to local minima (Sambridge and Moser, 2002; Sen and Stoffa, 2013). Researchers also make use of different objective functions and regularization techniques (Burstedde and Ghattas, 2009; van Leeuwen and Herrmann, 2013) to make the inversion more convex, mitigating the need for global solvers.

In this paper we integrate a fast Helmholtz solver with the global optimization method Particle Swarm Optimization (PSO) to invert a velocity model without an initial model. We first give an overview of the field expansion method we use for quickly solving the Helmholtz equation and modify it to reduce artifacts and accurately simulate ex-

FIELD EXPANSION
In this section we briefly describe the field expansion method to solve the Helmholtz equation for a perturbed layer medium. Similar to some reflectivity codes, such as that by Bouchon (2003), this solver operates in the spatial Fourier domain and as a result describes the response of an infinitely repeating domain subject to a periodic point source, as shown in Figure 1. A detailed description and derivation of the solver can be found in Malcolm and Nicholls (2011).

The forward solver describes a velocity model consisting of $M$ layers with velocities $c_m$. Each layer is divided by an interface at depth $a_0$; this interface is perturbed from flat by the function $g_m(x)$, which has a zero mean. This is illustrated in Figure 1. The scattered field in each layer $v_m(x,y)$ satisfies the acoustic Helmholtz equation

\[ \nabla^2 v_m + k_m^2 v_m = 0, \]

at some frequency $f$. We begin by describing the case for a set of $M$ flat layers and then expand the forward solver to handle perturbations of the interface shape.

Flat Layers
In the flat layer case we set the perturbation $g_m(x) = 0$ and apply the outgoing wave condition (OWC) to the velocity model. This requires that in the uppermost layer, energy must be propagating upwards away from the model and in the bottom layer, energy must be propagating away from the model downwards. Subject to the OWC and the Helmholtz equation the scattered field, $v_m$, in the spatial frequency domain indexed by modes $p$ has the form

\[ v_m(x,y) = \sum_{p=-\infty}^{\infty} \tilde{d}_p e^{i(\alpha_p y - \beta_p x)} + u_p e^{i(\alpha_p y + \beta_p x)} \]

and $\alpha_p = \alpha + (2\pi/d) p$, $\beta_p = \sqrt{k_m^2 - \alpha_p^2}$ where $\alpha_p$ is a constant determined by the angle of the incident source, $d_0$ is the midpoint of each of the flat layers and $d$ is the domain spacing (Pélig, 1980). For the shallowest and deepest layers, the midpoint is equal to the closest interface, $d_0 = a_1$ and $d_0 = a_M$. Note that in Equation 2 the up-going $u_p$ and down-going $d_p$ spatial Fourier coefficients completely describe the field as all of the other variables are determined directly from the layered model. To determine these unknown coefficients, we impose the Dirichlet and Neumann boundary

\[ \alpha_p^2 < k_m^2 \]

\[ \alpha_p^2 > k_m^2 \]

Figure 1: An illustration of the repeating domain structure implicit in the forward solver. The domain is periodic and there is a point source at the same location in each of the repeating domains.
conditions

\[
\begin{align*}
v_{m-1} - v_m &= \xi_m \quad \text{at} \quad y = a_m \\
\partial y v_{m-1} - \partial y v_m &= \psi_m \quad \text{at} \quad y = a_m
\end{align*}
\] (4)

for each layer at depth \(a_m\). In Equation 4, \(\xi\) and \(\psi\) are the incident field and the normal derivative of the incident field respectively. Combining the expression for the scattered field, Equation 2, and the boundary conditions, we can express the equalities given by the Equation 4 as a system of linear equations for each spatial mode \(p\). We write this system as

\[
\mathbf{\xi}_p = \mathbf{f}_p,
\] (5)

where \(\mathbf{\xi}_p\) is a penta-diagonal matrix with entries completely determined by the velocity model,

\[
\mathbf{\xi}_p = (u_0, d_1, u_1, \ldots, d_{M-1}, u_{M-1}, d_M)^T
\] (6)

is the unknown vector of up-going and down-going coefficients, and

\[
\mathbf{f}_p = (\xi, \psi, 0, 0)^T
\] (7)

is the source vector. Because the spatial modes are orthogonal, the choice coefficients \(u_p\) and \(d_p\) can be calculated independently from each other mode by mode. Once calculated, the field can be determined by summing across all calculated modes using Equation 2. The entries of \(\mathbf{\xi}_p\) are given in Malcolm and Nicholls (2011) and follow from the insertion of Equation 2 into Equation 4. To insert a point source into this formulation, we specify the incident field for a periodic point source \(s_{p,n}\) at \(y = a_0\). This periodic point source introduces the same \(f\) in the right hand side of the flat layer case

\[
\xi_p(x, y) = \frac{1}{2\pi d} e^{i(p(x - a_n) + p(\psi(y - a_n))}
\] (8)

We summarize the steps needed to calculate the field for a perturbed velocity model in Algorithm 1.

