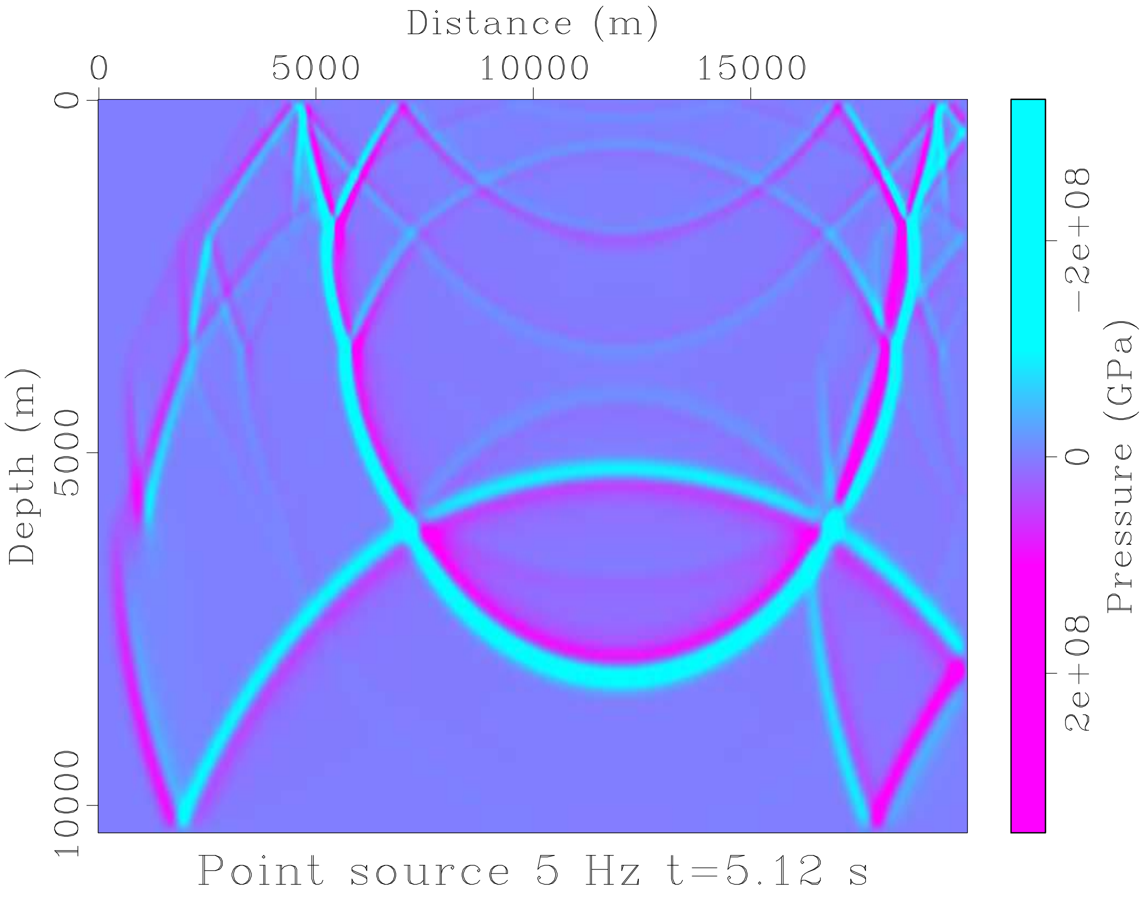
THE RICE INVERSION PROJECT

Mario Bencomo, Jie Hou, Yin Huang, William Symes, and Muhong Zhou

Annual Report 2014



TRIP14 — TABLE OF CONTENTS

Seismic Inverse Problems: Recent Developments in Theory and Practice

William W. Symes

ABSTRACT

The widely used least-squares approach to seismic inversion may fail due to insufficient data spectrum. The model extension modification to least-squares inversion can eliminate this full bandwidth requirement. For linearized extended modeling about smooth background parameters, a geometric-optics based analysis explains the increased robustness of extended inversion, and relates its characteristics explicitly to those of travel-time inversion.

INTRODUCTION

Inference of earth structure from seismic data may be formulated as an inverse problem, via a choice of earth model domain M, data range D, and forward or modeling map $\mathcal{F} : M \rightarrow D$. M and D may be (subsets of) Hilbert spaces. and the inverse problem set as a nonlinear least squares problem: given d, choose m to minimize

$$J_{\rm OLS}[m;d] = \frac{1}{2}\|\mathcal{F}[m] - d\|^2. \tag{1}$$

Specific choices of M, D, and $\mathcal{F}$ appropriate for seismology are suggested below. For now, note that seismic measurements are *time series*, so that D consists of functions of time, amongst other experimental variables, with definite and limited bandwidth.

This formulation, along with many variations and refinements, has been studied since the late '70s. In the last ten years, 3D computational realizations have become feasible, and successful least squares inversions have revealed subsurface maps of unprecedented clarity - see for example Virieux and Operto (2009); Plessix et al. (2010); Vigh et al. (2010). However, the oscillatory nature of seismic data combines with the nonlinearity of $\mathcal{F}[m]$ to create numerous stationary points of $J_{\rm OLS}$, most of which are uninformative about earth structure . Because of problem size, only rapidly convergent iterations related to Newton's method are computationally feasible. These are local optimization methods, finding stationary points, whereas only the global minimizer or a close approximation contains useful information. Low frequency data helps by providing a larger region of attraction for the global minimizer, but it is not always possible to acquire data with sufficiently high signal/noise ratio at sufficiently low frequencies to take advantage of this observation (Plessix et al., 2010).

This paper focuses on two ideas for modification of $J_{\rm OLS}$ to produce optimization tasks more amenable to calculus-based methods: (i) relaxation via extension, that is, enlargement of the domain of $\mathcal{F}$, and (ii) linearization and scale separation, which permit analysis of the extended inverse problem via high frequency asymptotics. Both ideas originate in industrial seismic data processing; the specific replacement for (1) presented here is a variant of *Wave Equation Migration Velocity Analysis*, or *WEMVA*. Our principal result is this: the objective Hessian at a consistent model-data pair constrains perturbations in travel time *slopes*. That is, this variant of WEMVA is related to *stereotomography* (Billette and Lambaré, 1998), at least in quadratic approximation. Similar assertions seem to be widely believed of WEMVA in general, but so far as I know this paper sketches the first explicit demonstration.

The extension described below is only one of many possible. See Symes (2009), Symes (2008) for other possibilities and extensive overview.

The abbreviated discussion to follow glosses over many details of analysis and physics, not all of which are trivial or acknowledged explicitly.

EXTENSION AND LINEARIZATION

The simplest useful physical description of seismic wave propagation is provided by *constant density acoustics*, in which earth structure is characterized by a *sound velocity* $v(\mathbf{x})$, a positive function of position $\mathbf{x} \in \mathbf{R}^3$, or equivalently by $m = v^{-2}$. The extension discussed here replaces m with a bounded coercive self-adjoint operator $\bar{m}$ on $L^2(\mathbf{R}^3)$. In this extended acoustics, the pressure field p resulting from a point isotropic radiator with time dependence $w(t)$ located at position $\mathbf{x}_s$ is the solution of the initial value problem

$$\begin{aligned}\left(\bar{m}\frac{\partial^2 p}{\partial t^2} - \nabla^2 p\right)(\mathbf{x},t;\mathbf{x}_s) &= w(t)\delta(\mathbf{x}-\mathbf{x}_s),\\ p(\mathbf{x},t;\mathbf{x}_s) &= 0,\, t << 0. \qquad (2)\end{aligned}$$

The physical significance of this extension is the relaxation of the "no action at a distance" axiom of continuum mechanics. Abusing notation by writing $\bar{m}$ also for the distribution kernel of $\bar{m}$, if $(\mathbf{x},\mathbf{y}) \in \text{supp}\, \bar{m}$, then a change in volume at $\mathbf{x}$ may result in a change in pressure at $\mathbf{y}$. Stolk (2000) and Blazek et al. (2013) show that problems like (2) have sensible solutions, depending smoothly on the coefficient $\bar{m}$ in an appropriate sense. Define the extension operator E by $Em(\mathbf{x},\mathbf{y}) = m(\mathbf{x})\delta(\mathbf{x},\mathbf{y})$: that is, E maps a function of $\mathbf{x}$ into the corresponding multiplication operator. Then with $m = v^{-2} \in L^\infty(\mathbf{R}^3)$ and $\bar{m} = Em$ (the "physical" case), (2) becomes the ordinary acoustic wave equation.

A simplified model of seismic data is the trace of p on a horizontal surface $\{\mathbf{x} : x_1 = z_r\}$ (that is, x_1 is the depth coordinate, traditionally denoted z - thus z_r for the constant depth of recording). I will ignore all issues arising from finite sampling, and regard the receivers as occupying a continuum of dimension $n-1$. Likewise, regard the positions of the sources as occupying a continuum, for convenience a subset of another horizontal surface $\{\mathbf{x} : x_1 = z_s\}$. Denote the horizontal coordinates by $(x_2, ..., x_n) = \mathbf{x}'$, and the source and receiver coordinate vectors by $\mathbf{x}_r = (z_r, \mathbf{x}'_r), \mathbf{x}_s = (z_s, \mathbf{x}'_s)$.

With these conventions, the data recorded in a seismic survey define a function of $\bar{m}$, the the *forward map* or *modeling operator* $\mathcal{F}$:

$$(\mathcal{F}[\bar{m}])(\mathbf{x}'_r, t; \mathbf{x}'_s) = \chi_{r,s}(\mathbf{x}'_r, \mathbf{x}'_s)\chi_t(t)p((z_r, \mathbf{x}'_r), t; (z_s, \mathbf{x}'_s)) \tag{3}$$

in which $\chi_{r,s} \in C_0^\infty(\mathbf{R}^{2(n-1)})$ and $\chi_t \in C_0^\infty(\mathbf{R})$ account for the finite extent of source and receiver positions and duration of recording. Write $\mathcal{F}_\delta[\bar{m}]$ for the special case defined by $w = \delta$ in (2). Then $\mathcal{F}_\delta[\bar{m}]$ is the distribution kernel of an operator closely related to the Dirchlet-to-Neumann map which plays a role in many inverse problems.

Introduce the linearization about a physical coefficient m:

$$F_\delta[m] = D\mathcal{F}_\delta[E[m]],\ F[m] = w * F_\delta[m] \tag{4}$$

and the "horizontal offset" constraint: apply F only to operator perturbations of m satisfying $\bar{m}(\mathbf{x}, \mathbf{y}) = \bar{m}'(x_1, \mathbf{x}', \mathbf{y}')\delta(x_1 - y_1)$. That is, a volume perturbation can only result in a pressure perturbation at the same depth. [This constraint can be weakened]. We assume from now on that m is smooth, so that solutions of (2) can be approximated by geometric optics. Then under certain conditions on the geometry of bicharacteristics,

1. $F_\delta[m]$ is a bounded operator on $L^2(\mathbf{R}^5)$;

2. $F_\delta[m]^* F_\delta[m]$ is a member of the class OPS^0 of *pseudodifferential operators* of order 0, elliptic in the conic set of "illuminated reflectors". In fact, for any pseudodifferential operator P of order 0, $F_\delta[m]^* P F_\delta[m]$ is pseudodifferential. Similarly, for any pseudodifferential P of order 0, $F_\delta[m] P F_\delta[m]^*$ is also a pseudodifferential operator of order 0. The symbols of both operators, hence the operators themselves, depend smoothly on m.

3. For each smooth $m, \delta m$, there exists a pseudodifferential operator $Q[m, \delta m]$ of order 1, for which
$$DF_\delta[m, \delta m] = D^2\mathcal{F}_\delta[E[m]](E[\delta m], \cdot) = F_\delta[m]Q[m, \delta m].$$
Q is essentially skew-symmetric:
$$Q[m, \delta m] + Q[m, \delta m]^* = R[m, \delta m] \in OPS^0(\mathbf{R}^5).$$
Q is smooth in m, linear in δm.

The necessary geometric conditions, and the meaning of "illuminated reflector", are explained for example by Stolk et al. (2009) and ten Kroode (2012). The first result in item 2 is well-known, the second is peculiar to extended modeling. The factorization of DF does not seem to have been described in the prior literature.

A WELL-POSED INVERSE PROBLEM

The linearized version of the inverse problem introduced earlier is: given $\delta d = d - \mathcal{F}[m]$, find $m, \delta m$ to minimize $\|F[m]E[\delta m] - \delta d\|$. Introduce a linear map A on the space of distribution kernels whose kernel equals the range of E. For example, under the "horizontal offset" condition, a natural choice would be $A\bar{m} = [\mathbf{x}', \bar{m}']$. In terms of distribution kernels, $A\delta\bar{m} = (\mathbf{x}' - \mathbf{y}')\bar{m}'(x_1, \mathbf{x}', \mathbf{y}')$. This choice of A acts on the kernel of $\bar{m}$ as a pseudodifferential operator of order 0. For consistent data, $\delta d = F[m]\delta\bar{m}, A\delta\bar{m} = 0$, the solution $m, \delta\bar{m}$ minimizes

$$J_{\rm DS}[m, \delta\bar{m}] = \frac{1}{2}\left(\|F[m]\delta\bar{m} - \delta d\|^2 + \alpha^2\|A\delta\bar{m}\|^2\right).$$

for any positive α.

$J_{\rm DS}$ is just as likely to exhibit severe nonconvexity as is $J_{\rm OLS}$ - indeed the former includes the latter as a section. Kern and Symes (1994) observed that for this type of problem, the *reduced objective*

$$\tilde{J}_{\rm DS}[m] = \min_{\delta\bar{m}} J_{\rm DS}[m, \delta\bar{m}]$$

is smooth, independently of the spectrum of w - in fact even for limit case $w = \delta$. This conclusion follows immediately from the facts cited above. The normal operator

$$N[m] = F_\delta[m]^* F_\delta[m] + \alpha^2 A^* A \in OPS^0$$

is microlocally elliptic, and can be made elliptic and indeed invertible with suitable regularization of F_δ - assume that this has been done. Then

$$\tilde{J}_{\rm DS}[m] = J_{\rm DS}[m, \delta\bar{m}[m]],\ \ \delta\bar{m}[m] = N[m]^{-1}F_\delta[m]^*\delta d.$$

Thus $\tilde{J}_{\rm DS}[m]$ can be rewritten as a sum of terms of the form $\langle d, F_\delta[m](...)F_\delta[m]^* d\rangle$: the elided factors are products of pseudodifferential operators depending smoothly on m, whence the second fact cited in item 2 implies the claimed smoothness.

Remark: The analogue of $F_\delta[m](...)F_\delta[m]^*$ for ordinary, rather than extended, modeling is *not* pseudodifferential, which implies that the linearized problem stated at the beginning of this section has non-smooth objective for $w = \delta$, leading to the observed proliferation of stationary points for band-limited w.

As $\tilde{J}_{\rm DS}$ is smooth, it is well-approximated near a global minimizer m_* by its Hessian quadratic form. This form is easiest to analyze in case the data is *consistent* with m_*, that is, $F[m_*]\delta\bar{m}[m_*] = \delta d, A\delta\bar{m}[m_*] = 0$. A page or so of algebra, in which one systematically makes use of consistency to drop various terms, results in

$$D^2\tilde{J}_{\rm DS}[m_*](\delta m_1, \delta m_2) = \langle [A, Q[m_*, \delta m_1]]\delta\bar{m}[m_*], [A, Q[m_*, \delta m_2]]\delta\bar{m}[m_*]\rangle.$$

To understand the significance of this expression, it is necessary to compute the *principal symbol* q_1 of Q, a smooth function on the punctured cotangent bundle $T^*(\mathbf{R}^5)\backslash\{0\}$, homogeneous in the fiber variables ζ, ξ', η'. In the simplest case, in which each source or receiver point is connected by a unique ray to any point in the domain of $\delta\bar{m}$, a computation of the type pioneered by Beylkin (1985), Rakesh (1988) yields

$$q_1[m,\delta m](x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta') = i\zeta(DT[m,\delta m](\mathbf{x},\mathbf{x}'_r(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta'))$$

$$+DT[m,\delta m](\mathbf{y},\mathbf{x}'_s(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta'))). \tag{5}$$

In (5), DT is the perturbation of geometrical-optics traveltime along the ray (assumed unique) between $\mathbf{x}$ and $(z_r,\mathbf{x}'_r)$ or $(z_s,\mathbf{x}'_s)$, with $\mathbf{x}'_r$ and $\mathbf{x}'_s$ defined as follows (see ten Kroode (2012) for more on this construction).

Denote by $\mathbf{X}'_r(\mathbf{x},\hat{\xi})$ the horizontal coordinates of the intersection, if any, with $x_1 = z_r$ of the ray of geometric optics with initial data $(\mathbf{x},\hat{\xi})$. Define $\mathbf{X}'_s(\mathbf{x},\hat{\xi})$ similarly. The notation reminds the reader that the ray, hence the location and time of surface arrival, depends only on the unit vector $\hat{\xi}$ of ξ. For an open conic set of $(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta')$, there exist unique data $(\mathbf{x},\xi(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta'))$ and $(\mathbf{y},\eta(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta'))$ so that

1. $\zeta = \xi_1 + \eta_1$
2. $m(\mathbf{x})|\xi|^2 = m(\mathbf{y})|\eta|^2$
3. $\xi_1, \eta_1 > 0$
4. $\xi_i = \xi'_i, \eta_i = \eta'_i$ for $i > 1$

Then

$$\begin{array}{rcl} \mathbf{x}'_r(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta') & = & \mathbf{X}_r(\mathbf{x},\hat{\xi}(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta')) \\ \mathbf{x}'_s(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta') & = & \mathbf{X}_s(\mathbf{y},\hat{\eta}(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta')) \end{array} \tag{6}$$

The symbol of the "annihilator" A is simply $a_0(x_1,\mathbf{x}',\mathbf{y}',\zeta,\xi',\eta') = \mathbf{x}' - \mathbf{y}'$. According to the calculus of pseudodifferential operators (for instance Taylor (1981)), the principal symbol of $[A,Q]$ is

$$-i\{a_0,q_1\} = -i(\nabla_{\xi'} - \nabla_{\eta'})q_1$$

$$= \nabla_{\mathbf{x}'_r} DT_r \cdot \nabla_{\hat{\xi}}\mathbf{X}_r \cdot \zeta(\nabla_{\xi'} - \nabla_{\eta'})\hat{\xi}$$

$$+\nabla_{\mathbf{x}'_s} DT_s \cdot \nabla_{\hat{\eta}}\mathbf{X}_s \cdot \zeta(\nabla_{\xi'} - \nabla_{\eta'})\hat{\eta} \tag{7}$$

in which the subscripts on DT signify the collections of arguments at which it is to be evaluated, per (5).

It follows from the calculus, in particular from Gårding's inequality, that for sufficiently oscillatory $\delta\bar{m}$ - that is, for a sufficiently dramatic separation of scales - the Hessian form is positive definite on subspaces of δm for which $\{a_0, q_1\}$ is nonvanishing in the support of $\delta\bar{m}$. The expression (7) suggests the meaning of this condition. The first factor in each of the two summands can be interpreted as the *slope* of arrival time curves - *all* of them, not just a few that are picked for analysis. The second factor in each case is proportional to a geometric amplitude (see for example Zhang et al. (2003)). This factor *must* be present: if the amplitudes are zero, then the presence or absence of reflectors (that is, support of $\delta\bar{m}$) cannot furnish the data with information about kinematics. The third factor is a real symbol of order zero, as the derivatives of the unit vector are homogeneous of order -1.

CONCLUSION

Apparently, under certain circumstances, the WEMVA objective function described in this paper is sensitive to the same aspects of the model m as is slope tomography (Billette and Lambaré, 1998; Chauris and Noble, 2001), but without requiring the picking of travel times. Picking is implicitly accomplished by data weighting, via the presence of the perturbational model $\delta\bar{m}$ and geometric amplitude in the Hessian kernel. The explicit symbol computation (5), (7) suggest that a detailed analysis of the Hessian singular spectrum may be feasible.

The computations presented here relied on strict ray geometric assumptions, which may to some extent be relaxed - see ten Kroode (2012) for a closely related discussion.

ACKNOWLEDGEMENTS

The author thanks Fons ten Kroode, Christiaan Stolk, and Jie Hou for many insights and helpful disucssions. The author acknowledges the generous long-tem support provided by the sponsors of The Rice Inversion Project. A version of this paper appeared in the Proceedings of the the conference on Inverse Problems: from Theory to Application, held in Bristol, UK, August 2014 under the auspices of the Institute of Physics.

REFERENCES

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.

Billette, F., and G. Lambaré, 1998, Velocity macro-model estimation from seismic reflection data by stereotomography: Geophysical Journal International, **135**, 671–680.

Blazek, K., C. C. Stolk, and W. Symes, 2013, A mathematical framework for inverse wave problems in heterogeneous media: Inverse Problems, **29**, 065001:1–34.

Chauris, H., and M. Noble, 2001, Two-dimensional velocity macro model estimation from seismic reflection data by local differential semblance optimization: applications synthetic and real data sets: Geophysical Journal International, **144**, 14–26.

Kern, M., and W. Symes, 1994, Inversion of reflection seismograms by differential semblance analysis: Algorithm structure and synthetic examples: Geophysical Prospecting, **99**, 565–614.

Plessix, R.-E., G. Baeten, J. W. de Maag, M. Klaassen, R. Zhang, and Z. Tao, 2010, Application of acoustic full waveform inversion to a low-frequency large-offset land data set: 81st Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 930–934.

Rakesh, 1988, A linearized inverse problem for the wave equation: Communications on Partial Differential Equations, **13**, 573–601.

Stolk, C. C., 2000, On the modeling and inversion of seismic data: PhD thesis, Universiteit Utrecht.

Stolk, C. C., M. V. de Hoop, and W. Symes, 2009, Kinematics of shot-geophone migration: Geophysics, **74**, WCA18–WCA34.

Symes, W., 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790.

———, 2009, The seismic reflection inverse problem: Inverse Problems, **25**, 123008:1–24.

Taylor, M., 1981, Pseudodifferential Operators: Princeton University Press.

ten Kroode, F., 2012, A wave-equation-based Kirchhoff operator: Inverse Problems, 115013:1–28.

Vigh, D., W. Starr, J. Kapoor, and H. Li, 2010, 3d full waveform inversion on a Gulf of Mexico WAZ data set: 81st Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 957–961.

Virieux, J., and S. Operto, 2009, An overview of full waveform inversion in exploration geophysics: Geophysics, **74**, WCC127–WCC152.

Zhang, Y., G. Zhang, and N. Bleistein, 2003, True amplitude wave equation migration arising from true amplitude one-way wave equations: Inverse Problems, **19**, 1113–1138.

An Approximate Inverse to the Extended Born Modeling Operator

Jie Hou and William W. Symes

ABSTRACT

Given a correct (data-consistent) velocity model, reverse time migration (RTM) correctly positions reflectors but generally with incorrect amplitudes and wavelet. Iterative least squares migration corrects amplitude and wavelet by fitting data in the sense of Born modeling, that is, replacing migration by Born inversion. However, least squares migration also requires a correct velocity model, and may require many migration/demigration cycles. In this paper, we modify RTM in the subsurface offset domain to create an asymptotic (high-frequency) approximation to extended least squares migration. This extended Born inversion operator outputs extended reflectors (depending on subsurface offset as well as position in the earth) with correct amplitude and phase, in the sense that similarly extended Born modeling reproduces the data to good accuracy. While the theoretical justification of the inversion property relies on ray-tracing and stationary phase, application of the weight operators does not require any computational ray-tracing at all. The computation expense of the extended Born inversion operator is roughly the same as that of extended RTM, and the inversion (data fit) property holds even when the velocity is substantially incorrect. The approximate inverse operator differs from extended RTM only in the application of data- and model-domain weight operators, and takes the form of an adjoint in the sense of weighted inner products in data and model space. Since the Born modeling operator is approximately unitary with respect to the weighted inner products, a weighted version of conjugate gradient iteration dramatically accelerates the convergence of extended least squares migration. An approximate least squares migration may be extracted from the approximate extended least squares migration by averaging over subsurface offset.

INTRODUCTION

Reverse time migration (RTM) (Baysal et al., 1983; Loewenthal and Mufti, 1983; Whitmore, 1983) produces kinematically accurate short-scale reflectivity, with reflectors positioned as accurately as migration velocity permits. RTM images may be degraded by amplitude anomalies, low-frequency noise, and wavelet side-lobes (Bednar and Bednar, 2006; Mulder and Plessix, 2004). Least squares migration (LSM)

(Nemeth et al., 1999; Dutta et al., 2014) compensates for all of these defects to some extent. LSM is actually a linearized inversion (Bourgeois et al., 1989), that is, a method for choosing short-scale reflectivity as a perturbation of the migration velocity model so as to achieve a sample-by-sample mean square best-fit to data via Born modeling. It has been formulated as an iterative process involving repeated migrations and demigrations (Nemeth et al., 1999; Dutta et al., 2014), in asymptotic approximation via Generalized Radon Transform (GRT) inversion (Beylkin, 1985; Bleistein, 1987; De Hoop and Bleistein, 1997), and as a true amplitude modification of wave-equation migration, using both one-way (Zhang et al., 2005) and two-way (Zhang et al., 2007) propagators.

In the following pages we describe a modification of RTM that approximately inverts the space shift extension of Born forward modeling (Rickett and Sava, 2002; Symes, 2008). That is, it produces an image volume depending not just on position in the subsurface but also on a subsurface offset parameter (vector, in 3D). By analogy with the usual terminology, one might term this approximate linearized inversion an extended LSM. Averaging the output over offset with a weight equal to one at zero offset produces an approximate inversion of the ordinary Born modeling operator, that is, an approximate LSM, provided that the velocity model is consistent with the data.

A concise mathematical description of this approximate inversion operator takes the form

$$\bar{F}^{\dagger}[v_0] = W_{\text{model}}[v_0]\bar{F}^{*}[v_0]W_{\text{data}}[v_0]. \tag{1}$$

In this formula,

- $\bar{F}^{\dagger}[v_0]$ is the approximate inverse of the extended Born modeling operator $\bar{F}[v_0]$ at background velocity v_0;
- $\bar{F}^{*}[v_0]$ is the extended RTM operator, that is, the adjoint or transpose of the modeling operator $\bar{F}[v_0]$
- $W_{\text{model}}[v_0]$ and $W_{\text{data}}[v_0]$ are model- and data-domain weighting operators, defined in detail in the Theory section.

The central result of this paper is that an approximate inverse in the sense of high frequency asymptotics takes this form, with weighting operators having *explicitly computable* integral forms, depending only on phase space variables and the velocity field. The derivation of this remarkable fact is sketched in the Theory section and recounted in detail in the appendices. It is a feature of space-shift extended modeling, and has no precise analogue for ordinary Born modeling.

ten Kroode (2012) described the construction of such an operator for 3D extended Kirchhoff modeling (that is, modeling reflections from interfaces, rather than from perturbations in the material parameter fields). Our work is directly inspired by his. The construction described in ten Kroode (2012) approximately inverts both a space-shift extended Kirchhoff operator and an angle-domain analogue, via the Radon transform relation between space-shift and scattering angle described first by Sava and Fomel (2003). Ten Kroode suggests that an inverse is possible in the *unfocused* case, in which image energy is spread over subsurface offset (or nontrivially dependent on scattering angle). Zhang and co-workers (Bleistein et al., 2005; Zhang et al., 2007; Zhang and Sun, 2008; Xu et al., 2011; Tang et al., 2013) have described an analogous construction of an inversion operator that produces accurate images and angle-domain extended gathers in the focused case. In all of these works (and in ours), geometric optics/acoustics (ray theory) justifies the inversion construction - but no ray theory constructions appear in the final result! The critical observation at the root of this remarkable fact is due to Bleistein et al. (2005): the Hessian ("Beylkin") determinants arising in stationary phase approximation of the normal operator $F^*[v_0]F[v_0]$ factor into reciprocal geometric amplitudes and other, ray-independent, terms. These geometric amplitude factors *cancel* the geometric amplitudes present in the propagating fields. The remaining expressions are free of ray-theoretic quantities.

Our results differ in several respects from those cited in the last paragraph. Unlike ten Kroode (2012), we base our construction on Born (rather than Kirchhoff) modeling, to produce an approximate extended LSM operator. The underlying mechanism of the inverse construction is somewhat clearer in this case. Also, just as Born modeling is the linearization of the full-waveform modeling operator, extended Born modeling is the linearization of an extended version of full-waveform modeling (Symes, 2008), and the approximate inverse may be useful in accelerating gradient-based, image-domain full waveform inversion algorithms (Sun and Symes, 2012; Biondi and Almomin, 2014). While most of the cited work concerns the scattering angle extension, the link between the angle domain and the subsurface offset domain is simply a transform (Sava and Fomel, 2003). There being no intrinsic additional information content in the angle representation, we present our results exclusively in terms of subsurface offset. Unlike much previous work (for instance, (Xu et al., 2011), see discussion surrounding equation (10)), we do not assume that the background velocity v_0 is kinematically correct. Like the Kirchhoff inversion of ten Kroode (2012), our Born inversion operator produces extended models with accurate amplitudes even when the image volume is unfocused. Such accurate extended inversion is critical for the success of image-domain velocity updating schemes (Kern and Symes, 1994; Liu et al., 2013, 2014; Lameloise et al., 2014).

We give implementation details and numerical illustrations for the 2D constant-density acoustics version of an approximate inversion taking the form given in equation (1). Our work seems to be the first to confirm explicitly, by numerical example, that this operator is actually an inversion of the Born modeling operator $F[v_0]$: that is, that the output of the inversion operator, input to the modeling operator, reproduces the data with reasonable accuracy.

The form of the approximate inverse (1), with symmetric positive definite weight operators, also seems to be new, and has a remarkable implication: for norms in model and data spaces given by the weight operators, $\bar{F}[v_0]^\dagger$ is the adjoint of $\bar{F}[v_0]$. That is, the extended modeling operator is approximately unitary with this choice of norms. The authors have verified the consequent drastic convergence acceleration for extended Least Squares Migration via Krylov subspace iteration. These results will be reported elsewhere.

As other authors have suggested, a 3D approximate inversion operator may be written in precisely the same form (equation (1)), with a virtually identical derivation. However, explanation of the theory is somewhat simpler in 2D, the computation implementation is a good deal simpler, examples are less demanding, and results are easier to present.

We end this overview with two caveats. First, we have considered only very idealized acquisition geometry (and that in 2D): we have neglected the implications of coarse sampling, more complex source receiver geometry such as OBS recording, broadband technology, and availability of direct measurements of quantities other than pressure for formulation of RTM-based approximate inversion operators. Others have addressed some of these issues (Tang et al., 2013); some or all will arise in any practical application. Second, while our derivation produces an inverse regardless of focusing, the model-domain weight operator $W_{\mathrm{model}}[v_0]$ simplifies greatly in the focused case (or equally well for laterally homogeneous velocities), and the examples presented here are limited to these special cases. The Discussion section describes the additional steps required to implement full image-volume inversion.

The next section describes space-shift linearized modeling and migration, and the construction of the approximate inverse operator. The following section presents several 2D examples and illustrates the features of the approximate inverse mentioned above. We follow the examples with a brief discussion of various possibilities for modification or extension of these results.

THEORY

In this section, we will first review the concepts of extended Born modeling operator, its adjoint operator and their high frequency approximations. We will then modify the adjoint operator into an approximate inverse operator. Finally, we will discuss implementation details for the inverse operator.

Extended Born Modeling Operator and its Adjoint

The 2D constant density acoustic wave equation with causal initial condition is :

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 u}{\partial t^2}(\mathbf{x},t) - \nabla^2 u(\mathbf{x},t) = f(t,\mathbf{x},\mathbf{x_s}); \qquad u(\mathbf{x},t) \equiv 0, t \ll 0 \tag{2}$$

Here $\mathbf{x}$ denotes position within a model of the Earth, $v(\mathbf{x})$ is the acoustic velocity, $u(\mathbf{x},t)$ is the acoustic potential, and $f(t,\mathbf{x},\mathbf{x_s})$ is the source term. We assume throughout this paper that v is constant in the half-space $z < 0$, that is, that $z = 0$ is an absorbing surface.

In the Born approximation, the velocity model is split into a smooth part v_0 and a singular or oscillatory part δv:

$$v(\mathbf{x}) = v_0(\mathbf{x}) + \delta v(\mathbf{x}) \tag{3}$$

These two parts respectively correspond to a smooth long-wavelength, large scale background model, which will not produce reflections, and a short-wavelength, small scale perturbation model, which contains all the high resolution features.

Of course, one can construct a perturbation expansion for any decomposition of v into two summands, however the corresponding perturbation approximation to the acoustic field is most accurate when the scales are separated, that is, δv has very small mean over distances on which v_0 varies significantly (Symes, 2009).

The first order perturbation in the acoustic potential field δu corresponding to δv may be expressed in terms of the causal Green's function $G(\mathbf{x},\mathbf{y},t)$ for a given background model v_0. Restricting δu to the source and receiver positions $\mathbf{x_s},\mathbf{x_r}$ results in an integral operator expression for the Born modeling operator $F[v_0]$:

$$(F[v_0]\delta v)(\mathbf{x_s},\mathbf{x_r},t) = \frac{\partial^2}{\partial t^2}\int d\mathbf{x}d\tau G(\mathbf{x_s},\mathbf{x},\tau)\frac{2\delta v(\mathbf{x})}{v_0(\mathbf{x})^3}G(\mathbf{x},\mathbf{x_r},t-\tau) \tag{4}$$

The adjoint operator $F[v_0]^*$ is the operator implemented by one common variant of Reverse Time Migration: it is applied to a set of data trace perturbations $\delta d(\mathbf{x}_s,\mathbf{x}_r,t)$ by

$$(F^*[v_0]\delta d)(\mathbf{x}) = \frac{2}{v_0(\mathbf{x})^3}\int d\mathbf{x_s}d\mathbf{x_r}dtd\tau G(\mathbf{x_s},\mathbf{x},\tau)G(\mathbf{x},\mathbf{x_r},t-\tau)\frac{\partial^2}{\partial t^2}\delta d(\mathbf{x_s},\mathbf{x_r},t) \tag{5}$$

An appropriate version of subsurface offset extended Born modeling introduces dependence of δv (but not v_0) on an additional parameter, $\mathbf{h}$, essentially the offset between sunken source and sunken receiver in Claerbout's survey-sinking imaging condition (Claerbout (1985), Symes (2008), Stolk et al. (2009b)). In terms of Green's functions, the subsurface extended Born Modeling Operator and its adjoint (applied to a data perturbation δd) are :

$$(\bar{F}[v_0]\delta\bar{v})(\mathbf{x_s},\mathbf{x_r},t) = \frac{\partial^2}{\partial t^2}\int d\mathbf{x}d\mathbf{h}d\tau G(\mathbf{x_s},\mathbf{x}-\mathbf{h},\tau)\frac{2\delta\bar{v}(\mathbf{x},\mathbf{h})}{v_0(\mathbf{x})^3}G(\mathbf{x}+\mathbf{h},\mathbf{x_r},t-\tau) \quad (6)$$

$$(\bar{F}^*[v_0]\delta d)(\mathbf{x},\mathbf{h}) = \frac{2}{v_0(\mathbf{x})^3}\int d\mathbf{x_s}d\mathbf{x_r}dtd\tau G(\mathbf{x_s},\mathbf{x}-\mathbf{h},\tau)G(\mathbf{x}+\mathbf{h},\mathbf{x_r},t-\tau)\frac{\partial^2}{\partial t^2}\delta d(\mathbf{x_s},\mathbf{x_r},t) \quad (7)$$

In Claerbout's original conception, the subsurface offset $\mathbf{h}$ is horizontal. ten Kroode also adopts this convention, and we follow it here. Thus we write h rather than $\mathbf{h}$ for the (scalar) horizontal subsurface offset in 2D.

High Frequency Approximation

The progressing wave approximation (Courant and Hilbert (1962)) of the Green's function is

$$G(\mathbf{x_s},\mathbf{x},t) \simeq a(\mathbf{x_s},\mathbf{x})S(t-T(\mathbf{x_s},\mathbf{x})). \quad (8)$$

In equation (8), the amplitude $a(\mathbf{x_s},\mathbf{x})$ and the travel time $T(\mathbf{x_s},\mathbf{x})$ solve the transport and eikonal equation respectively, and $S(t)$ is a singular, causal waveform, the choice of which depends on the space dimension. The approximation (8) is only valid locally, between the source point and the nearest caustic or conjugate (multipath) point. The conclusions we draw below are valid more globally, however, provided that the Traveltime Injectivity Condition holds: a two-way traveltime along a reflected ray pair determines the one-way traveltimes of source and receiver rays. ten Kroode (2012) gives a detailed justification for the global validity of similar conclusions in the 3D case. We confine ourselves in this paper to numerical evidence for global 2D results.

In 2D case, the leading singularity is proportional to the generalized function $S(t) = t_+^{-1/2} = t^{-1/2}H(t)$. Replacing the Green's function by the progressing wave approximation (8) in the expression (6) for the extended Born modeling operator and using the identity (Gel'fand and Shilov (1958)),

$$t_+^{-1/2} * t_+^{-1/2} = (\Gamma(\frac{1}{2}))^2H(t) = \pi H(t) \quad (9)$$

we obtain

$$(\bar{F}[v_0]\delta\bar{v})(\mathbf{x_s},\mathbf{x_r},t) \simeq \frac{\partial}{\partial t}\int d\mathbf{x}dha_sa_r\delta(t-T_s-T_r)\frac{2\pi\delta\bar{v}(\mathbf{x},h)}{v_0(\mathbf{x})^3} \quad (10)$$

in which we have denoted amplitudes $a(\mathbf{x_s},\mathbf{x}-h), a(\mathbf{x}+h,\mathbf{x_r})$ as a_s, a_r and traveltime $T(\mathbf{x_s},\mathbf{x}-h), T(\mathbf{x}+h,\mathbf{x_r})$ as T_s, T_r. We can also give the same treatment to the migration operator :

$$(\bar{F}[v_0]^*\delta d)(\mathbf{x},h) \simeq \frac{2\pi}{v_0(\mathbf{x})^3}\int d\mathbf{x_s}d\mathbf{x_r}a_s a_r \frac{\partial}{\partial t}\delta d(\mathbf{x_s},\mathbf{x_r},T_s+T_r) \tag{11}$$

Asymptotic Inverse Operator

The derivation of the inverse operator starts from understanding the high frequency leading order behavior of the normal operator $\bar{F}^*\bar{F}$. Both a bit of foresight and simplification of the computations suggest examining the modified normal operator $(I_t\bar{F})^*(I_t\bar{F})$ instead, with I_t the causal indefinite time integration operator, inverse to $\partial/\partial t$. Combining equations (10) and (11) yields an integral representation for $(I_t\bar{F})^*(I_t\bar{F})$. Appendix A sketches a lengthy but standard computation based on the Principle of Stationary Phase, showing that this integral representation has an asymptotic (high frequency, short scale) approximation in the form of an oscillatory integral:

$$\begin{aligned}(I_t\bar{F})^*(I_t\bar{F})\delta\bar{v}(x,z,h) \approx & -\int dk_x dk_z dk_h e^{i(k_x x+k_h h+k_z z)}\widehat{\delta v}(k_x,k_h,k_z)\\ & \times\left[\frac{2\pi v_0^{-5}}{k_{xz}k_{hz}}Pa_r^2 a_s^2\left(\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r}\right)^{-1}\right]\end{aligned} \tag{12}$$

The integrand factor in brackets will be explained below; if it were identically = 1, then up to nuisance factors the right-hand side would recover the velocity perturbation $\delta\bar{v}$, accurately at short length scales.

The remainder of this section explains how the factor in brackets simplifies, all ray-trace dependent quantities cancel, additional filtering and scaling operators applied to data (input) and model (output) lead to an approximate identity, and precisely how it comes to have the form of the main result, equation (1).

The integrand in equation (12) is a function of the extended phase space variables (x,z,h,k_x,k_z,k_h). Several of its factors are related to source and receiver rays traced from $(x\pm h,z)$ to the surface. These include

- source and receiver geometric amplitudes (spreading factors) a_s and a_r;
- rates of change of receiver ray angle (with the vertical) α_r with respect to receiver coordinate x_r;
- rates of change of source ray angle (with the vertical) α_s with respect to source coordinate x_s;

The initial slowness vectors $\mathbf{k}_s$ of the source ray (at $(x-h,z)$) and $\mathbf{k}_r$ of the receiver ray (at $(x+h,z)$) solve the system of equations

$$\begin{aligned} k_x &= \frac{k_x^r + k_x^s}{2} \\ k_h &= \frac{k_x^r - k_x^s}{2} \\ k_z &= \frac{k_z^r + k_z^s}{2} \\ v(x+h,z)^2((k_x^r)^2 + (k_z^r)^2) &= v(x-h,z)^2((k_x^s)^2 + (k_z^s)^2) \end{aligned} \tag{13}$$

The first three conditions in the system (13) follow from the stationary phase conditions (A-6), as explained in the Appendix A. The last condition expresses the equality of temporal frequency along source and receiver rays. These conditions together amount to a version of Snell's Law appropriate for space-shift extended modeling.

Rays with initial conditions $(x+h, z, k_x^r, k_z^r)$ and $(x-h, z, k_x^s, k_z^s)$ intersect source and receiver datums at points (x_s, z_s) and (x_r, z_r), thus making x_r, x_s functions of the extended phase space variables (x, z, h, k_x, k_z, k_h), and therefore also the geometric amplitudes a_r and a_s. The ray angles are as well, since for instance $\tan \alpha_r = k_x^r/k_z^r$, therefore so are their derivatives with respect to source and receiver coordinates.

The other factors in the bracketed integrand factor in (12) are *explicit, algebraic* functions of the phase variables, whose definition does not require ray tracing at all. The (x,z) and (h,z) wavenumbers are defined as

$$k_{xz} = (k_x^2 + k_z^2)^{\frac{1}{2}}, \; k_{hz} = (k_h^2 + k_z^2)^{\frac{1}{2}}. \tag{14}$$

The remaining factor P is homogeneous of degree zero in (k_x, k_z, k_h), and depends algebraicaly on these frequency variables and on $v(x-h,z), v(x,z)$ and $v(x+h,z)$. A full definition of P is given in Appendix A. For now, note that $P = 1$ when $h = 0$.

Next, we invoke the result of Appendix B, relating geometric amplitudes to angular rates of change:

$$\begin{aligned} a_r^2 &= \frac{1}{8\pi^2} \frac{v_r}{cos\theta_r} \frac{d\alpha_r}{dx_r} \\ a_s^2 &= \frac{1}{8\pi^2} \frac{v_s}{cos\theta_s} \frac{d\alpha_s}{dx_s} \end{aligned} \tag{15}$$

In these expressions, θ_r and θ_s are arrival angles of receiver and source rays at receiver and source. Since the receiver and source ray data are functions of the phase variables, so are θ_r and θ_s.

Remarkably, the amplitudes and angular rates of change in (12) *cancel* due to (15). This cancellation of both geometric amplitudes occurs only for extended modeling.

The analogous computation for single shot records, for example, leads to cancellation of the receiver ray amplitude a_r only. In that setting, inversion requires an additional operator, essentially Claerbout's division imaging condition (Stolk et al., 2009a), to compensate for the source amplitude field. For the horizontal offset extension, the additional integrations over the subsurface offset variable lead to an additional α_s derivative via stationary phase, thus canceling the source amplitude as well. See Xu et al. (2011) for a version of this argument using scattering angle rather than subsurface offset.