\section*{Algorithm 1 Perturbed Helmholtz Solver}

\begin{itemize}
\item Determine number of spatial modes \(P\)
\item Determine order of Taylor series \(N\) for \(p = 1\) to \(P\) do
\item Construct source vector \(\mathbf{R}_p\)
\item Construct velocity model matrix \(\mathbf{\xi}_p\)
\item Solve \(\mathbf{\xi}_p \mathbf{\psi}_p = \mathbf{f}_p\) to get \([u_{\psi,n}, d_{\psi,n}]\)
\item for \(n = 1\) to \(N\) do
\item from \(d_{n-1}\) and \(u_{n-1}\) Construct \(\mathbf{R}_{p,n}\)
\item Solve \(\mathbf{\xi}_p \mathbf{\psi}_p = \mathbf{f}_p\) to get \([u_{\psi,n}, d_{\psi,n}]\)
\item end for
\item Use \([u_{\psi,n}, d_{\psi,n}]\) to calculate field \(v_n\) with Equation 9
\end{itemize}

\section*{Practical Considerations}

The forward solver described in the previous section works both in the spatial and temporal frequency domain, making it difficult to directly compare to real or synthetic seismic data. If the domain spacing \(d\) is small, energy from other domains can cross into each other, generating edge effects. Setting the domain spacing very large relative to the interest area greatly mitigates this effect because the energy of a point source decreases due to geometric spreading. This comes at a small computational cost of increasing the number of spatial frequencies \(p\) needed to determine the solution.

In addition to the leakage effects, the source’s periodic nature generates a periodic standing wave known as the Rayleigh or Wood’s anomalies. The mathematical origin of this problem can be seen in Figure 4. In this series expansion, it is obvious that the expression for \(\xi_p(x, y)\) will become very large if one of the low order \(\beta_p\) is very near zero. From the expression for \(\beta_p\) in Equation 3, this

\[
\xi_p(x, y) = \frac{1}{2\pi d} e^{i(p(x - a_n) + p(\psi(y - a_n))}
\] (9)

Like the expression for the flat layer field, \(d_{(\psi,n)}\) and \(u_{(\psi,n)}\) completely determine the field. In addition, the Dirichlet and Neumann boundary conditions are applied to the perturbed interface, changing the boundary conditions to

\[
\begin{align*}
v_{m-1} - v_m &= \xi_m \quad \text{at} \quad y = a_m + g_m(x) \\
\partial y v_{m-1} - \partial y v_m &= \psi_m \quad \text{at} \quad y = a_m + g_m(x)
\end{align*}
\] (10)

This expansion about \(e\) can be applied to the system of equations in Equation 10 as done in Malcolm and Nicholls (2011). When expanded, a recursion relationship becomes apparent such that field coefficients of order \(n\) are determined by the lower order terms \(0\ldots(n - 1)\) and each higher order coefficient can be determined by solving

\[
\mathbf{\xi}_p \mathbf{\psi}_p = \mathbf{R}_p
\] (11)

for a different right hand side \(\mathbf{R}_p\) in Equation 11. For the zeroth order term, \(\mathbf{R}_{p,0}\) is equal to the right hand side of the flat layer case given by Equation 5. For higher order terms, the entries are found by equating terms with the same \(n\). The resulting coefficients are given in Malcolm and Nicholls (2011).
method to handle large perturbations, a Padé approximant (Baker and Graves-Morris, 1996) is used to enhance both the rate of convergence and permissible perturbation size as detailed in Malcolm and Nicholls (2011). With the addition of the Padé approximant, absorptive top layer, and large spacing between domain size, data generated using the field expansion method is comparable to finite difference simulations, as shown in Figure 4.

**OPTIMIZATION METHOD**

In this section we describe the global optimization method, PSO, used to solve the inverse problem of estimating a velocity model given a set of measured seismic traces. PSO is modeled on the behavior of flocks in which a collection (swarm) of potential solutions (agents) move through model space and communicate their own successes and failures with the other agents to find a global minima (Eberhart and Kennedy, 1995).