Taking advantage of this observation and simplifying, equation (12) becomes

$$\begin{aligned}(I_t\bar{F})^*(I_t\bar{F})\delta\bar{v}(x,z,h) \approx -\int dk_x dk_z dk_h e^{i(k_x x+k_h h+k_z z)}\widehat{\delta v}(k_x,k_h,k_z)\\ \times\left[\frac{v_0^{-5}}{32\pi^3 k_{xz}k_{hz}}P\frac{v_s}{\cos\theta_s}\frac{v_r}{\cos\theta_r}\right]\end{aligned} \tag{16}$$

The integrand on the right hand side still appears entangled with ray-theoretic constructions, namely the arrival angles θ_r, θ_s. To eliminate these, a further modification of the normal operator is necessary. The expression (10) for the modeling operator using asymptotic Green's functions implies that

$$\frac{\partial}{\partial z_r}I_t\bar{F}[v_0]\delta\bar{v}(\mathbf{x_s},\mathbf{x_r},t) \approx -\int d\mathbf{x}dha_s a_r\frac{\partial T_r}{\partial z_r}\frac{\partial\delta}{\partial t}(t-T_s-T_r)\frac{2\pi\delta\bar{v}(\mathbf{x},h)}{v_0(\mathbf{x})^3}$$

$$= -\frac{\partial}{\partial t}\int d\mathbf{x}dha_s a_r\frac{\partial T_r}{\partial z_r}\delta(t-T_s-T_r)\frac{2\pi\delta\bar{v}(\mathbf{x},h)}{v_0(\mathbf{x})^3} \tag{17}$$

From the eikonal equation,

$$\frac{\partial T_r}{\partial z_r} = -\frac{\cos\theta_r}{v_r} \tag{18}$$

Combine equations (17) and (18), use the abbreviations $D_{z_r} = \partial/\partial z_r$, and cancel the two time derivatives with time integrations to obtain

$$I_t D_{z_r} I_t\bar{F}[v_0]\delta\bar{v}(\mathbf{x_s},\mathbf{x_r},t) \simeq \int d\mathbf{x}dha_s a_r\frac{\cos\theta_r}{v_r}\delta(t-T_s-T_r)\frac{2\pi\delta\bar{v}(\mathbf{x},h)}{v_0(\mathbf{x})^3} \tag{19}$$

Precisely the same manipulations with the source-related quantities lead to

$$I_t D_{z_s} I_t D_{z_r} I_t\bar{F}[v_0]\delta\bar{v}(\mathbf{x_s},\mathbf{x_r},t) \approx \int d\mathbf{x}dha_s a_r\frac{\cos\theta_r}{v_r}\frac{\cos\theta_s}{v_s}\delta(t-T_s-T_r)\frac{2\pi\delta\bar{v}(\mathbf{x},h)}{v_0(\mathbf{x})^3} \tag{20}$$

The right-hand side of equation (20) defines an operator differing from $I_t\bar{F}[v_0]$ only in having additional receiver- and source-dependent factors multiplying the amplitudes. Therefore the stationary phase computations in Appendix A combined with the amplitude-angle relations of Appendix B lead to an asymptotic approximation similar to (16):

$$(I_t\bar{F})^*(I_tD_{z_s}I_tD_{z_r}I_t\bar{F})\delta\bar{v}(x,z,h) \approx -\int dk_x dk_z dk_h e^{i(k_x x+k_h h+k_z z)}\widehat{\delta v}(k_x,k_h,k_z) \times \left[\frac{v_0^{-5}}{32\pi^3 k_{xz}k_{hz}}P\right] \tag{21}$$

The source wavefield is downgoing, the receiver wavefield upcoming (again, we assume a homogeneous velocity in $z < \max(z_s, z_r)$). Therefore if $d = \bar{F}[v_0]\delta\bar{v}$ is in the range of the extended Born modeling operator,

$$I_tD_{z_r}I_td(x_r,t;x_s) = \frac{1}{8\pi^3}\int dk_s dk_r d\omega \exp(i(k_s x_s + k_r x_r + \omega t))\left(+\sqrt{\frac{1}{v_0^2}-\left(\frac{k_r}{\omega}\right)^2}\right)\widehat{I_td}(k_r,\omega;k_s). \tag{22}$$

$$I_tD_{z_s}I_td(x_r,t;x_s) = \frac{1}{8\pi^3}\int dk_s dk_r d\omega \exp(i(k_s x_s + k_r x_r + \omega t))\left(-\sqrt{\frac{1}{v_0^2}-\left(\frac{k_s}{\omega}\right)^2}\right)\widehat{I_td}(k_r,\omega;k_s). \tag{23}$$

(this observation is due to ten Kroode (2012)). It follows from (22), (23) that the operator $-(I_t)^*(I_tD_{z_s})(I_tD_{z_r})I_t$ has the same effect on data output by the forward map $\bar{F}[v_0]$ as the positive definite symmetric operator $W_{\rm data}[v_0]$, defined by

$$W_{\rm data}[v_0]d(x_r,t;x_s) = (I_t)^*\frac{1}{8\pi^3}\int dk_s dk_r d\omega \times \exp(i(k_s x_s+k_r x_r+\omega t))\left(\sqrt{\frac{1}{v_0^2}-\left(\frac{k_r}{\omega}\right)^2}\right)\left(\sqrt{\frac{1}{v_0^2}-\left(\frac{k_s}{\omega}\right)^2}\right)(\widehat{I_td})(k_r,\omega;k_s). \tag{24}$$

Explicitly,

$$W_{\rm data}[v_0]\bar{F}[v_0] = -(I_t)^*I_tD_{z_s}I_tD_{z_r}I_t\bar{F}[v_0]. \tag{25}$$

Note that $W_{\rm data}[v_0]$ depends only on the values of v_0 near the sources and receivers: it is completely independent of the behaviour of v_0 for $z > \max(z_r, z_s)$.

Oscillatory integral operators of the type appearing on the right-hand side of equation (21) have come to be called *pseudodifferential*, and have a number of important properties, of which we must use several. For example, the product of two such operators is another such: the Fourier representation amplitude (or *symbol*) of the product is the product of the symbols of the operator factors, up to an asymptotically negligible error. It follows that such operators (on scalar functions) commute up to an asymptotically negligible error. An operator with nonvanishing symbol is asymptotically invertible, and the symbol of the inverse is the reciprocal of the symbol. See for instance Taylor (1981) for an account of the calculus of pseudodifferential operators.

It follows that we can write the right-hand side of equation (21) as

$$\approx (W_{\rm model}[v_0]^{-1}\delta\bar{v})(x,z,h) \tag{26}$$

in which

$$\begin{aligned} W_{\rm model}[v_0]^{-1}u(x,z,h) =& \frac{1}{8\pi^3}\int dk_x dk_h dk_z \, \exp(i(k_x x + k_h h + k_z z))\frac{v_0(x,z)^{-5}}{4k_{xz}k_{hz}} \\ & \times P(x,z,h,k_x,k_z,k_h)\hat{u}(k_x,k_z,k_h). \end{aligned} \tag{27}$$

As follows from the facts mentioned above,

$$\begin{aligned} W_{\rm model}[v_0]u(x,z,h) =& \frac{1}{8\pi^3}\int dk_x dk_h dk_z \, \exp(i(k_x x + k_h h + k_z z))\frac{4v_0(x,z)^{5}k_{xz}k_{hz}}{P(x,z,h,k_x,k_z,k_h)} \\ & \times \hat{u}(k_x,k_z,k_h). \end{aligned} \tag{28}$$

Combining equations (21), (25), and (26) establishes the main conclusion of our paper, equation (1), with $W_{\rm model}$ defined in equation (28) and $W_{\rm data}$ defined in equation (24).

We end this section by describing how the approximate inverse operator (1) defines an approximate (non-extended) least squares migration. Given a velocity perturbation $\delta v(x,z)$, the corresponding extended model is

$$\delta\bar{v}(x,z,h) = \delta v(x,z)\delta(h),$$

Let

$$\delta d = F[v_0]\delta v$$

be the corresponding Born data. Then

$$\delta\bar{v} = W_{\rm model}[v_0]\bar{F}[v_0]^* W_{\rm data}[v_0]\delta d,$$

whence

$$\delta v(x,z) = \int dh \phi(h)(W_{\rm model}[v_0]\bar{F}[v_0]^* W_{\rm data}[v_0]\delta d)(x,z,h) \tag{29}$$

for any weight function $\phi(h)$ satisfying $\phi(0) = 1$.

The arbitrariness of the weight function $\phi(h)$, subject only to the constraint $\phi(0) = 1$, might seem strange. Viewing the formula (29) in terms of the related angle-domain image volume, as in (Sava and Fomel, 2003), suggests an alternate meaning for this formula. Via the Radon transform, a weighted average over offset is equivalent to a weighted average over angles of the corresponding angle-domain volume. A weight function $\phi(h)$ spread uniformly over the offset range, as we have used in the examples of the next section, corresponds to an angle domain weight function concentrated near zero (scattering angle). For well-focused noise-free data, inversion using a small range of scattering angles should be reasonably accurate, as indeed the examples presented in the next section attest. On the other hand, a choice of $\phi(h)$ concentrated near $h = 0$ would correspond to estimating δv as a stack over a wide range of scattering angles. One might expect the estimate so obtained to be less sensitive to incoherent or numerical noise.

Implementation Details

In the implementation of the asymptotic inverse operator, any suitable time- or frequency-domain method can be used to approximate $\bar{F}$ and $\bar{F}^*$. We have used a time-domain centered difference scheme of order 2 in time and 8 in space to solve the acoustic wave equation, and the well-known adjoint state method (Plessix, 2006) to approximate $\bar{F}^*$.

For $W_{\rm data}$, equation (25) shows that for model-consistent data, either the definition (24) in terms of one-way operators, or the equivalent expression in terms of source and receiver vertical derivatives (dipole source/receiver) produces the same result, at least in principle. In the numerical experiments reported in the next section, we have chosen the dipole approach. To avoid explicitly computing dipole responses, we have used a trick available for streamer geometry with free surface and relatively shallow tow depth: the ghost sources and receivers automatically supply scaled dipoles. Assuming the source tow depth to be z_s, the free surface Green's function $G_{\rm free}$ is related to the full space (absorbing boundary) Green's function G by

$$G_{\rm free}(x,z,t;x_s,z_s) = G(x,z,t;x_s,z_s) - G(x,z,t;x_s,-z_s) \approx 2z_sD_{z_s}G(x,z,t;x_s,0) \tag{30}$$

By reciprocity, a similar approximation applies to the receiver. In application, if $\bar{F}[v_0]$ is computed with absorbing boundary, then $\bar{F}[v_0]^*$ can be caculated with free surface, or vis-versa - in either case, with appropriate inclusion of I_t factors, an approximation to $W_{\rm data}$ for the absorbing surface problem ensues.

This approximation proved quite convenient and produced consistent results as the central finite different implementation (Hou and Symes, 2014) for a first round of numerical experiments. It might even be applied to actual streamer data with shallow and uniform tow depth. However it is is only good to perhaps half of the notch frequency, so strictly limits resolution. For more or less arbitrary but uniform source and receiver depths and sampling, the one-way propagator construction (24) would be preferable. As mentioned in the introduction, more sophisticated streamer geometry, OBS recording, and less favorable sampling all would require modification of even the 3D version of our computations.

Application of the weight operator $W_{\rm model}[v_0]$ in principle requires the evaluation of an oscillatory integral (28). However in two special cases, the factor P may be ignored: as established in Appendix A, $P = 1$ if either $h = 0$ or if v_0 is independent of x. The first case applies to approximate least-squares migration: if model and data are consistent, then image energy focuses at $h = 0$ and the values of P for nonzero h do not contribute to leading order in frequency (more technically, this is the *pseudolocal* property of $\bar{F}[v_0]^* W_{\rm data}[v_0]\bar{F}[v_0]$ and $W_{\rm model}[v_0]$: both are pseudodifferential, and the image of a physically consistent input is asymptotically negligible away from $h = 0$).

Accordingly, we have set $P = 1$ in our examples, which fall into one of these two classes. Then

$$W_{\rm model}[v_0] \sim 4v_0^5 L, \tag{31}$$

where L is the filter defined in the Fourier domain by $k_{xz}k_{hz}$: in other words,

$$L = \sqrt{-\nabla^2_{x,z}}\sqrt{-\nabla^2_{h,z}}. \tag{32}$$

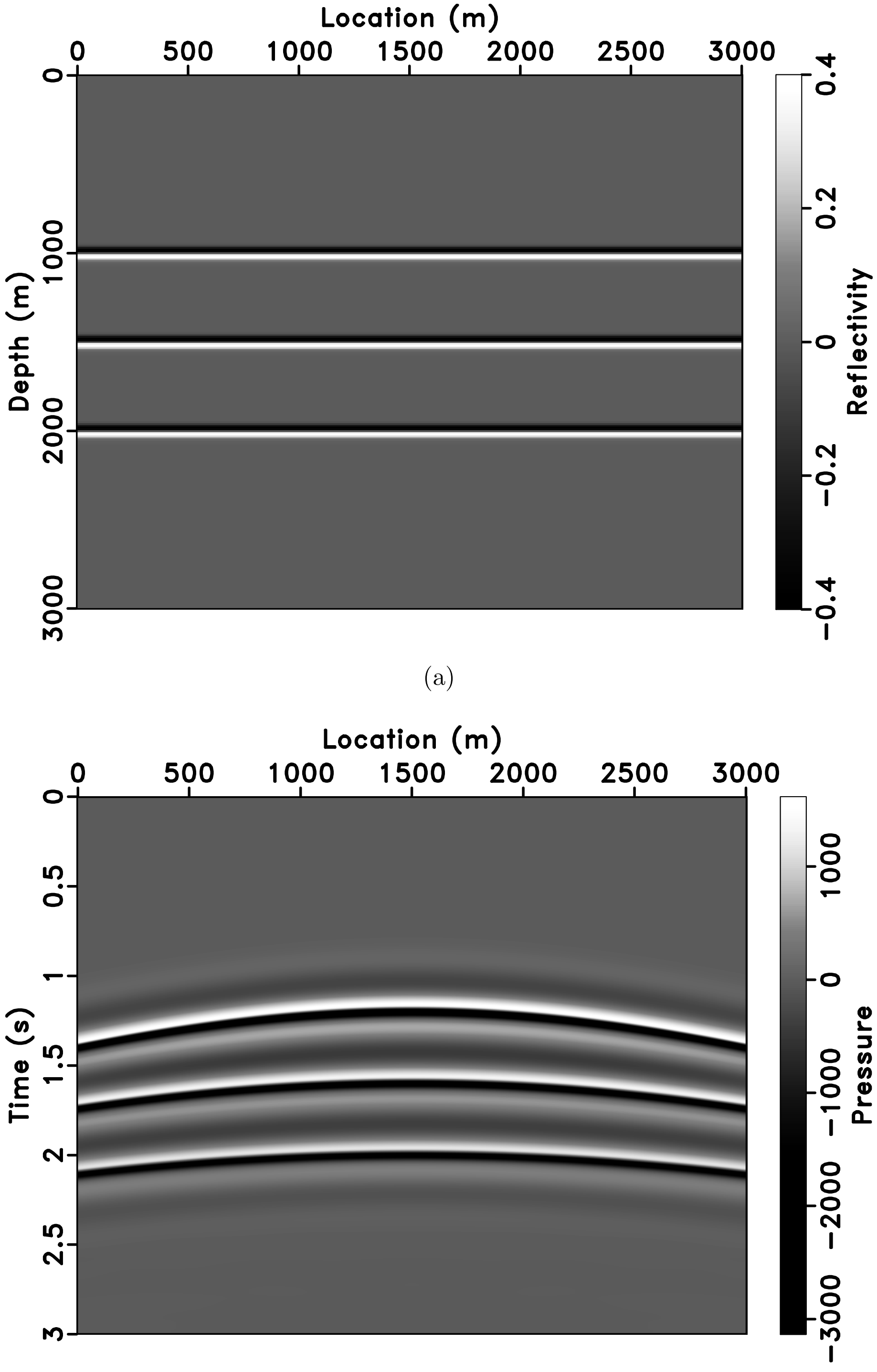
With these approximations to $W_{\rm data}$ and $W_{\rm model}$, the computational cost of the approximate inverse operator $\bar{F}[v_0]^\dagger$ very similar to that of the extended RTM operator $\bar{F}[v_0]^*$.

NUMERICAL EXAMPLES

In this section, we will use three numerical examples to illustrate the effectiveness of the inverse operator.

The first model, shown in Figure 1(a), combines three flat reflectors at $z = 1, 1.5, 2$ km with a constant (2500 m/s) background velocity model. The spatial sampling interval of the model is 10 m for both x and z axis. A (2.5-5-35-40) Hz bandpass wavelet with 1 ms time interval is used to simulate the Born data (2-8 Finite Difference Scheme). 76 shots are evenly spread on the surface ($z = 0$) every 40 m. All the shots will be recorded by 301 receivers deployed every 10 m on the surface.

One shot Born data ($x_s = 1500$ m) shown in Figure 1(b) is calculated using Equation (6). Both extended RTM (Equation (7)) and the new inverse operator (Equation (1)) are applied on the Born data. Comparing the migrated image (Figure 2(a)) and inverted reflectivity model (Figure 2(b)), we can clearly see the inverse operator can focus the energy much better than extended RTM. It has many least squares migration qualities, such as improved amplitudes, tighter wavelet. Thus, the inverse operator at least plays the role of space deconvolution. However, we can never recover the reflector perfectly due to the lack of the low frequency data. The inverted reflectivity model will not be a good standard for the effectiveness of the inverse operator. A good way to evaluate the inverse operator would be to compare the “observed” data of the true model (Figure 1(b)) and the resimulated data of inverted reflectivity model (Figures



(a)

(b)

Figure 1: (a) Reflectivity Model (δv) with a constant background model ($v_0 = 2500$ m/s) (b) One-shot ($x_s = 1500$ m) Simulated Born Data appinv/Fig/layer layermodel,layerdata

3(a)). The comparisons (Figure 3) show that the data resimulated from the inverted reflectivity model is almost same as the original data. The degree of approximation shows clearly in the comparison of middle traces of original and resimulated data, Figure 7(a).

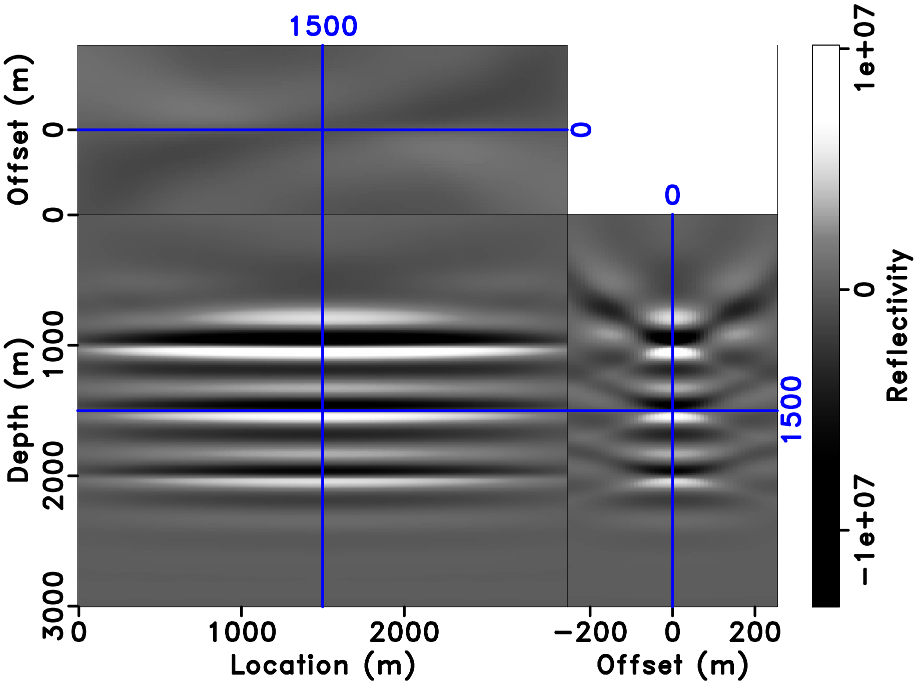
Another point worth noting is that the inverse operator is valid, that is, produces a data-fitting model, no matter whether the background velocity is correct or not. Same comparison between the extended RTM and the inverse operator has been carried out with incorrect background velocity model (90% of true velocity), see Figure 4. The comparison indicates the above analysis is true even in the presence of velocity error. The resimulation is displayed in Figure 5(a), which should be compared to Figure 1(b). The difference appears in Figure 5(b). Some divergence is unavoidable near the boundary as a result of the acquisition geometry. Apart from that, the resimulation matches the original Born data extremely well. We conclude that the operator defined in equation (1) is an accurate approximate inverse to the extended Born modeling operator, at least for data arising from physical (non-extended Born) modeling.

We call the process defined by equation (29) *non-extended approximate inversion*, for the purposes of this section. In our examples, we use the simplest choice of weight function, $\phi \equiv 1$, that is, stacking along the h-axis. Both the full volume (Figure 6) and middle trace comparisons in Figure 7(b) illustrate the precision of the non-extended approximate inversion, and the generally low frequency nature of the error.

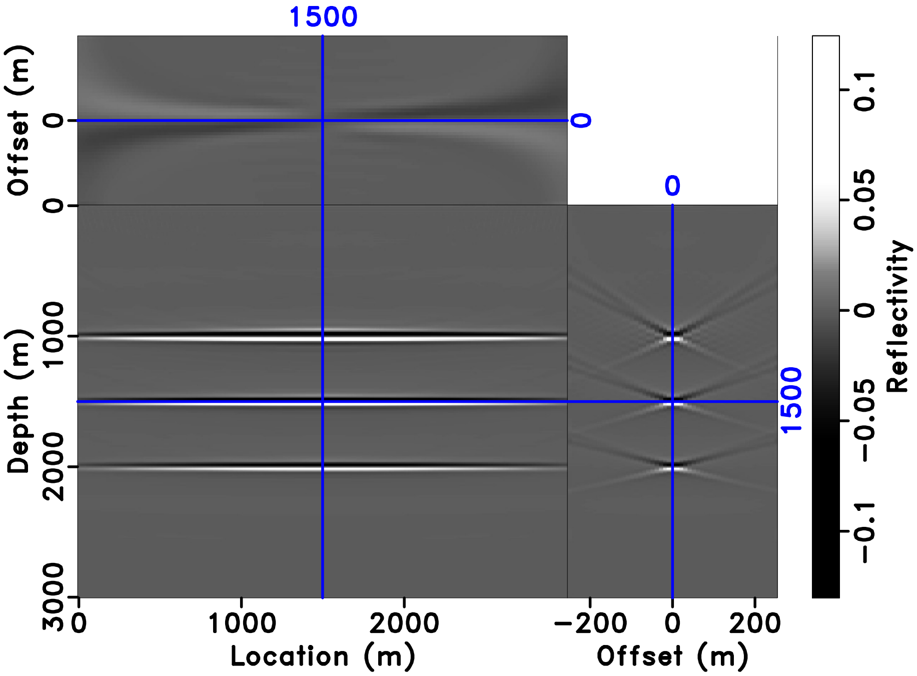
The simple geometrical optics computation of previous section will fail in the presence of caustics (or multipathing). The second example will show that the conclusions expressed by equations (1) and (29) are still valid even in the presence of multipathing. The background velocity model for the second example contains a low velocity Gaussian lens. A flat horizontal reflector is placed right below the lens at the depth of 2 km. This model is very similar to the one used by Nolan and Symes (1996) and Stolk and Symes (2004). The numerical implementation has the same configuration as the first example. Because of the Gaussian lens, the rays will certainly focus and form a triplication after going through the lens. The rays and wavefronts are shown in Figure 8(b). We can clearly see that this model produces multipathing and caustics.

The inverse operator defined in equation (1) produces the reflectivity model shown in Figure 9(a). From the image perspective, we clearly reproduce the flat reflector below the lens with no kinematic artifacts (Stolk and Symes, 2004), consistent with kinematic prediction in Stolk et al. (2009b). Resimulation with Born modeling operator from the inverted reflectivity model predicts data very close to the input data (Figure 10(a), 11(a)).

Finally, we apply our approximate inversion to Born data for the Marmousi Model. We smooth the velocity model as the background model and take the difference as the reflectivity model, shown in Figure 12(a) and 14(a). Born data for Marmousi model has 231 common shot gathers every 40 m and each shot has 921 receivers every 10 m

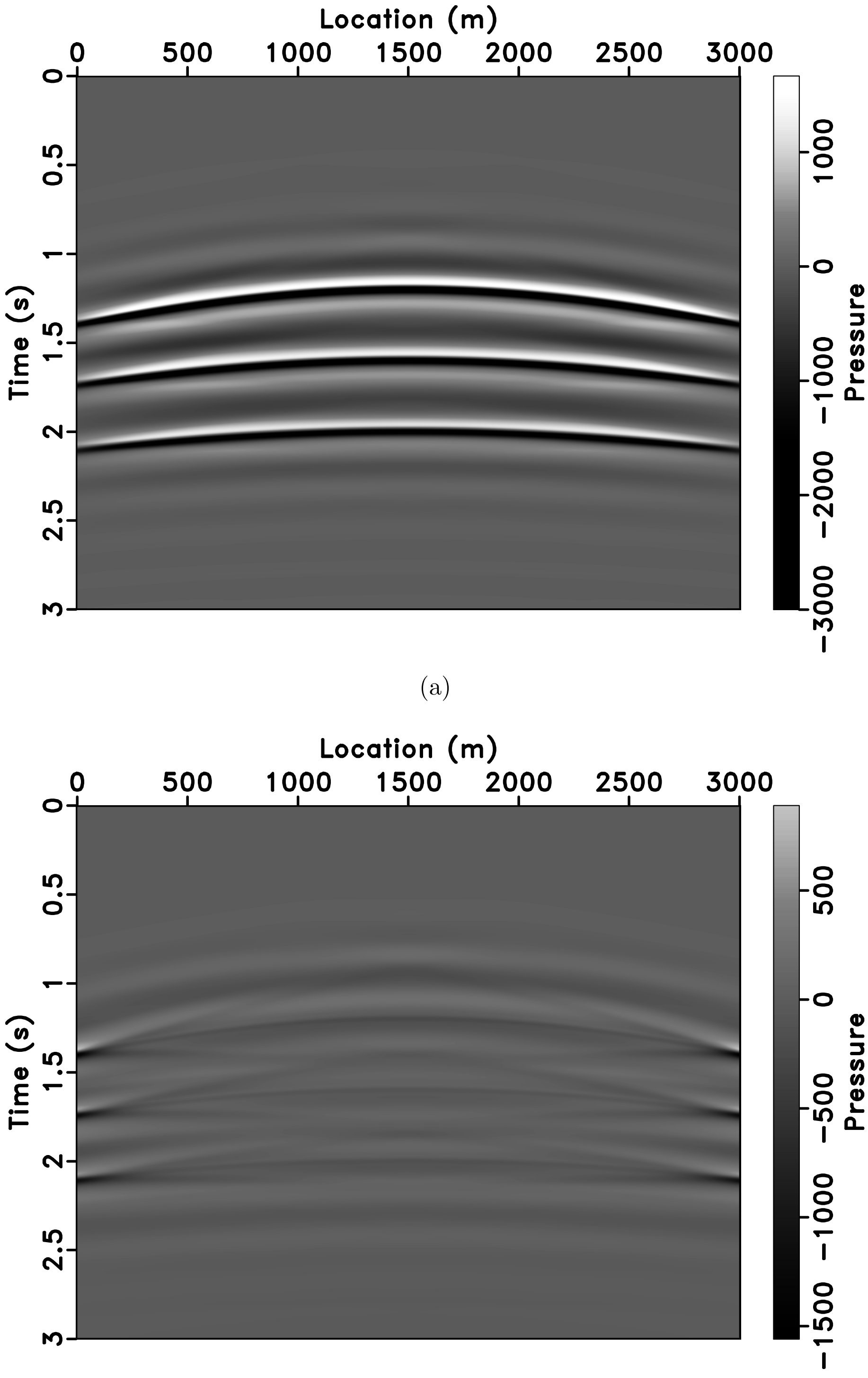


(a)



(b)

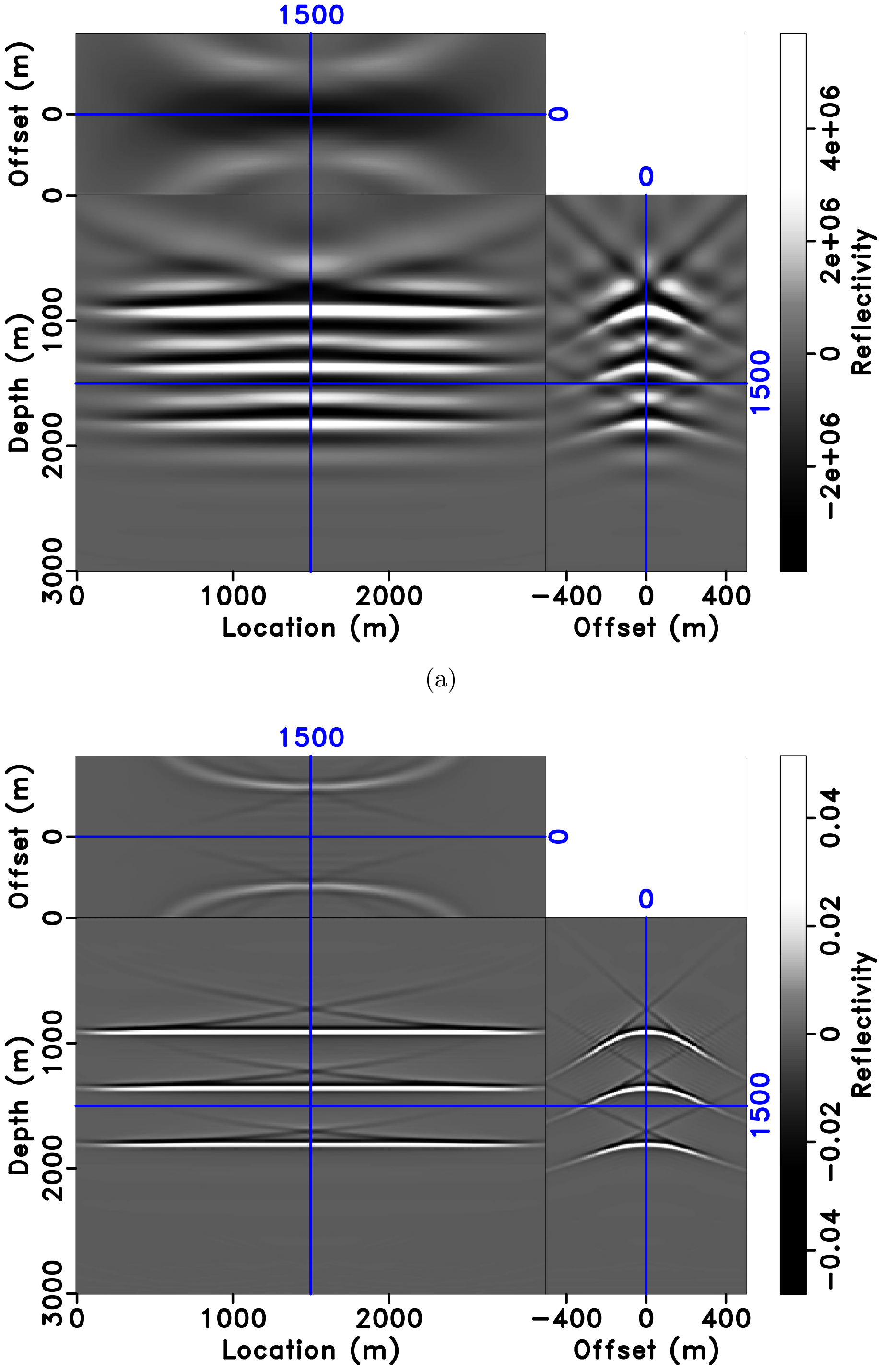
Figure 2: (a) Extended RTM image (b) Extended Inverted Reflectivity Model appinv/Fig/layer layerimagrtm,layerimag



(a)

(b)

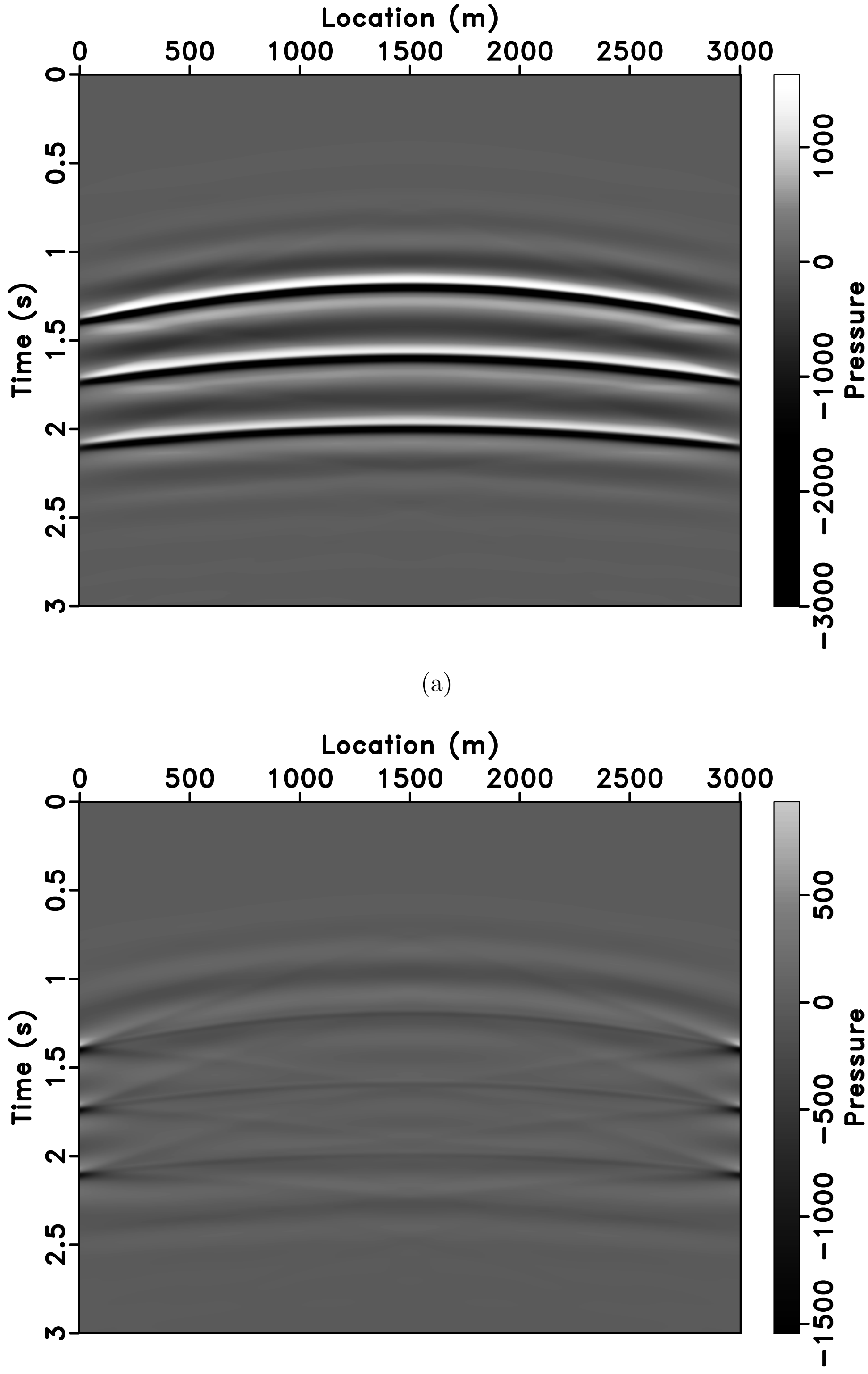
Figure 3: (a) Resimulated Data of the Inverted Reflectivity Model (b) Difference between the resimulated data and original data appinv/Fig/layer layerinvdata,datadiff



(a)

(b)

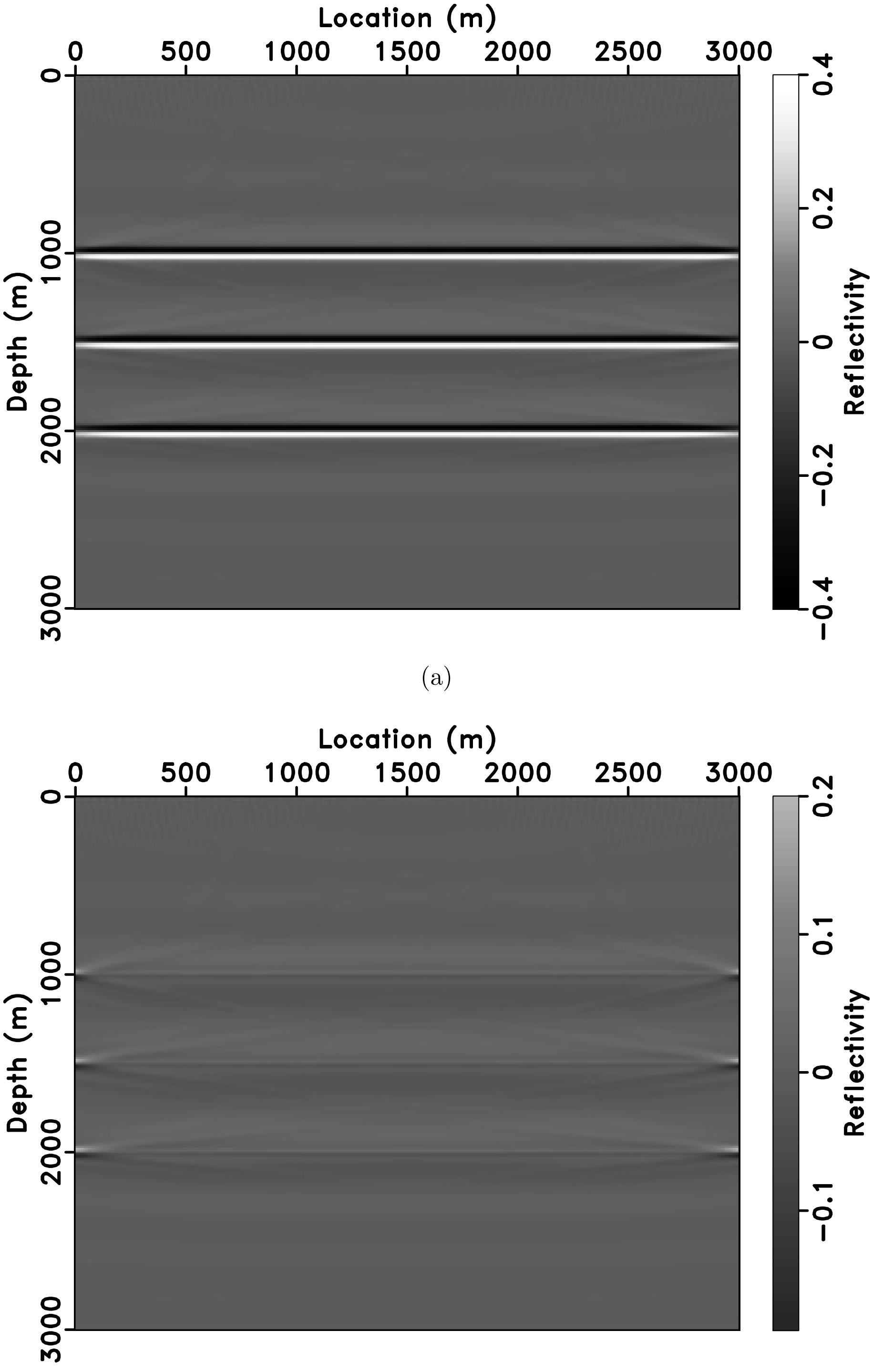
Figure 4: (a) Extended RTM image (b) Extended Inverted Reflectivity Model, both using an incorrect background velocity model appinv/Fig/layer layerimagrtmwrong,layerimagwrong



(a)

(b)

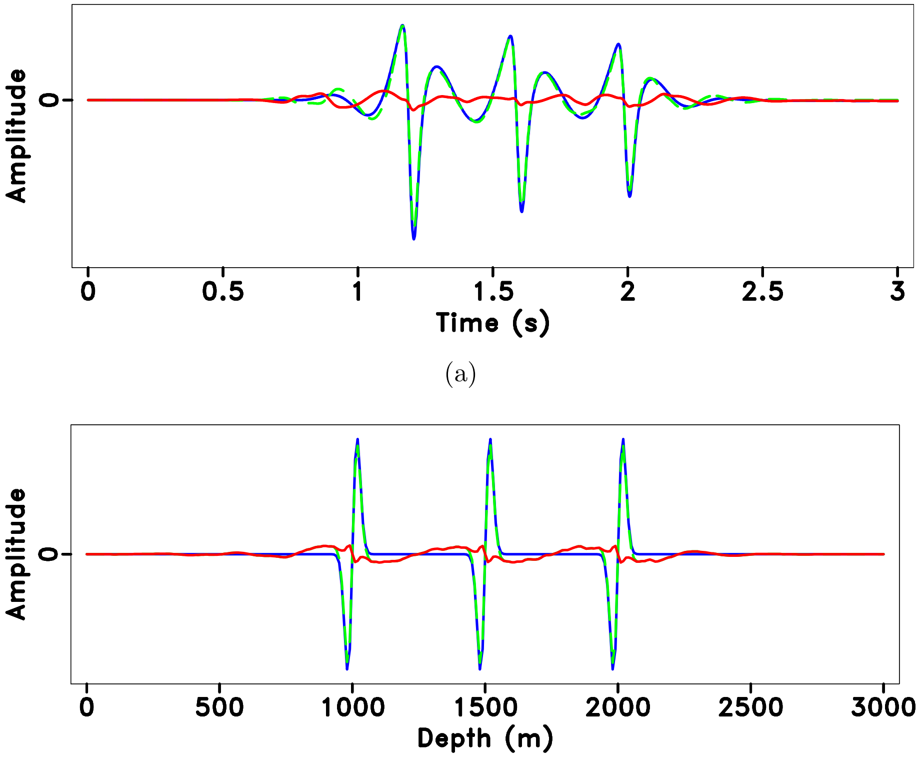
Figure 5: (a) Resimulated Data of the Inverted Reflectivity Model using an incorrect background velocity (b) Difference between the resimulated data and original data appinv/Fig/layer layerdatawrong,datadiffwrong



(a)

(b)

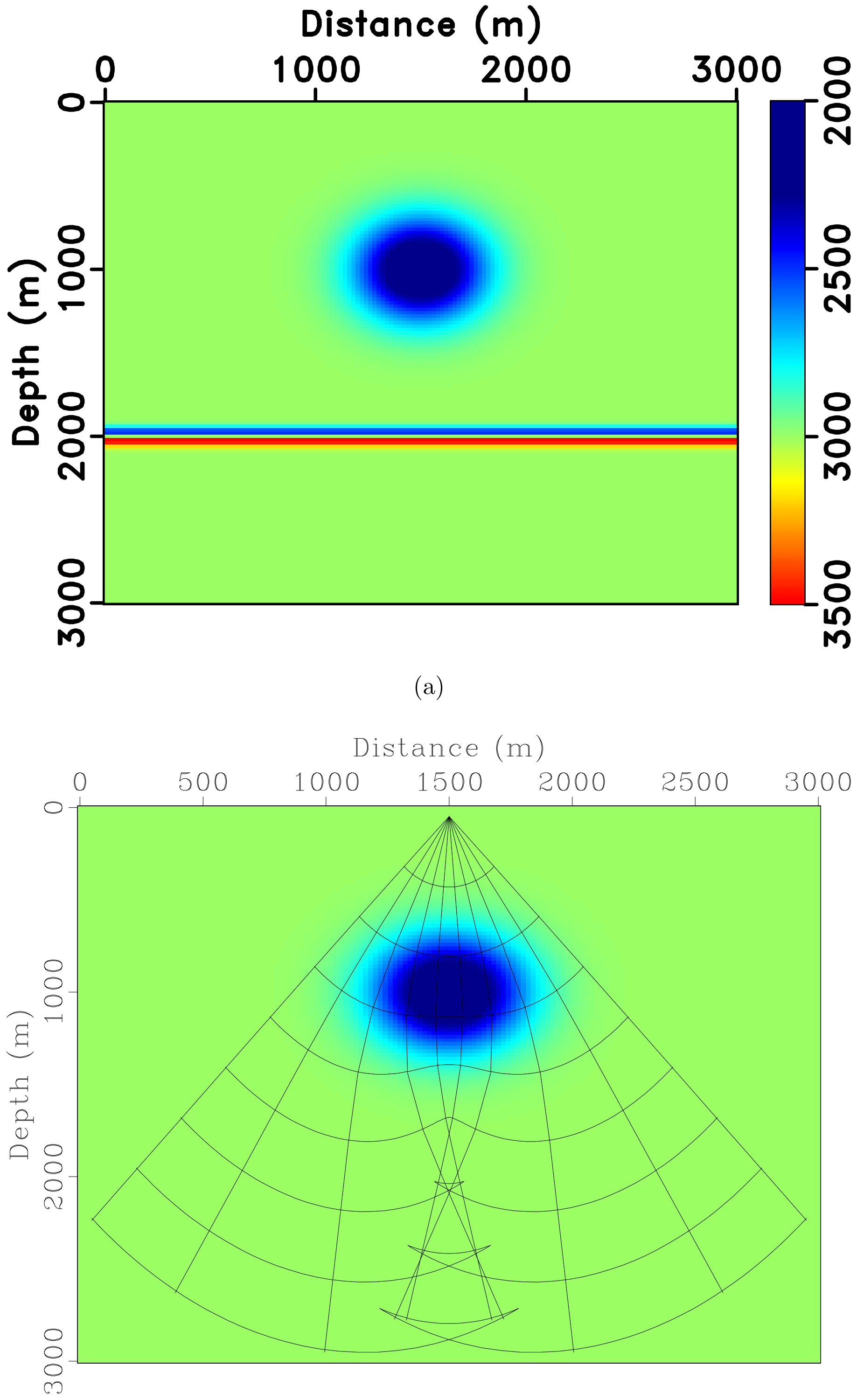
Figure 6: (a) Non-extended Inverted Reflectivity Model ($\sum_h i(x,h)$, where ($i(x,h)$ is the extended Inversion result) (b) Difference between the non-extended inversion result and original reflectivity model $\boxed{\text{appinv/Fig/layer layerstack,refdiff}}$



(a)

(b)