In the PSO algorithm, a collection of agents are randomly initialized across the entire range of model space with a random velocity at which they move through model space. At each iteration, all agents evaluate a cost function based on their current position; in our implementation this cost function is the mean square error between the measured and observed fields. All agents keep track of the personal best, lowest scoring, location in model space it has visited and its corresponding score. The agents then move through model space according to their velocity. Each agent’s velocity is then updated to accelerate the agent towards its personal best location, the lowest scoring location it has visited in model space, and the neighborhood best location, the lowest scoring location across a subset of agents.

We implemented two forms of PSO: traditional PSO and local best PSO. The two variants of PSO only differ in how the neighborhood best is defined for each agent. In traditional PSO, each agent’s velocity is influenced by the best known solution across the entire swarm. In local best PSO, each agent instead only sees the best known solution among its adjacent index neighbors (Bratton and Kennedy, 2007). For example in a swarm of 10 agents, the 5th agent’s best neighborhood location is calculated among the personal best of agents with indices [4,5,6] and for the 10th agent the best is determined among indices [9,10,1]. This variation of the algorithm slows the speed of which information is transmitted across the swarm and prevents the algorithm from converging too early to a local minima (Kennedy and Mendes, 2006).

The steps can be summarized through the following pseudo code describing the swarm, consisting of \( N_a \) agents each with position vector \( \mathbf{x} \) and velocity vector \( \mathbf{v} \). For each agent in the swarm,

1. Initialise personal best score to infinity: \( p_i = \infty \)
2. Initialise swarm position from a uniform distribution across all of model space: \( x_i = U[-1,1] \)
3. Initialise swarm velocity from a uniform distribution in some range: \( v_i = U[-vel_{\text{max}}, vel_{\text{max}}] \)
4. Evaluate forward solve and calculate agent’s score: \( G(X_i) \)
5. Record personal best score and location if better than current best: if \( p_i < G(X_i) \) then \( p_i = G(X_i) \) and \( x_i = x_i \)
6. Determine neighborhood best location, \( g_i \). If local best PSO: \( g_i \leftarrow \min(p_1, p_2, \ldots, p_{N_a}) \). If normal PSO: \( g_i \leftarrow \min(p_1^i, p_2^i, \ldots, p_{N_a}^i) \).
7. Calculate new position: \( x_i = x_i + v_i \)
8. Calculate stochastic difference vectors, difference between best locations multiplied by uniform distribution: \( d_k = U[0,1] \times (g_k - x_i), \ d_j = U[0,1] \times (p_j - x_i) \)
9. Calculate new velocity: \( v_i = v_i + \gamma v_i + \alpha_1 d_j + \alpha_2 d_k \)
10. Clamp velocities: if any entries of \( v_i \) exceed \( vel_{\text{max}} \) or \( -vel_{\text{max}} \) then set the entries to \( vel_{\text{max}} \) or \( -vel_{\text{max}} \)
11. Go to step 4 until maximum number of iterations is reached.

In the expression for the velocity update above, \( \gamma \) is the inertia term controlling how much the previous velocity is maintained from the prior iteration and \( \alpha_1 \) and \( \alpha_2 \) are the acceleration terms that determine how much the personal best and neighborhood best alter each agent’s velocity at each iteration. For all of the simulations performed in this paper the following parameters were used: \( N_a = 40, \ \gamma = 0.9, \ \alpha_1 = 1.49, \ \text{and} \ \alpha_2 = 1.49 \). The maximum velocity \( vel_{\text{max}} \) was clamped to \( \pm 0.05 \) of the total search space. Note that in the above algorithm, the evaluation of each agent’s cost function is independent from one another and the forward solvers can be parallelized across all agents.

To apply PSO to the field expansion framework, the velocity model
must be converted into a vector of real numbers. For an M layer
flat velocity model, the model can easily be converted into a vector
of length \(2M-1\) containing \(M\) velocities and \(M-1\) layer boundaries,
one for each of the boundaries. For the perturbed layer case, each of
the layers are assigned \(N_{\text{perurbed}}\) values that described the height
of perturbation at \(N_{\text{perurbed}}\) equally spaced locations across the region
of interest of each interface. A cubic interpolation was then applied
along each layer to define the surface for all locations. Under this
parameterization, a perturbed velocity model could be described by
\((M-1) \times N_{\text{perurbed}} + 2M-1\) values. The model is then be normal-
ized to the range of [-1, 1] by dividing each value by the permissible
perturbation, velocity, or layer depth limit.

**EXPERIMENTS**

IBest vs PSO

To compare the convergence characteristics and robustness of local
best PSO vs normal PSO, we generated a synthetic dataset using the
field expansion method consisting of 3 layers with interfaces at a depth
of 500 and 1200 meters and velocities 1500, 2500, 3500 m/s from
shallower to deepest. The domain size or the spacing between periodic
point sources was set to 20 kilometers with the synthetic shot fired in
the center of the domain with 512 receivers spanning \(\pm 3000\) m from
the source. Both regular PSO and local best PSO were run for 1000
iterations with a velocity range of \([1000 \ 6000]\) m/s, interface depth
range of \([100 \ 2000]\) meters, and perturbation limit of 0 to force the
layers to be flat. The inversion was then run 50 times for each algo-

Figure 5 (left) shows the mean difference in cost function as a
function of iteration and Figure 5 (right) shows percentage of simul-
ations whose best score was less than .25 as a function of iteration.
The local best PSO is initially slow to converge to a minima, but consist-
tently reaches the minima more frequently. Table 1 shows the mean
and standard deviation for the inversions.

Prior

In the previous simulations, we unrealistically assumed that we had
no prior information of the velocity model other than constraining the
interface and velocity limits to reasonable values. However, in almost
all instances we would have some prior knowledge of the velocity ob-
tained from methods such as NMO. For example, one could draw a
set of velocity model from the NMO semblance image to generate a
collection of prior velocity models. To mimic this, instead of assum-
ing a completely random starting model from the entire velocity and
interface limits, we adopt a semi-global approach in this experiment.

We draw an initial model from a uniform distribution around the true
model with a spread of \([-300, 300]\) m for the interface limit and \([-500,
500]\) m/s for the velocity distribution. As in the simulations above,
the swarm can move through the entire range of velocities and inter-
face limits. Only the initial population was altered to incorporate prior
knowledge of the velocity limits. We then ran the inversion using local
best PSO 50 times with the same parameters as the two simulations
in the previous sub-section, with the maximum number of iterations
decreased from 1000 to 500 as the algorithm converged much more
quickly.

Changing the initial swarm from a purely random model to a more
informed prior model greatly improved the performance of the inver-
sion. As shown in Figure 5, all of the inversions reached a misfit of less
than .25 within 100 iterations and the average data misfit as function
of iteration is significantly lower than the traditional or local best PSO.
The inversions using the prior achieved extremely accurate estimates of
the velocity models at 500 iterations, as shown in Table 1.

**Anticline with Prior**

In the final experiment, we generate a 3 layer velocity model with a
velocity of \([1500, 2500, 3500]\) m/s from shallowest to deepest. The
top interface is flat while the deeper interface has an anticline shaped
perturbation, as shown by the dashed lines in Figure 6. We then gener-
ate 6 synthetic shot records at 3 Hz at the locations shown in Figure 6.
We next perform local best PSO with an informed prior with a uniform
distribution around the true model with a spread of \([-400, 400]\) m for
the interface location and \([-500, 500]\) m/s for the velocity distribution.
For the experiment, the interface was discretized into 13 interpolation
points and we truncate the Taylor series expansion of the field after 8
terms. We ran the inversion 5 times for 500 iterations. The results are
summarized in Figure 6. All of the runs of the algorithm were suc-
cessful in recovering the anticline interface from a limited number of
shot records.

**CONCLUSION**

In this paper we altered the domain size and added a small absorptive
term to make the field expansion solver comparable to finite differ-
ence methods. Based on this fast solver, we presented an inversion
algorithm using PSO to estimate a velocity model without an initial
guess. Furthermore, we demonstrated how prior information can be
incorporated into the PSO framework to greatly improve accuracy of
the recovered velocity model. This opens up the possibility of estimat-
ing velocities, with error bounds, without the specification of an initial
velocity model.

**ACKNOWLEDGMENT**

This material is based upon work supported by the National Science
Foundation under Grant No. (DGE-0806676) & (NSF-1115406) and the
MIT Earth Resources Laboratory.
EDITED REFERENCES
Note: This reference list is a copyedited version of the reference list submitted by the author. Reference lists for the 2015 SEG Technical Program Expanded Abstracts have been copyedited so that references provided with the online metadata for each paper will achieve a high degree of linking to cited sources that appear on the Web.

REFERENCES