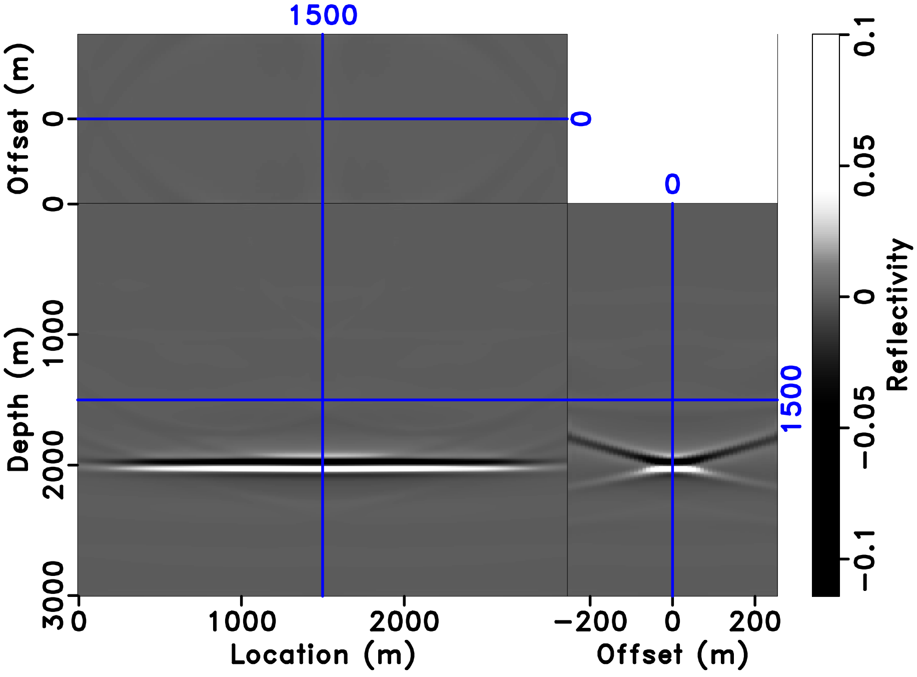
Figure 7: One trace comparison (x = 1500 m) between the observed data (blue solid line) and predicted data from inverted reflectivity model (green dashed line). The difference is shown as the red solid line. (b) One trace comparison (x = 1500 m) between the reflectivity model (blue solid line) and non-extended inverted reflectivity model (green dashed line). The difference is shown as the red solid line. appinv/Fig/layer onetrace,onetraceref



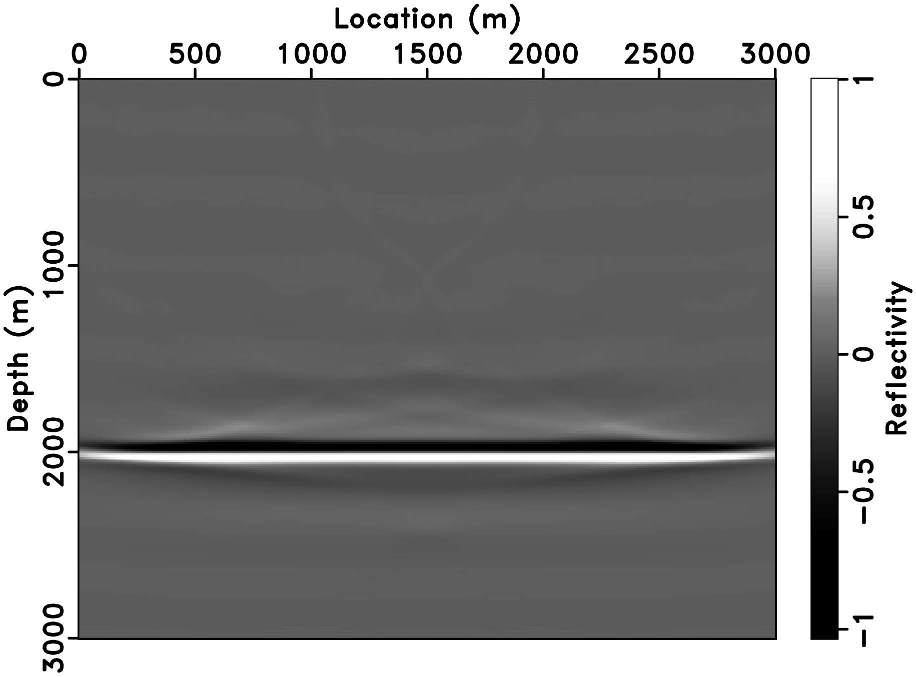
(a)

(b)

Figure 8: (a) Gaussian lens background velocity model with a reflector at 2 km (b) The rays and wavefronts in the Gaussian lens velocity model appinv/Fig/gauss vmod,gasswray

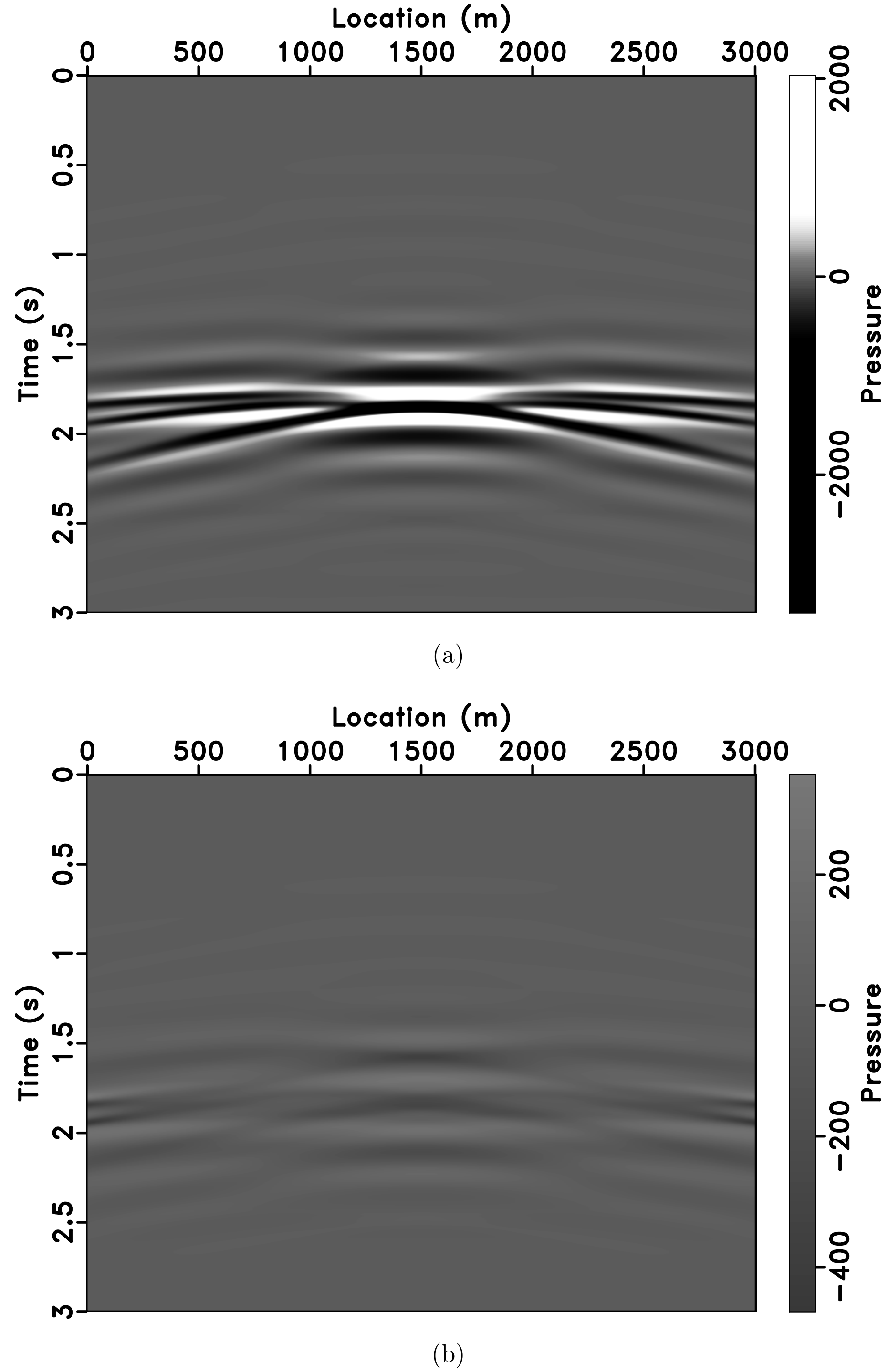


(a)



(b)

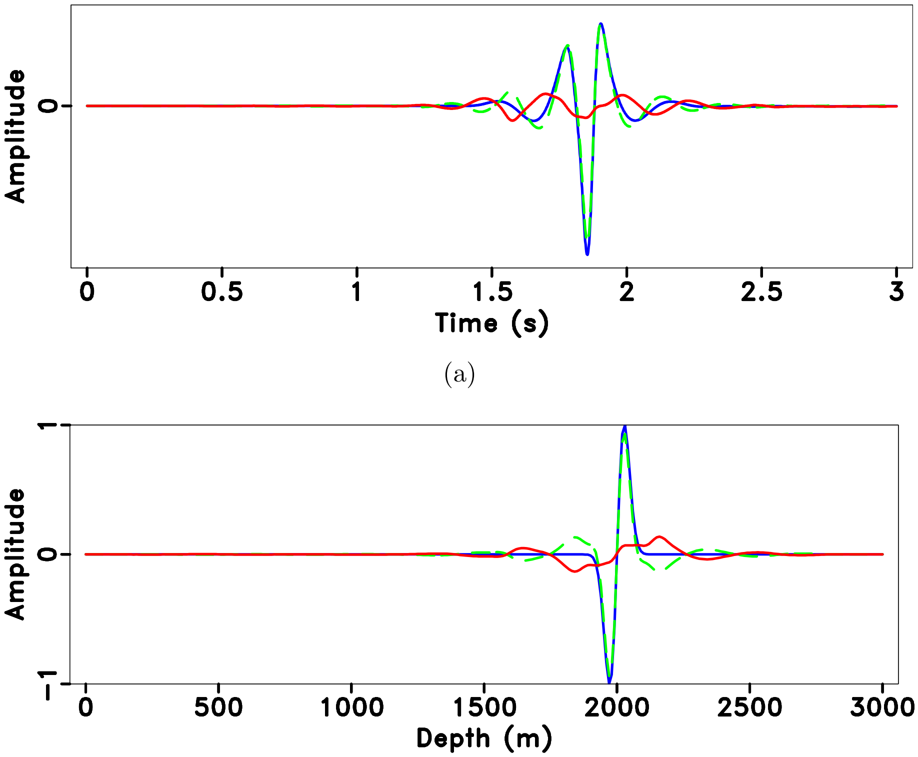
Figure 9: (a) Extended Inverted Reflectivity Model (b) Non-extended Inverted Reflectivity Model appinv/Fig/gauss image,stack



(a)

(b)

Figure 10: (a) One-shot ($x_s = 1500$ m) Resimulated Born Data (b) Difference between the resimulated data and original data appinv/Fig/gauss invdata,gassdatadiff



(a)

(b)

Figure 11: One trace comparison (x = 1500 m) between the observed data (blue solid line) and predicted data from inverted reflectivity model (green dashed line). The difference is shown as the red solid line. (b) One trace comparison (x = 1500 m) between the reflectivity model (blue solid line) and non-extended inverted reflectivity model (green dashed line). The difference is shown as the red solid line. appinv/Fig/gauss gassonedata,gassoneref

(fixed spread). The inverse operator has been applied on the Born data (middle shot shown in Figure 15(a)). The non-extended (stacked) inversion (Figure 14(b)) result is very similar visually to the reflectivity used in data synthesis (Figure 14(a)) - note that the grey scales used in these plots are identical.

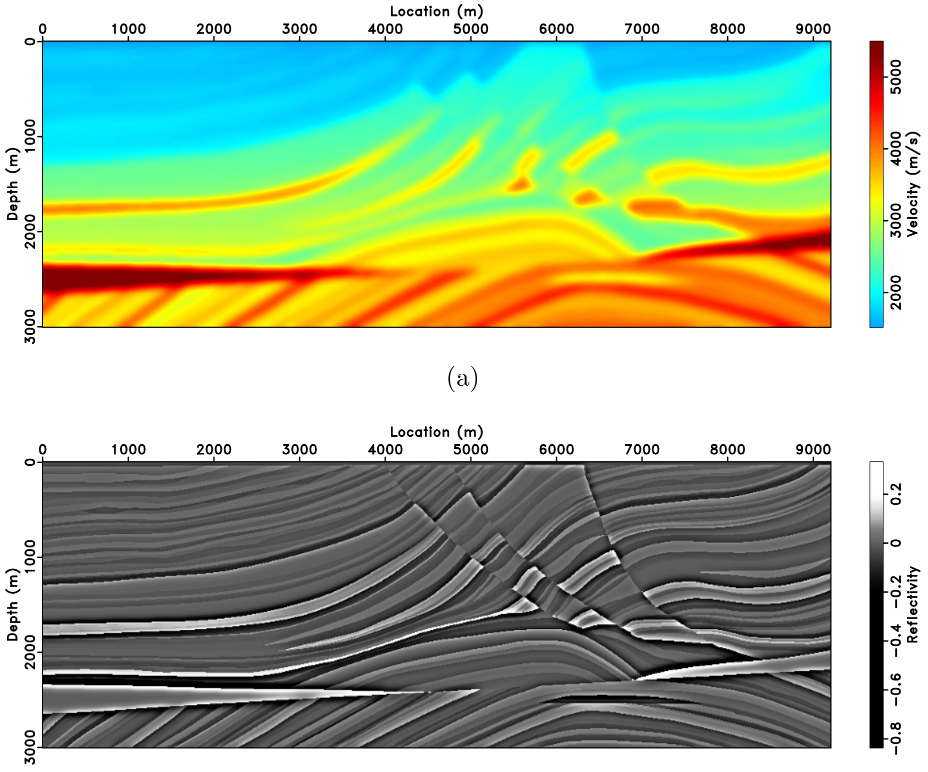
The approximate inversion (Figure 14(b)) is only an approximateion, of course. On the one hand, the inverse operator is only asymptotically correct. We can see this point from the fact that the difference between input and resimulated data is mainly low frequency. On the other hand, the theory leading to the conclusion in equation (29) explicitly ignores the possibility of scattering over π, that is, refraction. Some of the remaining energy in the residual data panel (Figure 15(c)) is refracted.

We further compare the approximate inverse operator with ELSM (Extended Least Squares Migration), that is, extended Born inversion via an iterative method (we use conjugate gradient iteration), in the same spirit as LSM (Nemeth et al., 1999; Dutta et al., 2014). A coarser grid (20 m for the spatial grid interval and 2 ms for the time interval) has been used to reduce the computational cost of ELSM. The approximate inverse operator yields the reflectivity model in Figure 17(a). Starting from the approximate inversion as an initial guess, we conduct 20 iterations ELSM (Figure 17(b)). On a visual comparison basis, the approximate inversion result displays no dramatic difference from ELSM result, except suffering from some low frequency noise in the shallow part. However, a quantitative study shown in the misfit comparison (Figure 18) reveals the significant difference in terms of data misfit. The approximate inversion result produces around 40% relative misfit (black dashed line), while the 20 iteration ELSM starting from the approximate inversion result achieves roughly 10% fit error (black solid line). Note that the approximate inverse operator has almost the same computational cost as a single application of extended RTM, whereas 7 or 8 iterations of ELSM (each involving a migration/demigration pair) starting from zero reflectivity were required to achieve the same 40% error reduction.

As mentioned in the introduction, the weighted adjoint form of the approximate inverse operator invites inclusion in a weighted conjugate gradient algorithm. Here, we merely present the convergence history of this algorithm, applied to the same problem. This algorithm, starting from zero reflectivity, achieves an error reduction of 10% in 6 or 7 iterations, as opposed to the 20 equally expensive iterations of ELSM starting from the approximate inversion result, or many more iterations of ELSM starting from zero reflectivity. In 20 iterations, the weighted conjugate gradient iteration reaches an RMS error reduction factor of 3.5%. The authors will discuss this algorithm in more detail elsewhere.

DISCUSSION

The obvious application of this construction is to accelerate iterative Least Squares Migration (LSM), both extended and non-extended variants. As mentioned in the



(a)

(b)

Figure 12: (a) Smoothed Background Velocity Model (b) Reflectivity Model appinv/Fig/marm marmbvel,marmref

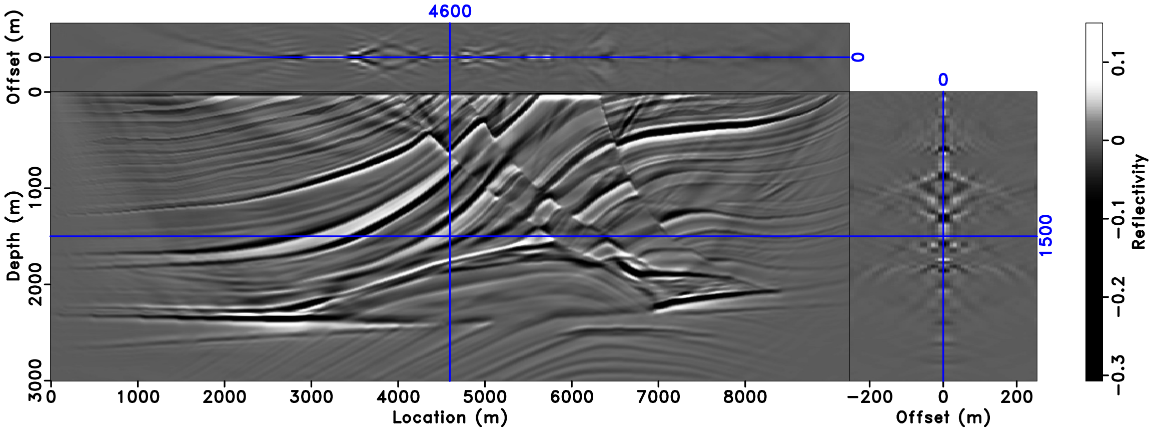
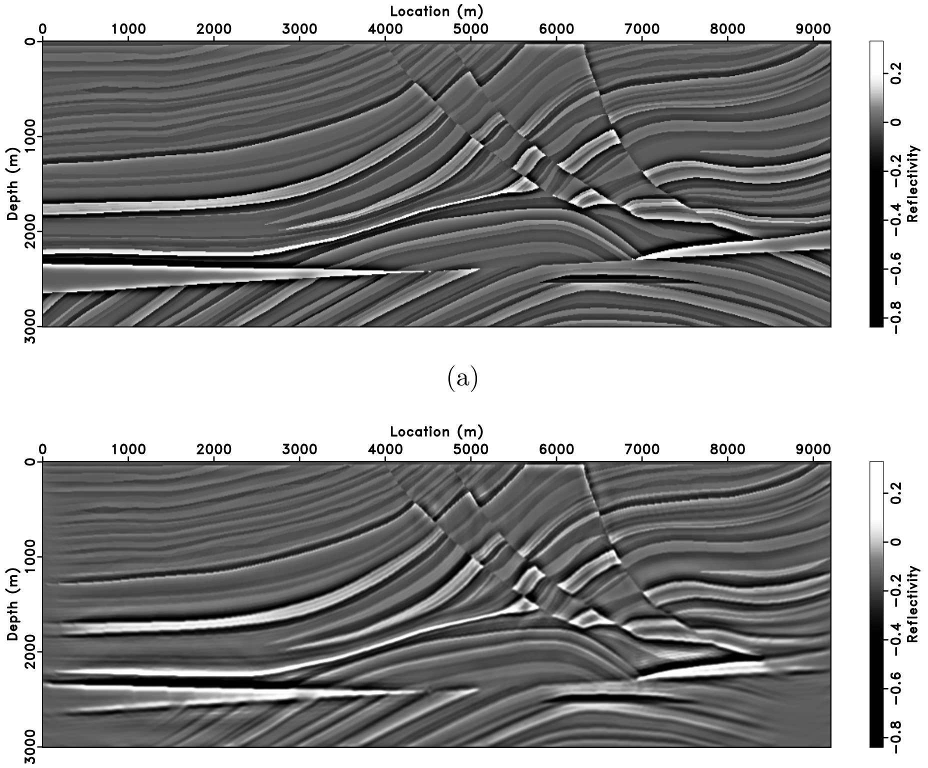


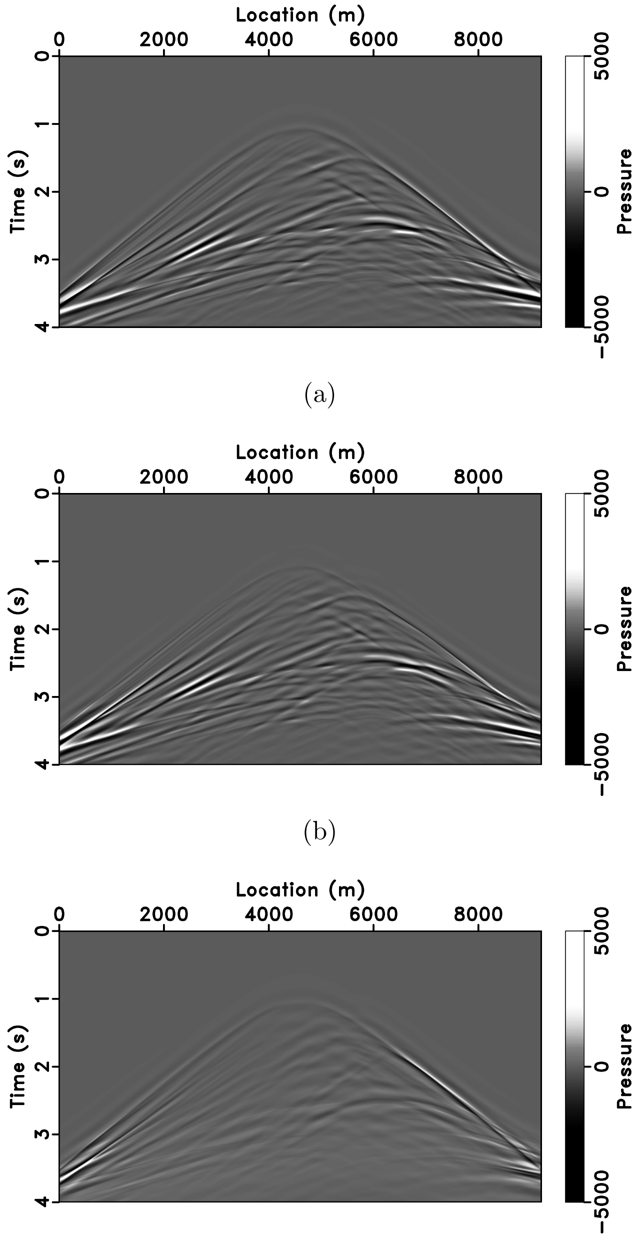
Figure 13: Extended Inverted Reflectivity Model appinv/Fig/marm marmimag



(a)

(b)

Figure 14: (a) Reflectivity Model (b) Non-extended Inverted Reflectivity Model appinv/Fig/marm marmref,marmstack

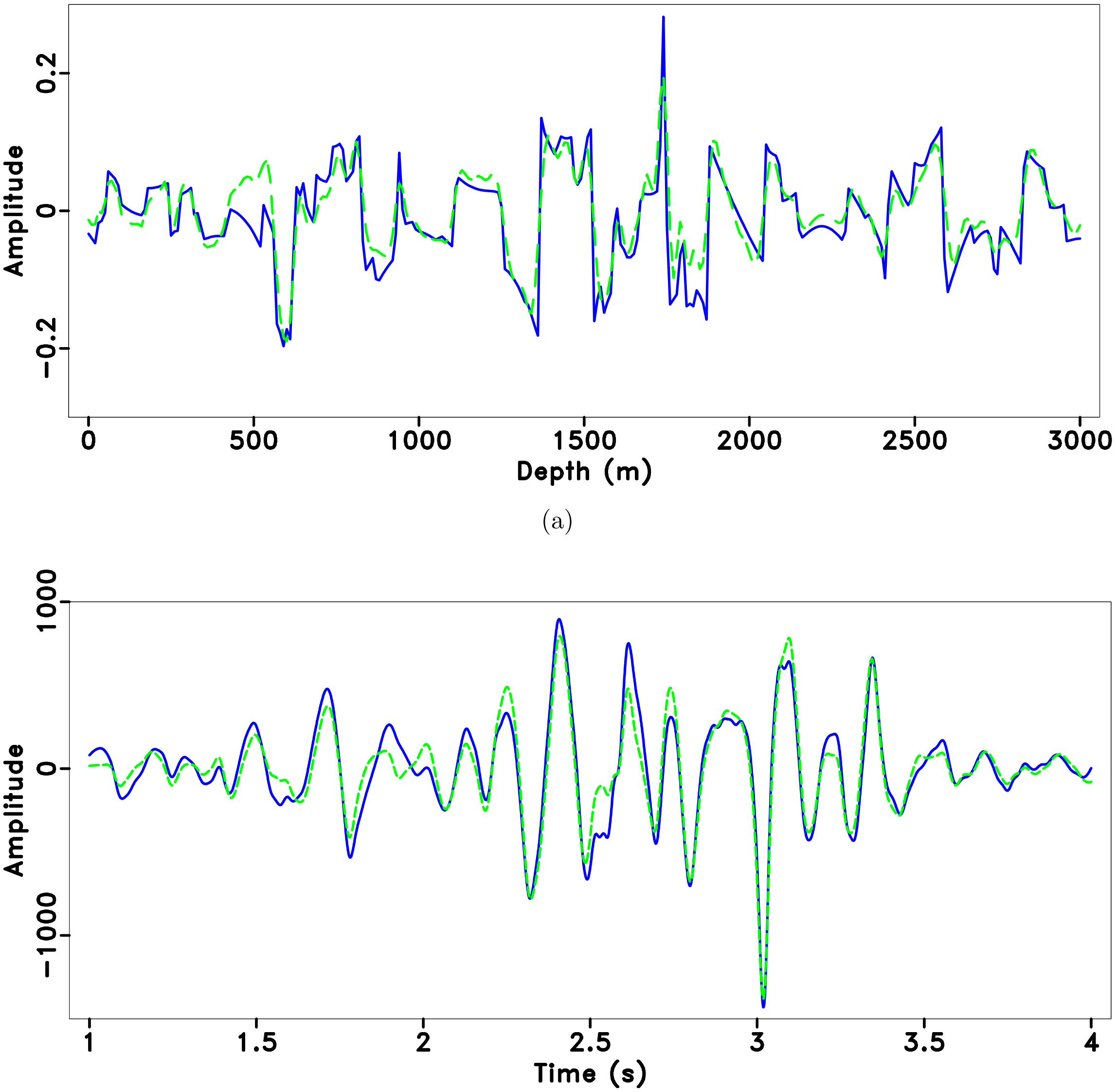


(a)

(b)

(c)

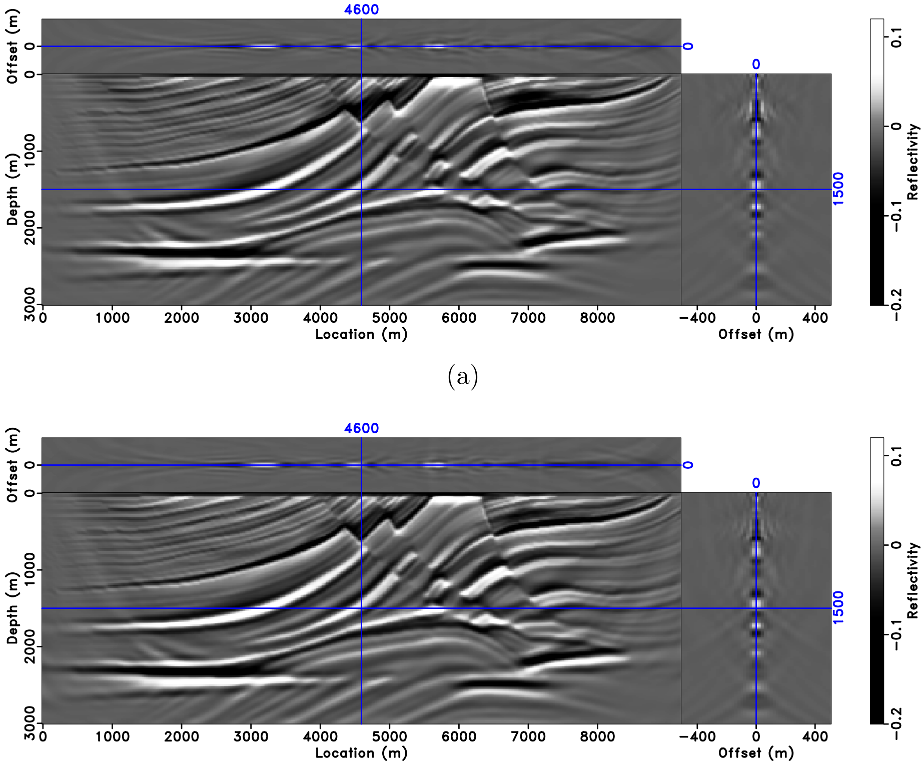
Figure 15: (a) One-shot (x_s = 4600 m) Born Data (b) One-shot (x_s = 4600 m) Resimulated Born Data (c) Difference between the resimulated data and original data appinv/Fig/marm marmdataone,marmdataf,marmdatadiff



(a)

(b)

Figure 16: (a) One trace comparison (x = 4600 m) between the reflectivity model (blue solid line) and non-extended inverted reflectivity model (green dashed line). (b) One trace comparison (x = 4600 m) between the observed data (blue solid line) and predicted data from the inverted reflectivity model (green dashed line). appinv/Fig/marm marmonecom,marmonedatacom



(a)

(b)

Figure 17: (a) Approximately inverted reflectivity model (b) 20 iterations ELSM image starting from the approximate inversion, plotted on same grey scale appinv/Fig/marm imag,imaglsm

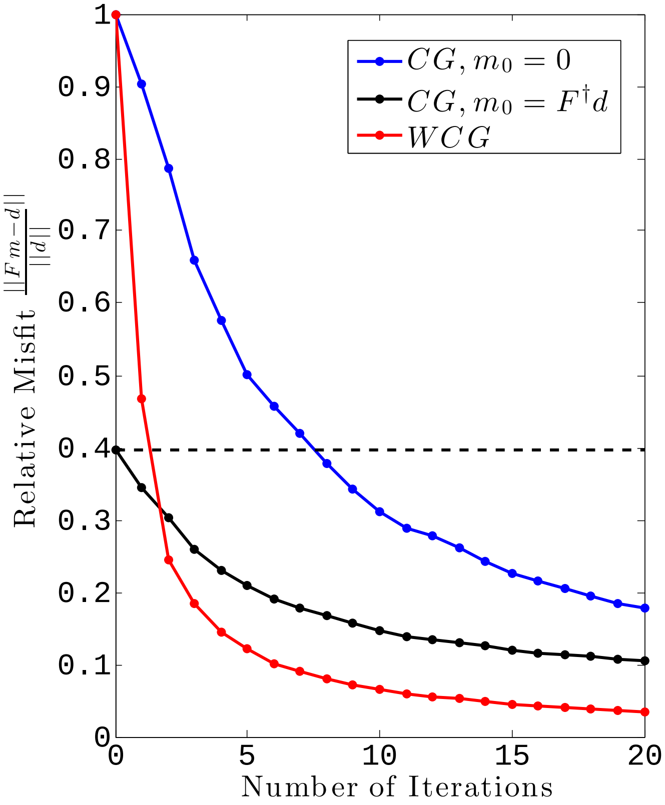


Figure 18: Relative Misfit as a function of the number of iterations: the black dashed line is the approximate inversion result; the blue solid line is the ELSM result using Conjugate Gradient algorithm and starting from zero reflectivity; the black solid line is the ELSM result using Conjugate Gradient and starting from the approximate inversion; the red solid line is the ELSM result using Weighted Conjugate Gradient (WCG) iteration. appinv/Fig/marm misfit

introduction, the form of the approximate inverse provides a straightforward acceleration mechanism, as it shows that the extended Born modeling operator is approximately unitary in specific, computable weighted norms. The main question in this regard is the necessary extent of the subsurface offset range. We have emphasized that this operator is negligibly more computationally intensive than *extended* RTM, however extended RTM is considerably more expensive than ordinary RTM: the additional expense comes in the sums over offset implicit in the definition of extended modeling or migration, and is roughly proportional to the extent of the offset axis. If an accurate velocity is available, which focuses the data, this problem is mitigated, of course. In this case, weight function ϕ appearing in (29) is in principle constrained only by the requirement that $\phi(0) = 1$, except for finite-frequency effects. Since a weighted stack over subsurface offset is equivalent to a weighted stack over scattering angle, the choice of ϕ will have important ramifications even for the focused case: stacking with $\phi = 1$ is equivalent (roughly) to using only the zero scattering angle, or zero offset data, and would likely lead to suboptimal noise suppression. The design of an optimal weight ϕ in the focused case, and the necessary extent of the offset range for the unfocused case and effective algorithms for determining it, are important open questions.

The operator defined in equations (1), (24), (28) approximately inverts the extended Born modeling operator, therefore may be used to estimate the *extended* reflectivity in automated velocity model building methods that depend on extended inversion (Liu et al., 2013, 2014; Lameloise et al., 2014), or for AVO studies. Note however that in our examples we have adopted the approximation $P = 1$, which strictly speaking is accurate only if the image energy focuses at $h = 0$ (that is, the velocity and data are compatible) or v_0 is laterally homogeneous. In general, P is not $\equiv 1$ in the full extended phase space volume. It is actually quite possible to remove this final impediment to asymptotic inversion: P is defined explicitly in Appendix A as a function of phase variables, and pseudodifferential operators such as W_{model} are effectively of low rank computable with relative efficiency, in comparison to a general matrix multiplication of the same dimension (Bao and Symes, 1996; Demanet and Ying, 2011). The improvement obtainable by using one of these techniques to drop the approximation $P = 1$ remains to be investigated. Any such computation involves the Fourier transform of the extended model, or a transform of equivalent complexity. In 2D, this is already a 3D transform, while for 3 spatial dimensions, the transform is of dimension 5, which is a daunting prospect. However transformation to the angle domain after Sava and Fomel (2003), considered by many authors to be an essential step (ten Kroode, 2012; Tang et al., 2013), is usually accomplished via the Fourier transform hence involves the same computational load.

CONCLUSION

A simple modification of subsurface offset extended RTM produces an asymptotic inverse to the extended Born scattering operator. Implementation of straightforward, and numerical experiments suggest that within its domain of applicability, this inversion operator is quite accurate. The asymptotic inverse takes the form of the adjoint of the modeling operator with respect to weighted norms in model and data spaces, hence offers the possibility of greatly enhancing the convergence of iterative methods for extended Born inversion.

ACKNOWLEDGEMENTS

We are grateful to the sponsors of The Rice Inversion Project (TRIP) for their long-term support, and to Shell International Exploration and Production Inc. for its support of Jie Hou's PhD research. We thank Fons ten Kroode for inspiring our work. Thanks also go to Jon Sheiman, Henning Kuehl, Peng Shen and Yujin Liu for very helpful discussion. We have benefited greatly from the high performance computating resources provided by the Texas Advanced Computing Center, University of Texas - Austin, and by the Rice University Research Computing Support Group. The Seismic Unix and Madagascar open source software packages have been critically useful in our work. We appreciate the constructive comments from Assistant Editor M. Sacchi, Associate Editor I. Jones, referee J. Sheiman, and two anonymonous referees.

APPENDIX A

PSEUDODIFFERENTIAL EXPRESSION OF THE NORMAL OPERATOR

The goal of this appendix is an expression for the slightly modified normal operator $(I_t\bar{F}[v_0])^*$ $(I_t\bar{F}[v_0])$ in the form of an oscillatory integral

$$(I_t\bar{F}[v])^*(I_t\bar{F}[v])\delta\bar{v}(\mathbf{x},h) \approx \int d\mathbf{k}A(\mathbf{x},\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{x}}\hat{\delta v}(\mathbf{k})$$

over the frequency variables $\mathbf{k} = (k_x, k_h, k_z)$, modulo errors decaying faster than the amplitude A at large frequency. In fact we will express the amplitude as a product of two factors, one depending on ray-trace quantities, the other depending only on values of velocity and the phase variables. The ray-dependent quantities are eventually eliminated via the identities proved in Appendix B and some further modifications of the normal operator, leading to the main result of this paper.

Begin by combining the asymptotic expressions (10) and (11), eliminating the time integral in the delta functions, and taking into account the fact that $I_t^* = -I_t$ to obtain an integral expression for the modified normal operator introduced at the beginning of the Theory section:

$$(I_t\bar{F})^*(I_t\bar{F})\delta\bar{v}(\mathbf{x},h)$$

$$\simeq -\frac{4\pi^2}{v_0(x,z)^6}\int dx_s dx_r a_s a_r \int d\mathbf{x}' dh' a'_s a'_r \delta(\phi(x_s,x_r,x,h,z)-\phi(x_s,x_r,x',h',z'))\delta\bar{v}(\mathbf{x}',h') \qquad \text{(A-1)}$$

where $\phi(x_s,x_r,x,h,z) = T(x_s,x-h,z) + T(x_r,x+h,z)$ is the two way traveltime.

We recall the abbreviations

$$T_s = T(x_s,x-h,z),\ T_r = T(x_r,x+h,z),$$

and note that

$$\frac{\partial T_s}{\partial h} = -\frac{\partial T_s}{\partial x}, \frac{\partial T_r}{\partial h} = \frac{\partial T_r}{\partial x},$$

An asymptotic evaluation of this integral follows along the lines pioneered Beylkin (1985), as recast by Symes (1998). Account for the delta function $\delta(\phi(x_s,x_r,x,h,z)-\phi(x_s,x_r,x',h',z'))$ by writing z' as a function of x,h,z,x',h',x_s,x_r - as it will turn out, this possibility assumes that energy propagates vertically and reflectors are sub-horizontal, otherwise other space variables should be treated as dependent - then rewrite using the inverse Fourier transform:

$$\begin{aligned}(I_t\bar{F})^*(I_t\bar{F})\delta\bar{v}(\mathbf{x},h) \simeq & -\frac{4\pi^2}{v_0(x,z)^6}\int dx_s dx_r a_s a_r \int d\mathbf{x}' dh' a_s' a_r' \delta(z' - Z(x_s,x_r,x,h,z,x',h')) \\ & \times \left|\frac{\partial\phi}{\partial z'}\right|^{-1} \frac{1}{8\pi^3}\int dk_{x'}dk_{z'}dk_{h'}\hat{\delta v}(k_{x'},k_{h'},k_{z'})e^{i(k_{x'}x'+k_{h'}h'+k_{z'}z')}\end{aligned} \tag{A-2}$$

The principle of stationary phase (Guillemin and Sternberg, 1979; Bleistein et al., 2001) is used to evaluate the multiple integral for large wavenumber. This result approximates the integral of a rapidly fluctuating function $g(\mathbf{y})e^{i\omega\psi(\mathbf{y})}$ for large ω by a sum of terms, one for each stationary phase point $\mathbf{y}^*$ (that is $\nabla\psi(\mathbf{y}^*) = 0$). The general form can be written as :

$$\begin{aligned}&\int_{\mathbf{R}^m} d\mathbf{y}g(\mathbf{y})e^{i\omega\psi(\mathbf{y})} \\ &\approx \sum_{\nabla\psi(\mathbf{y}^*)=0}\left(\frac{2\pi}{\omega}\right)^{\frac{m}{2}} e^{\frac{\pi i}{4}\mathrm{sgn\ Hess}\ \psi(\mathbf{y}^*)}|\det \mathrm{Hess}\psi(\mathbf{y}^*)|^{-\frac{1}{2}}g(\mathbf{y}^*)e^{i\omega\psi(\mathbf{y}^*)}\end{aligned} \tag{A-3}$$

Employing this approximation, the right-hand side of equation (A-2) is

$$\int dk_{x'}dk_{z'}dk_{h'}\hat{\delta v}(k_{x'},k_{h'},k_{z'})\int_{\mathbf{R}^m} d\mathbf{y}g(\mathbf{y})e^{ik_{z'}\psi(\mathbf{y})} \tag{A-4}$$

where in the general statement of the stationary phase principle (A-3) we have set

$$\begin{aligned}& m = 4, \mathbf{y} = (x_s, x_r, x', h') \\ & g(\mathbf{y}) = -\frac{1}{2\pi v_0(x,z)^6}a_s a_r a_s' a_r'\left|\frac{\partial\phi}{\partial z'}\right|^{-1} \\ & \psi(\mathbf{y}) = \frac{k_{x'}}{k_{z'}}x' + \frac{k_{h'}}{k_{z'}}h' + Z(x_s,x_r,x,h,z,x',h')\end{aligned} \tag{A-5}$$

and $k_{z'}$ plays the role of large parameter ω. To employ this approximation, the Hessian

$$\mathrm{Hess}\ \psi(\mathbf{y}^*) = \left(\frac{\partial^2\psi}{\partial y_i \partial y_j}\right)_{i,j=1}^m$$

must be nonsingular at each stationary phase point, and we must compute the signature and determinant of the Hessian at each such point. After standard simplifications, the stationary phase conditions are

$$\begin{aligned}&\mathbf{x} = \mathbf{x}',\ h = h' \\ &(k_{x'}, k_{z'}) \text{ is parallel to } \nabla_{(x',z')}\phi \\ &(k_{h'}, k_{z'}) \text{ is parallel to } \nabla_{(h',z')}\phi\end{aligned} \tag{A-6}$$

During the calculation of the Hessian, the integrations are naturally paired as (x, x_r) and (h, x_s). Each pair of integrals gives rise to a Hessian determinant factor.

The Hessian of the phase ψ can be expressed as :

$$\text{Hess} = \begin{pmatrix} \dfrac{\partial^2 Z}{\partial x_r^2} & \dfrac{\partial^2 Z}{\partial x_r \partial x_s} & \dfrac{\partial^2 Z}{\partial x_r \partial x'} & \dfrac{\partial^2 Z}{\partial x_r \partial h'} \\ \dfrac{\partial^2 Z}{\partial x_s \partial x_r} & \dfrac{\partial^2 Z}{\partial x_s^2} & \dfrac{\partial^2 Z}{\partial x_s \partial x'} & \dfrac{\partial^2 Z}{\partial x_s \partial h'} \\ \dfrac{\partial^2 Z}{\partial x' \partial x_r} & \dfrac{\partial^2 Z}{\partial x' \partial x_s} & \dfrac{\partial^2 Z}{\partial x'^2} & \dfrac{\partial^2 Z}{\partial x' \partial h'} \\ \dfrac{\partial^2 Z}{\partial h' \partial x_r} & \dfrac{\partial^2 Z}{\partial h' \partial x_s} & \dfrac{\partial^2 Z}{\partial h' \partial x'} & \dfrac{\partial^2 Z}{\partial h'^2} \end{pmatrix} \tag{A-7}$$

In computing the Hessian, all derivatives must be performed before the stationary phase identities (A-6) are applied. The solution $z' = Z(x_s, x_r, x, h, z, x', h')$ must satisfy the Two-way Traveltime Equation

$$\phi(x_s, x_r, x', h', z') = \phi(x_s, x_r, x, h, z) \tag{A-8}$$

So we differentiate this equation, regarding x' as independent of x etc., and afterwards combine with the stationary phase condition (A-6) to obtain:

$$\frac{\partial^2 Z}{\partial x_r^2} = 0, \quad \frac{\partial^2 Z}{\partial x_s^2} = 0, \quad \frac{\partial^2 Z}{\partial x_r \partial x_s} = 0 \tag{A-9}$$

Also

$$\begin{gathered} \frac{\partial^2 \phi}{\partial x_r \partial x'} + \frac{\partial^2 \phi}{\partial x_r \partial z'} \frac{\partial Z}{\partial x'} + \frac{\partial \phi}{\partial z'} \frac{\partial^2 Z}{\partial x_r \partial x'} = 0 \\ \frac{\partial^2 \phi}{\partial x_r \partial h'} + \frac{\partial^2 \phi}{\partial x_r \partial z'} \frac{\partial Z}{\partial h'} + \frac{\partial \phi}{\partial z'} \frac{\partial^2 Z}{\partial x_r \partial h'} = 0 \\ \frac{\partial^2 \phi}{\partial x_s \partial x'} + \frac{\partial^2 \phi}{\partial x_s \partial z'} \frac{\partial Z}{\partial x'} + \frac{\partial \phi}{\partial z'} \frac{\partial^2 Z}{\partial x_s \partial x'} = 0 \\ \frac{\partial^2 \phi}{\partial x_s \partial h'} + \frac{\partial^2 \phi}{\partial x_s \partial z'} \frac{\partial Z}{\partial h'} + \frac{\partial \phi}{\partial z'} \frac{\partial^2 Z}{\partial x_s \partial h'} = 0 \end{gathered} \tag{A-10}$$

Note that the upper left hand 2 × 2 block of the Hessian consists of zeroes, and of course the Hessian is symmetric. That is, the Hessian has the block structure:

$$\text{Hess} \sim \begin{pmatrix} 0 & A \\ A^T & B \end{pmatrix} \tag{A-11}$$

This special structure implies that the Hessian at the stationary point has exactly the same number of positive as negative eigenvalues, that is, the signature of the Hessian is zero. For details of this argument see Symes (1998).

The block structure (A-11) also allows us to reduce the determinant to that of a 2×2 matrix, squared:

$$\det \text{Hess} = - \begin{vmatrix} \dfrac{\partial^2 Z}{\partial x' \partial x_r} & \dfrac{\partial^2 Z}{\partial x' \partial x_s} \\ \dfrac{\partial^2 Z}{\partial h' \partial x_r} & \dfrac{\partial^2 Z}{\partial h' \partial x_s} \end{vmatrix}^2 \tag{A-12}$$

The four elements of this matrix have similar structure and can be analyzed in the same way. Take the first element as an example. Substitute equation A-10 into the first element, we get

$$\frac{\partial^2 Z}{\partial x' \partial x_r} = -\left(\frac{\partial \phi}{\partial z'}\right)^{-2} \left(\frac{\partial^2 \phi}{\partial x_r \partial x'} \frac{\partial \phi}{\partial z'} - \frac{\partial^2 \phi}{\partial x_r \partial z'} \frac{\partial \phi}{\partial x'}\right)$$

$$= -\left(\frac{\partial \phi}{\partial z'}\right)^{-2} \det \begin{pmatrix} \frac{\partial}{\partial x_r} \nabla_{\mathbf{x}'} \phi \\ \nabla_{\mathbf{x}'} \phi \end{pmatrix} \tag{A-13}$$

Applying same analysis on other elements, and using the stationary phase conditions $x' = x, h' = h, z' = z$ (since all derivatives have been computed) leads to

$$|\det \text{Hess}|^{-1/2} = \left[\left(\frac{\partial \phi}{\partial z}\right)^{-4} \begin{pmatrix} \det \begin{pmatrix} \frac{\partial}{\partial x_r} \nabla_{(x,z)} \phi \\ \nabla_{(x,z)} \phi \end{pmatrix} & \det \begin{pmatrix} \frac{\partial}{\partial x_s} \nabla_{(x,z)} \phi \\ \nabla_{(x,z)} \phi \end{pmatrix} \\ \det \begin{pmatrix} \frac{\partial}{\partial x_r} \nabla_{(h,z)} \phi \\ \nabla_{(h,z)} \phi \end{pmatrix} & \det \begin{pmatrix} \frac{\partial}{\partial x_s} \nabla_{(h,z)} \phi \\ \nabla_{(h,z)} \phi \end{pmatrix} \end{pmatrix} \right]^{-1} \tag{A-14}$$

Write $s = 1/v_0$ for slowness, $s_\pm = s(x \pm h, z)$. Note that the eikonal equation asserts that

$$\nabla_{x,z} T_s \cdot \nabla_{x,z} T_s = s_-^2, \;\; \nabla_{x,z} T_r \cdot \nabla_{x,z} T_r = s_+^2. \tag{A-15}$$

Since the lengths of the traveltime gradients are independent of the source and receiver coordinates, their derivatives with respect to these coordinates must be orthogonal to the gradients. A simple way to express this fact is to write

$$\nabla_{x,z} T_s = s_-(\sin \alpha_s, \cos \alpha_s) \tag{A-16}$$

$$\nabla_{x,z} T_r = s_+(\sin \alpha_r, \cos \alpha_r) \tag{A-17}$$

so

$$\begin{array}{ll} \frac{\partial}{\partial x_s} \frac{\partial T_s}{\partial x} = \frac{\partial T_s}{\partial z} \frac{\partial \alpha_s}{\partial x_s}, & \frac{\partial}{\partial x_s} \frac{\partial T_s}{\partial z} = -\frac{\partial T_s}{\partial x} \frac{\partial \alpha_s}{\partial x_s} \\ \frac{\partial}{\partial x_r} \frac{\partial T_r}{\partial x} = \frac{\partial T_r}{\partial z} \frac{\partial \alpha_r}{\partial x_r}, & \frac{\partial}{\partial x_r} \frac{\partial T_r}{\partial z} = -\frac{\partial T_r}{\partial x} \frac{\partial \alpha_r}{\partial x_r} \end{array} \tag{A-18}$$

The elements of the matrix in (A-14) simplify due to (A-18): for example, the (1,1) element becomes

$$\begin{vmatrix} \frac{\partial}{\partial x_r} \frac{\partial T_r}{\partial x} & \frac{\partial}{\partial x_r} \frac{\partial T_r}{\partial z} \\ \frac{\partial T_r}{\partial x} + \frac{\partial T_s}{\partial x} & \frac{\partial T_r}{\partial z} + \frac{\partial T_s}{\partial z} \end{vmatrix} = \frac{\partial \alpha_r}{\partial x_r} \begin{vmatrix} \frac{\partial T_r}{\partial z} & -\frac{\partial T_r}{\partial x} \\ \frac{\partial T_r}{\partial x} + \frac{\partial T_s}{\partial x} & \frac{\partial T_r}{\partial z} + \frac{\partial T_s}{\partial z} \end{vmatrix} = s_-^2 + \nabla_{x,z} T_r \cdot \nabla_{x,z} T_s$$

by virtue of the eikonal equation (A-15). Evaluating the other elements similarly,

$$|\text{det Hess}|^{-1/2} = \left(\frac{\partial\phi}{\partial z}\right)^4 \left[\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r} \left| \begin{array}{cc} s_+^2 + \nabla_{x,z}T_r\cdot\nabla_{x,z}T_s & s_-^2 + \nabla_{x,z}T_r\cdot\nabla_{x,z}T_s \\ s_+^2 + \nabla_{x,z}T_r\cdot\nabla_{h,z}T_s & -s_-^2 - \nabla_{x,z}T_r\cdot\nabla_{h,z}T_s \end{array} \right|\right]^{-1}$$

$$= -\frac{1}{2}\left(\frac{\partial\phi}{\partial z}\right)^4 \left(\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r}\right)^{-1} \big[(s_-^2 s_+^2 + (\nabla_{x,z}T_r\cdot\nabla_{x,z}T_s)(\nabla_{x,z}T_r\cdot\nabla_{h,z}T_s)$$

$$+ (s_-^2 + s_+^2)(\nabla_{x,z}T_r\cdot\nabla_{x,z}T_s + \nabla_{x,z}T_r\cdot\nabla_{h,z}T_s))\big]^{-1}$$

$$= -\frac{1}{2}\left(\frac{\partial\phi}{\partial z}\right)^4 \left(\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r}\right)^{-1} \left[\left(s_-^2\left(\frac{\partial T_r}{\partial z}\right)^2 + s_+^2\left(\frac{\partial T_s}{\partial z}\right)^2\right) + (s_-^2 + s_+^2)\left(\frac{\partial T_s}{\partial z}\frac{\partial T_r}{\partial z}\right)\right]^{-1} \tag{A-19}$$

Apart from the angle derivatives, this expression is actually algebraic in the phase variables and $s_\pm$. To see this, invoke the remaining stationary phase conditions (A-6) pertaining to the phase variables. Adding and subtracting ratios equivalent to these conditions, obtain

$$\begin{aligned} \frac{\partial T_r}{\partial x} &= \frac{1}{2}\frac{k_x + k_h}{k_z}\frac{\partial\phi}{\partial z} \\ \frac{\partial T_s}{\partial x} &= \frac{1}{2}\frac{k_x - k_h}{k_z}\frac{\partial\phi}{\partial z} \end{aligned} \tag{A-20}$$

Then the eikonal equation implies that

$$\begin{aligned} s_+^2 - \frac{1}{4}\left[\frac{k_x + k_h}{k_z}\right]^2\left(\frac{\partial\phi}{\partial z}\right)^2 &= \left(\frac{\partial T_r}{\partial z}\right)^2 \\ s_-^2 - \frac{1}{4}\left[\frac{k_x - k_h}{k_z}\right]^2\left(\frac{\partial\phi}{\partial z}\right)^2 &= \left(\frac{\partial T_s}{\partial z}\right)^2 \end{aligned} \tag{A-21}$$

For convenience, set

$$a_\pm = \frac{1}{4}\left[\frac{k_x \pm k_h}{k_z}\right]^2, \ \zeta_\pm = \frac{\partial T_r}{\partial z} \pm \frac{\partial T_s}{\partial z}$$

so $\zeta_+ = \partial\phi/\partial z$.

Adding the two equations (A-21), and rearranging, obtain ζ_-^2 in terms of ζ_+^2:

$$\zeta_-^2 = 2(s_+^2 + s_-^2) - [2(a_+ + a_-) + 1]\zeta_+^2. \tag{A-22}$$

Subtracting the two equations (A-21), squaring the result, eliminating ζ_-^2 using (A-22), and rearranging yields a quadratic equation for ζ_+^2:

$$a(\zeta_+^2)^2 + b\zeta_+^2 + c = 0, \tag{A-23}$$

in which

$$a = (a_+ - a_-)^2 + 2(a_+ + a_-) + 1 = \frac{k_{xz}^2 k_{hz}^2}{k_z^4}, \tag{A-24}$$

$$b = -2\left[(s_+^2 - s_-^2)\frac{k_x k_h}{k_z^2} + (s_+^2 + s_-^2)\right],\ c = (s_+^2 - s_-^2)^2. \tag{A-25}$$

Since we have assumed from the beginning that $\partial\phi/\partial z > 0$ in the region of interest, also when $h = 0$ so that $s_+ = s_-$, the choice of root is fixed:

$$\left(\frac{\partial\phi}{\partial z}\right)^2 = \zeta_+^2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}. \tag{A-26}$$

Remark: A very similar argument occurs in ten Kroode (2012), leading up to equation A-10.

We can now assemble the amplitude in the stationary phase approximation (A-3), using the integrand given in equation (A-5), the expression for the Hessian, and the frequency factor: we obtain for the integrand in (A-3)

$$\frac{2\pi s^6}{k_z^2} a_r^2 a_s^2 \left(\frac{\partial\phi}{\partial z}\right)^{-1} |\det \mathrm{Hess}|^{-1/2} = \frac{a_r^2 a_s^2}{k_z^2}\left(\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r}\right)^{-1} \tilde{P}(x, h, z, k_x, k_h, k_z) \tag{A-27}$$

in which

$$\tilde{P} = -\pi s^4 \left(\frac{\partial\phi}{\partial z}\right)^3 \left[\left(\left(\frac{s_-}{s}\right)^2\left(\frac{\partial T_r}{\partial z}\right)^2 + \left(\frac{s_+}{s}\right)^2\left(\frac{\partial T_s}{\partial z}\right)^2\right) + \left(\left(\frac{s_-}{s}\right)^2 + \left(\frac{s_+}{s}\right)^2\right)\left(\frac{\partial T_s}{\partial z}\frac{\partial T_r}{\partial z}\right)\right]^{-1} \tag{A-28}$$

From the defining relations (A-19), (A-21), (A-24), (A-25), and (A-26), it follows that $\tilde{P}$ is homogeneous of order zero in the phase variables k_x, k_h, k_z. The various components of $\tilde{P}$ simplify considerably when $h = 0$ (important because physical reflectivities are supported there): The term in brackets in (A-28) becomes precisely $(\partial\phi/\partial z)^2$, so

$$\tilde{P}(x, z, 0, k_x, k_h, k_z) = -\pi s^4 \frac{\partial\phi}{\partial z} = -2\pi s^5 \frac{k_z^2}{k_{xz}k_{hz}} \tag{A-29}$$

Dividing $\tilde{P}$ by the right hand side in (A-29) produces P, also homogeneous of order zero in the frequency variables and $\equiv 1$ for $h = 0$. Thus we arrive at the integral representation (12) in the theory section, with amplitude:

$$\frac{2\pi s^6}{k_z^2} a_r^2 a_s^2 \left(\frac{\partial\phi}{\partial z}\right)^{-1} |\det \mathrm{Hess}|^{-1/2} = -\frac{2\pi s^5}{k_{xz}k_{hz}} P a_r^2 a_s^2 \left(\frac{\partial\alpha_s}{\partial x_s}\frac{\partial\alpha_r}{\partial x_r}\right)^{-1} \tag{A-30}$$

Note that all factors on the RHS are functions of the phase variables, though some of them are apparently to be determined by ray tracing, as was claimed at the beginning of this appendix.

APPENDIX B

ANALYSIS OF GEOMETRIC AMPLITUDES

As far as we know, the relation explained in this appendix was appeared first in Zhang et al. (2005)). We rederive the relation in this appendix from a different perspective for the convenience of the reader.

The derivation starts from the transport equation. The transport equation in divergence form is :

$$\nabla \cdot (a^2 \nabla \tau) = 0 \tag{B-1}$$

Consider a region R formed by two rays radiating from the same point. Truncate this region with two lines l and L normal to the rays. Denote by α the angle subtended between the first ray and the vertical.

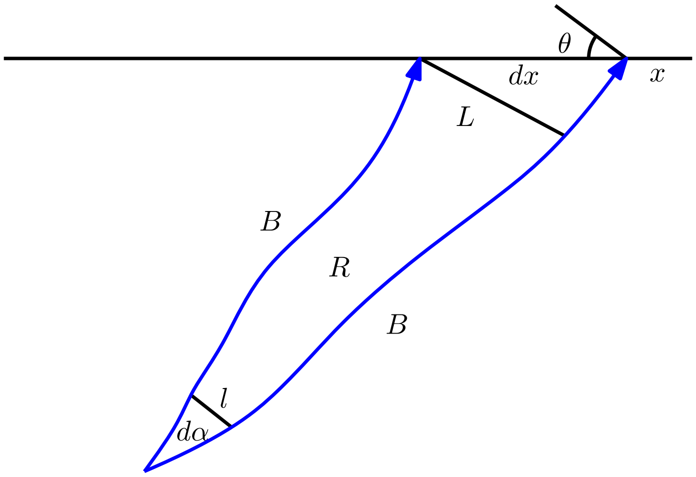


Figure B-1: Sketch of geometric amplitudes analysis appinv/. amp

Apply the 2D divergence theorem in the enclosed region R. We can get

$$\iint\limits_R \nabla \cdot (a^2 \nabla \tau) dR = \oint_C a^2 \nabla \tau \cdot \mathbf{n} dC = 0 \tag{B-2}$$

where $\mathbf{n}$ is the outward normal vector to the boundary $C = B \cup l \cup L$

- For $\mathbf{x}$ on B, the normal vector is perpendicular to the ray, $\nabla \tau \cdot \mathbf{n} = 0$
- For $\mathbf{x}$ on l and L, $\mathbf{n}$ is parallel to the ray, $\nabla \tau \cdot \mathbf{n} = \pm |\nabla \tau|$

Therefore,

$$\int_L A^2 |\nabla \tau| dL = \int_l a^2 |\nabla \tau| dl \tag{B-3}$$

which immediately leads to

$$A^2 = a^2 \frac{V}{v}\frac{dl}{dL} = a^2 \frac{V}{v}\frac{dl}{d\alpha}\frac{1}{\cos\theta}\frac{d\alpha}{dx} \tag{B-4}$$

We have denoted the velocity and amplitude near where the ray starts as v, a and those near where the ray terminates as V, A. In the vicinity of the starting point, the 2D geometric amplitude has the constant-velocity approximation

$$a^2 \simeq \frac{v}{8\pi^2 r} = \frac{v}{8\pi^2}\frac{d\alpha}{dl} \tag{B-5}$$

So in particular for the receiver ray,

$$a_r^2 = \frac{1}{8\pi^2}\frac{v_r}{\cos\theta_r}\frac{d\alpha_r}{dx_r} \tag{B-6}$$

and similarly for the source ray:

$$a_s^2 = \frac{1}{8\pi^2}\frac{v_s}{\cos\theta_s}\frac{d\alpha_s}{dx_s} \tag{B-7}$$

REFERENCES

Bao, G., and W. Symes, 1996, Computation of pseudo-differential operators: SIAM Journal on Scientific Computation, **17**, 416–429.

Baysal, E., D. D. Kosloff, and J. W. C. Sherwood, 1983, Reverse time migration: Geophysics, **48**, 1514–1524.

Bednar, J. B., and C. J. Bednar, 2006, Two-way vs. one-way: a case study style comparison: 76th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2343–2347.

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.

Biondi, B., and A. Almomin, 2014, Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion: Geophysics, **79**, WA129–WA140.

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

Bleistein, N., J. Cohen, and J. Stockwell, 2001, Mathematics of multidimensional seismic imaging, migration, and inversion: Springer.

Bleistein, N., Y. Zhang, S. Xu, G. Zhang, and S. H. Gray, 2005, Migration/inversion: think image point coordinates, process in acquisition surface coordinates: Inverse Problems, **21**, 1715–1744.

Bourgeois, A., B. Jiang, and P. Lailly, 1989, Linearized inversion: A significant step beyond pre-stack migration: Geophysical Journal International, **99**, 435–445.

Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publishers.

Courant, R., and D. Hilbert, 1962, Methods of mathematical physics: Wiley-Interscience, **II**.

De Hoop, M. V., and N. Bleistein, 1997, Generalized Radon transform inversions for reflectivity in anisotropic elastic media: Inverse Problems, **16**, 669–690.

Demanet, L., and L. Ying, 2011, Discrete symbol calculus: SIAM Review, **53**, 71–104.

Dutta, G., Y. Huang, W. Dai, X. Wang, and G. T. Schuster, 2014, Making the most out of the least (squares migration): 84th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 4405–4410.

Gel'fand, I., and G. Shilov, 1958, Generalized Functions: volume 1: Academic Press.

Guillemin, V., and S. Sternberg, 1979, Geometric Asymptotics: American Mathematical Society.

Hou, J., and W. Symes, 2014, An approximate inverse to the extended Born modeling operator: 84th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 3784–3789.

Kern, M., and W. Symes, 1994, Inversion of reflection seismograms by differential semblance analysis: Algorithm structure and synthetic examples: Geophysical Prospecting, **99**, 565–614.

Lameloise, C.-A., H. Chauris, and M. Noble, 2014, Improving the gradient of the image-domain objective function using quantitative migration for a more robust migration velocity analysis: Geophysical Prospecting, **62**, (in press).

Liu, Y., W. Symes, and Z. Li, 2014, Inversion velocity analysis via differential semblance optimization: Presented at the 76th Annual Meeting, Extended Abstracts, European Association of Geoscientists and Engineers.

Liu, Y., W. W. Symes, Y. Huang, and Z. Li, 2013, Linearized extended waveform inversion via differential semblance optimization in the depth oriented extension: 83rd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 4869–4874.

Loewenthal, D., and I. Mufti, 1983, Reversed time migration in spatial frequency domain: Geophysics, **48**, 627–635.

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

Nemeth, T., C. Wu, and G. Schuster, 1999, Least-squares migration of incomplete reflection data: Geophysics, **64**, 208–221.

Nolan, C. J., and W. Symes, 1996, Imaging and conherency in complex structure: 66th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 359–363.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Rickett, J., and P. Sava, 2002, Offset and angle-domain common image-point gathers for shot profile migration: Geophysics, **67**, 883–889.

Sava, P., and S. Fomel, 2003, Angle-domain common-image gathers by wavefield continuation methods: Geophysics, **68**, 1065–1074.

Stolk, C. C., M. V. de Hoop, and T. P. M. Op 't Root, 2009a, Reverse time migration inversion from single-shot data: 79th Annual International Meeting, Expanded

Abstracts, Society of Exploration Geophysicists, 2995–2999.

Stolk, C. C., M. V. de Hoop, and W. Symes, 2009b, Kinematics of shot-geophone migration: Geophysics, **74**, WCA18–WCA34.

Stolk, C. C., and W. Symes, 2004, Kinematic artifacts in prestack depth migration: Geophysics, **69**, 562–575.

Sun, D., and W. Symes, 2012, Waveform inversion via nonlinear differential semblance optimization: 82nd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, SI3.3.

Symes, W., 1998, Mathematics of reflection seismology. (available at http://www.trip.caam.rice.edu).

———, 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790.

———, 2009, The seismic reflection inverse problem: Inverse Problems, **25**, 123008:1–24.

Tang, B., S. Xu, and Y. Zhang, 2013, 3D angle gathers with plane-wave reverse time migration: Geophysics, **78**, S117–S123.

Taylor, M., 1981, Pseudodifferential Operators: Princeton University Press.

ten Kroode, F., 2012, A wave-equation-based Kirchhoff operator: Inverse Problems, 115013:1–28.

Whitmore, N. D., 1983, Iterative depth migration by backwards time propagation: 53rd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, S10.1.

Xu, S., Y. Zhang, and B. Tang, 2011, 3D angle gathers from reverse time migration: Geophysics, **76**, S77–S92.

Zhang, Y., and J. Sun, 2008, Practical issues of reverse time migration-true-amplitude gathers, noise removal and harmonic-source encoding: Presented at the 76th Annual Meeting, Extended Abstracts, European Association of Geoscientists and Engineers.

Zhang, Y., J. Sun, and S. H. Gray, 2007, Reverse time migration: amplitude and implementation issues: 77th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2145–2149.

Zhang, Y., G. Zhang, and N. Bleistein, 2005, Theory of true-amplitude one-way wave equations and true-amplitude common-shot migration: Geophysics, **70**.

Linearized Extended Waveform Inversion -PhD Thesis Proposal

Yin Huang, Rice University
yin.huang@rice.edu
Sep 30, 2014

ABSTRACT

Seismic full waveform inversion (FWI) is a nonlinear least squares data-fitting problem, which may have many stationary points apart from its global minimizer. A widely studied variant groups model parameters into coefficients of long and short scale shape functions (relative to a typical seismic wavelength), and linearizes the influence of the short scale parameters on predicted data. The resulting separable nonlinear least squares problem can be reduced to a problem in long scale parameters only, by solution of the quadratic minimization for the short scale parameters. The resulting problem still has many local minima. Then we introduce suitable nonphysical degrees of freedom to the short scale parameters, and augment the least squares objective with an appropriate quadratic penalty for the nonphysical parameters. The reduced problem for this extended separable FWI problem tends to have only stationary value near the global minimum. Therefore efficient local optimization algorithms (variants of Newtons method) could give the global optimum. Analytic properties of the prediction operator lead to efficient approximations of the reduced gradient. The goal of this thesis is to develop and demonstrate some of these efficient gradient calculations, and embed them in a robust optimization algorithm for solution of the seismic inverse problem.
We use acoustic constant density modeling as an example to illustrate the theory: the frequency stability, the shape of the reduced objective function and its relation with parameter choices and to show inversion results from synthetic data. These studies also suggest directions for future work.

INTRODUCTION

Seismic full waveform inversion (FWI) is a tool that is used to infer the interior structure of the earth from observed seismic waves. It is defined as a least squares problem with the prediction operator related to the solution operator of the initial and boundary values problem for a wave equation with spatially varying coefficients. The structure of the earth is described by these variable coefficients, which are called models. In contrast to the fact that wave propagation problem is usually called forward modeling, the problem of finding model parameters to predict the observed data which is recorded along the surface of the earth or the top or the bottom of the sea is called *full waveform inversion* (FWI).

In order to know the subsurface structure of a local part of the earth, exploration geophysisists usually perform many experiments. Each seismic experiment uses a localized energy source (”shot”) to initiate the motion of the ground (generate waves to propagate through the earth). Part of these waves will be reflected when they encounter discontinuities (reflectors) of the earth and another part will be refracted due to the earth model variation at different places. Part of these reflected and refracted waves will travel back to the surface and recorded by geophones (land) or hydrophones (marine). The recorded data is usually the displacement of particles or pressure field. Data is usually recoded separately for each shot.

Noises in data contribute to the inconsistency of these redundant data. Due to high redundancy in the observed data, FWI is over-determined, and because of the inconsistency of the data, it is usually difficult to drive FWI objective to a small value. The observed data contain a set of time series at different recording locations for different sources or shots. These time series are band-limited since they lack usually information at both very low and high frequency components. The wavelength of these time series are finite and bounded away from zero. Data could be divided into subsets (”gathers”) based on positions of shots that evoke the propagation of waves. A localized change in wave velocity sufficient to induce a travel-time shift by a wavelength between some source receiver pairs typically affects the data fit for some shots but not for others, and may generate spuriously good local fits (“cycle skipping”). Thus the least-squares data fitting function has many local minima far from its global minimum, which is equivalent to say that FWI could not give a good estimated model unless the initial model guess is close enough to the true model. The density of local minima increases, and the size of the basin of attraction of each local minimum decreases, as the central frequency of the data increases. That is, the least squares objective function is not stable with respect to data frequency content.

If we use different models to fit different shots data, the least squares objective function becomes much easier to drive to a global minimizer. Seismic data are usually recorded at a much smaller subset of the spatial domain of interest, such as only record data at a line (2D seismic) or a surface (3D seismic) along the top boundary of the model, or sometimes drill wells downward and record data inside wells. This is restricted by economic or environmental reasons. Thus one shot data is not enough to constrain the model. It is usually easy to adjust wave velocity locally to change the arrival time to each receiver position. In principle, FWI objective function with only one shot data could have minimum value 0. Fitting different shots data with different models results in a shot-dependent model (*model with shot coordinate extension*). A shot-dependent model is not physical, since the earth is unique and does not change for different experiments within seismic survey time; it is an example of an *extended* *model*. There are many other types of extension (ways of introducing nonphysical degrees of freedom into the model parameters). Inversion for a shot-dependent model (or any extension) by data fitting is underdetermined. Data will be easily over-fitted in this case, i.e. error between predicted data and observed data is smaller than noise level in the observed data. An extra constraint must be imposed to model parameters to suppress the nonphysical degrees of freedom, and thus recover some control over the model estimate. This extra constraint is usually referred to as *semblance*, which has a value zero at a physical model for some choices, or *stacking power* , which has a maximum value at physical model for other choices.

FWI theory is first introduced by Tarantola (1984) and then Gauthier et al. (1986) shows that FWI is computationally feasible and the ”Camembert” example, created in this paper shows the local minima phenomenon. Jannane and coworkers show that intermediate wavelengths change in model is hard to be captured in seismic data, due to acquisition geometries (the arrangement of sources and receivers during seismic experiments) and band-limited feature of the source (Jannane et al., 1989). Santosa and Symes did a comprehensive study of the capacity of the least-squares inversion for band-limited common shot data (data divided according to different source) with a layered velocity model (Santosa and Symes, 1989). More theories and applications of FWI can be found in this review paper by Virieux and Operto (2009). The method discribed in this paper aims at avoiding the local minima problem of FWI and producing a frequency stable objective function, the key to which is the extended model concept.

Inversion based on full wave extended modeling has been studied by a number of authors. Sun proposed a low frequency control method for extended full waveform inversion: use an artificial low-frequency data as control variables to minimize the non physicality of the extended model (Sun, 2012). The constrain has two terms: first one is the extended least squares misfit function between the nonlinear forward map (with low frequency data predicted) and the observed data plus a low-frequency data, which is the control variable; second one is a semblance penalty term. Numerical results suggest that the method is potential to recover the model from kinematically wrong starting model. Instead of minimizing the semblance term, Biondo and Almomin minimize the sum of a semblance penalty term and the extended least squares misfit function, and show numerically that this method is able to converge to a global minimal (Biondi and Almomin, 2012). There are still a lot of research need to be done to fully evaluate the extended FWI (could it really avoid cycle skipping? to what extend? why? and test it using more numerical experiments on complex models) and to make it practical (develop computationally efficent workflow). Although we will not disscuss more of this topic in this proposal, it could be an interesting future work. This extended FWI has a very similar form with our objective function and the semblance penalty term is pretty much the same as what we use in the proposal.

At the same time, the bulk of work on the extended model concept has concerned *linearized extended modeling and related inversion algorithms*, which is also the main focus of this proposal. For linearized extended modeling, the model is split into long-scale components (smoothly varying part of the model, referred to as the *velocity*) and short-scale components (oscillatory part of the model, referred to as the *reflectivity*). The recovery of the velocity is very important but usually difficult. Once the velocity is correct, the estimation of reflectivity is usually relatively easy. After scale separation of the model, the long scale components remain physical. Only the oscillatory components are extended (allowed to depend on parameters other than spatial coordinates). The predicted data is modeled via linearization (Born approximation), viewing the short-scale components as perturbation about the long-scale model. Both the long scale (velocity) and the short scale components (reflectivity) are to be determined through optimization methods. Differential semblance optimization (DSO) was first introduced by Symes (1986), in which the objective is to minimize how far away the reflectivity model is from a physical model. Here the reflectivity model depends on the smooth velocity model in some sense. Symes and Carazzone (1991) formulated the objective function as the sum of a data misfit term and a differential semblance term, and we use the same formulation as an intermediate function to get our objective function in this proposal, as Kern and Symes (1994) did. Formulations specific for layered earth model and plane-wave data were derived in Symes and Carazzone (1991) and both synthetic and field data examples shown that the method could produce reasonably good velocity and reflectivity model.

Due to the fast development in high performance computing, more and more researchers became interested in this subject in the most recent decade and various variations of this method have been studied. Most of these works focused on differential semblance objective function, a modified version of it or linear combinations of them. Some of them were dealing with surface offset model extension, which allow the model to depend on the distance between positions of sources and receivers (Chauris and Noble, 2001; Mulder and ten Kroode, 2002). Large percentage of most recent works were on subsurface offset model extension (Shen, 2012; Shan and Wang, 2013; Weibull and Arntsen, 2013; Tang and Biondi, 2011). Results from both subsurface offset domain and angle domain (reflection angle at discontinuities of the model) model extensions were shown in Shen and Symes (2008). Biondi and Zhang (2013) proposed to maximize the stack of the angle domain common image gathers (each gather is a function of depth and reflection angle, and is corresponding to a slice of the extended reflectivity at a position on the surface), by using an intermediate parameter to project the non-flatness of gathers along the angle axis to the update of the model. Differential semblance method was also applied to recover models with anisotropy (Weibull and Arntsen, 2014). Most of these reviewed articles above used short scale model (reflectivity) computed either from some kind of imaging operator, for example, reverse time migration operator (the adjoint of the linearized forward map), or by high frequency asymptotic approximate. The common feature of them is that their reflectivities inside the semblance term do not fit the data. Works using the computed short scale model (reflectivity) that could fit the data shown only preliminary numerical results (Kern and Symes, 1994; Liu et al., 2013). This proposal also uses the inverted reflectivity that could fit the data, at a given long scale model. We will study the efficeincy of computing the short scale model using iterative methods and suggest several potential speed up strategies. The implementation of an inversion workflow will use shot coordinate model extension, which is not widely used in recent literature.

Some papers parametrized the long scale smooth velocity model by a B-spline. This parametrization reduced the dimension of the smooth velocity space and help to control smoothness automatically by changing the coarseness of the grid space (Chauris and Noble, 2001; Shen and Symes, 2008). We choose to enforce the smoothness to the inverted velocity model by assume the velocity is in a Sobolev space. The linearized extended waveform inversion method is designed to deal with primary only reflection data: first order approximation to the forward modeling operator. It was also tested with the full waveform data: all parts of the observed data or the data generated by prediction operator in synthetic case, and gave reasonably good results (Shen and Symes, 2008; Weibull and Arntsen, 2013). An extensive reference list may be found in Symes (2008b), where it is argued that these methods address the inverse problem implicitly posed by migration velocity analysis. Biondi and Almomin (2014) presented a nested optimization scheme to solve a linearized extended inversion. They updated the model by first updating separately the background velocity and reflectivity using a scale mixed gradient, and then used them to update the non-separated model. Chauris and Plessix (2013) extended the DSO method to data with multiple reflections and a 1-D example was shown in the paper. Dealing with multiple reflections is a very important topic, but is outside the scope of this proposal.

Extended modeling relaxes the data fit criterion; a physicality constraint (for example DSO) suppresses the non-physical extension. The tension between the two is resolved at a physical model fitting the data. An objective function combining penalties for data misfit and non-physicality would seem an obvious approach to inversion, but contains the data misfit function so is frequency-dependent and just as likely to suffer from spurious local minima as is the standard least-squares objective.

The key to frequency-stable smoothness and unimodality is the *reduced objective*, which comes from a linearized data misfit and a linear inversion for short scale model (reflectivity) and thus depends only on the velocity (Kern and Symes, 1994; Liu et al., 2013). It is this reduced objective which is smooth and frequency-stable, though only for certain choices of physicality constraints (Stolk and Symes, 2003). The reduced objective may also be viewed as an example of variable projection (van Leeuwen and Mulder, 2009).

The crucial fact, underlying both the smoothness and the unimodality of the objective function studied here, turns out to be a very special property of the normal operator, or the composition of the adjoint of the linearized forward map with itself evaluated at a long scale model: it is (essentially, and under some circumstances) a *pseudodifferential* operator (Taylor, 1981), a type of oscillatory integral operator for which the high-frequency oscillations (or singularities) are in the same spatial position for both input and output, or the wavefront set of the output is a subset of the wavefront set of the input. This observation underlies asymptotic ("Kirchhoff") inversion methods (Stolk, 2000; Burridge et al., 1998; ten Kroode et al., 1998; Nolan and Symes, 1997; Beylkin and Burridge, 1990; Rakesh, 1988; Bleistein, 1987; Beylkin, 1985), but is equally important for understanding the behavior of inversion algorithms not based explicitly on ray computations. Smoothness of the extended inversion objective functions discussed here required that the constraint on non-physical parameters, essential for turning extended modeling into a tool for inversion, must also be pseudo-differential (Stolk and Symes, 2003).

In this proposal, we will review the extended modeling concept, then explain what is the reduced objective function, how do we get this objective. Then we will derive the gradient of this reduced objective function in a Sobolev space and use acoustic constant density modeling as an example to illustrate the frequency stability, the shape of the reduced objective function and its relation with parameter choices and to show inversion results from synthetic data.

Extended modeling has an abstract form that encompasses all of its concrete instances. The first section to follow introduces extended modeling in this abstract form. The second section presents our major results, formulated abstractly, *how* smooth objective functions of model parameters arise from data misfit and extended modeling, augmented with a particular type of constraint. And the third section illustrates them numerically via the acoustic shot coordinate model extension. The proposal ends with a discussion and proposed project.

EXTENDED MODELING

Denote by $M = \{m(\mathbf{x})\}$ the physical model space and by $\bar{M} = \{\bar{m}(\mathbf{x}, h)\}$ the extended model space, which contains the physical models as a subspace. The variable h is a parameter, such as shot coordinate, offset, ray parameter (for plane wave data) or parameter vector, subsurface offset, or scattering angle, which characterizes additional degrees of freedom in the extended model space, over and above position in the subsurface. Denote by D the data space. Assume the physical domain of the earth that we are interested in is $\Omega \subset \mathbf{R}^n$, with $n = 2$ or 3. We require $M = L^2(\Omega)$, $\bar{M} = L^2(\Omega \times \mathbf{R})$ and data space is $D = L^2([0, T] \times \Sigma_{s,r})$, with $\Sigma_{s,r} = \{(\mathbf{x}_s, \mathbf{x}_r)\}$ the collection of source receiver position pairs. All of these spaces are Hilbert spaces, each with an appropriate (possibly weighted) inner product.

M is identified with a subset of $\bar{M}$ - the identification is characteristic for each type of extended model. For the shot coordinate extension mentioned in the introduction, $\bar{m}(\mathbf{x}, h)$ defines a member of M if it's independent of h. For the subsurface offset extension (Prucha et al., 1999), physical models are focused at $h = 0$, that is, take the form $\bar{m}(\mathbf{x}, h) = m(\mathbf{x})\delta(h)$.

Denote by $F : M \mapsto D$ the forward map, or prediction operator, and by $\bar{F} : \bar{M} \mapsto D$ the extended forward map. $\bar{F}$ is an extended modeling operator because for $m \in M$,

$$\bar{F}[m] = F[m].$$

The abstract inverse problem may be formulated: given data $d \in D$, find $m \in M$ so that

$$F[m] \simeq d \Longleftrightarrow \bar{F}[\bar{m}] \simeq d, \bar{m} \in M. \tag{1}$$

The model is separated into a long scale background model m_l and a short-scale reflectivity model δm: $m \simeq m_l + \delta m$. The reflectivity is treated as a perturbation of the background model for modeling purposes:

$$F[m] \simeq F[m_l] + DF[m_l]\delta m.$$

Here DF denotes the linearized forward map, or Born approximation.

The abstract linearized inverse problem may be formulated as: given data $d \in D$, find $m_l, \delta m \in M$ so that

$$DF[m_l]\delta m \simeq \delta d = d - F[m_l]. \tag{2}$$

The same is done with extended models, with one difference: the background model will always be physical. Thus $\bar{m} \simeq m_l + \delta\bar{m}$, and

$$\bar{F}[\bar{m}] \simeq F[m_l] + D\bar{F}[m_l]\delta\bar{m}$$

That is, we approximate the extended forward map with its Born approximation at a physical background model, but with extended reflectivity. Thus the extended linearized inverse problem is: given d, find $m_l \in M, \delta\bar{m} \in \bar{M}$ so that

$$D\bar{F}[m_l]\delta\bar{m} \simeq \delta d = d - F[m_l]. \tag{3}$$

As noted in the introduction, and illustrated below, this problem is highly underdetermined.

LINEARIZED EXTENDED WAVEFORM INVERSION

Reduced Objective Function

As mentioned in the introduction, an additional ingredient is needed to drive extended models toward physical (non-extended) models. One approach, by far the most explored, is to introduce an operator which “measures” physicality by mapping physical models to the zero vector, and penalizing the size of its output. Such operators have come to be called *annihilators* (Brandsberg-Dahl et al., 2003).

Thus introduce an operator A on extended model space $\bar{M}$, satisfying the abstract annihilator property:

$$A\delta m = 0 \text{ for all } \delta m \in M.$$

We will introduce a concrete annihilator in the next section for shot coordinate model extension.

The solution to the basic linearized inverse problem (2) is a solution of the extended linearized inverse problem for which $\delta\bar{m}$ is physical, hence mapped to the zero vector by A. Thus (3) is equivalent to the system

$$\begin{aligned} D\bar{F}[m_l]\delta\bar{m} &\simeq \delta d \\ A\delta\bar{m} &\simeq 0. \end{aligned} \tag{4}$$

Define:

$$J[m_l, \delta\bar{m}] = \frac{1}{2}\|D\bar{F}[m_l]\delta\bar{m} - \delta d\|^2 + \frac{\alpha^2}{2}\|A\delta\bar{m}\|^2, \tag{5}$$

The weight α controls emphasis on physicality: as $\alpha \to \infty$, the minimizer of $J[m_l, \cdot]$ tends to the solution δm of (2), interpreted as a least squares problem, for fixed m_l, d.

The reduced objective $\tilde{J}[m_l]$ is the least value attained by $J[m_l, \delta\bar{m}]$ over the model space of reflectivity $\delta\bar{m}$.

$$\tilde{J}[m_l] = \min_{\delta\bar{m}} J[m_l, \delta\bar{m}]. \tag{6}$$

Here minimum is used instead of infimum when the block operator $\begin{pmatrix} D\bar{F}[m_l] \\ \alpha A \end{pmatrix}$ is bounded and coercive, which is not usually true for many choice of A and some kind of preconditioning and regularization may need, which will be discussed in details in the Proposed Project section. We assume temporarily that the block operator has been preconditioned and regularized and proceed as if it is bounded and coercive.

The analysis of $\tilde{J}$ begins with the observation that it is the minimum value of J over $\delta\bar{m}$ for fixed m_l, δd. Since J is quadratic in $\delta\bar{m}$, its solution is the solution of a linear system. Define the normal operator (or Hessian)

$$N[m_l] = D\bar{F}[m_l]^T D\bar{F}[m_l] + \alpha^2 A^T A.$$

Then the minimum value of J is attained at

$$\delta\bar{m} = N[m_l]^\dagger D\bar{F}[m_l]^T \delta d. \tag{7}$$

For some choice of A, $N[m_l]$ is invertible and in this case $N[m_l]^\dagger = N[m_l]^{-1}$. In the case that $N[m_l]$ is not invertible, $N[m_l]^\dagger$ is a pseudo-inverse. The numerical solution of $\delta\bar{m}$ involves an iterative process, which was referred to as PICLI method in Ehinger and Lailly (1993) for shot coordinate depended operator A.

Using formula (7) we can write an explicit expression for $\tilde{J}[m_l]$:

$$\tilde{J}[m_l] = \frac{1}{2}\|(D\bar{F}[m_l]N[m_l]^{\dagger}D\bar{F}[m_l]^T - I)\delta d\|^2 +$$

$$\frac{\alpha^2}{2}\langle\delta d, D\bar{F}[m_l]N[m_l]^{\dagger}A^T AN[m_l]^{\dagger}D\bar{F}[m_l]^T\delta d\rangle. \tag{8}$$

The significance of the formula (8) lies in a fact about the normal operator $N[m_l]$: under some circumstances, it is closely related to a *pseudo-differential operator*, a type of oscillatory integral operator. This is so principal because the modeling Hessian $D\bar{F}[m_l]^T D\bar{F}[m_l]$ has this property when the source of the forward map $\bar{F}$ is an impulse. This property of the normal operator leads to the smoothness of the reduced objective function (8). Numerical results will be included in the Example section for illustration. This relationship is contingent on kinematic and dynamic assumptions on the model and data: for example, for some extensions (for example, shot-record) multiple ray paths connecting sources and receivers with scattering points may not occur (Rakesh, 1988; Symes, 1998).

Gradient of Reduced Objective Function

Since from last section, the reduced objective function (8) is smooth on m_l, gradient method could be applied to minimize (8).

Given a perturbation dm_l, we get the directional derivative by chain rule:

$$D\tilde{J}[m_l]dm_l = D_{m_l}J[m_l,\delta\bar{m}]dm_l + D_{\delta\bar{m}}J[m_l,\delta\bar{m}]D_{m_l}\delta\bar{m}dm_l.$$

If $\delta\bar{m}$ satisfies equation (7) , the second term in the above equation vanishes and we get

$$D\tilde{J}[m_l]dm_l = \langle D^2\bar{F}[m_l][\delta\bar{m}, dm_l], D\bar{F}[m_l]\delta\bar{m} - \delta d\rangle.$$

The second order derivative $D^2\bar{F}[m_l]$ is the Hessian of the extended forward operator $\bar{F}[m_l]$. It takes two arguments and is a bilinear operator. Then define a bilinear operator $D^2\bar{F}[m_l]^T$ on space $\bar{M} \times D$ such that it satisfies for $dm_l \in M$, $d\bar{m} \in \bar{M}$ and $d \in D$

$$\langle D^2\bar{F}[m_l][d\bar{m}, dm_l], d\rangle = \langle dm_l, D^2\bar{F}[m_l]^T[d\bar{m}, d]\rangle. \tag{9}$$

$D^2\bar{F}[m_l]^T$ is called tomographic operator, or WEMVA operator in geophysics literature (Biondi and Sava, 2004; Biondi and Almomin, 2012). Note that both $D^2\bar{F}[m_l]$ and $D^2\bar{F}[m_l]^T$ depend on the long scale smooth model parameter.

Then the directional derivative becomes

$$D\tilde{J}[m_l]dm_l = \langle dm_l, D^2\bar{F}[m_l]^T[\delta\bar{m}, D\bar{F}[m_l]\delta\bar{m} - \delta d]\rangle,$$

In the L^2 sense, the gradient of $\tilde{J}[m_l]$ becomes

$$\nabla\tilde{J}[m_l] = D^2\bar{F}[m_l]^T[\delta\bar{m}, D\bar{F}[m]\delta\bar{m} - \delta d]. \tag{10}$$

NOTE: in the above derivation, we assume equation (7) is solved exactly. However, in practice, it is usually not true, since the solution $\delta\bar{m}$ are solved iteratively. Thus, we will perform gradient accuracy tests in the Example section.

Based on scale separation and various other reasons, we require the background model to be smooth in some sense, and thus the gradient too. One way to enforce smoothness is using B-spline to parametrize the velocity model into a low-dimensional space as a lot of researchers did. The choice here is to impose smoothness by a different inner product in the smooth velocity space.

Assume we consider the problem in 2-dimensional space. The discrete anisotropic Laplace operator on dm is defined as

$$(Ldm)_{i,j} = \omega_x^2\frac{dm_{i+1,j} - 2dm_{i,j} + dm_{i-1,j}}{dx^2} + \omega_z^2\frac{dm_{i,j+1} - 2dm_{i,j} + dm_{i,j-1}}{dz^2}.$$

If we choose the boundary condition appropriately, the operator L could be diagonalized by the discrete Fourier operator $\mathcal{F}$, which is unitary. That is there exists an diagonal operator Σ such that

$$\mathcal{F}^T\Sigma\mathcal{F} = L.$$

L is a uniform elliptic operator and $-L$ is a nonnegative operator with Dirichlet boundary condition. Thus the square root of $I - L$ exists and is invertible, with I being the identity operator. Define

$$\Lambda = (I - L)^{\frac{1}{2}}.$$

Thus $\Lambda = \mathcal{F}^T(I - \Sigma)^{\frac{1}{2}}\mathcal{F}$ and $(I - \Sigma)^{\frac{1}{2}}$ has diagonal approximately $(1 + \omega_x^2k^2dx^2 + \omega_z^2l^2dz^2)^{\frac{1}{2}}$ for small wave number kdx, ldz (number of waves that exist over the distance dx for x direction). Define the s inner product on the Sobolev space of the velocity model:

$$\langle dm_1, dm_2\rangle_s = \langle\Lambda^s dm_1, \Lambda^s dm_2\rangle,$$

where $\langle\cdot,\cdot\rangle$ is the ordinary inner product on the model space. Let Q be an operator which maps the model dm to some finite dimensional function space by discretization. Q^T is its dual operator in L^2 sense, i.e.

$$\langle Qdm, \phi\rangle_{l^2} = \langle dm, Q^T\phi\rangle.$$

with $\langle\cdot,\cdot\rangle$ the ordinary L^2 norm in L^2 function space and $\langle\cdot,\cdot\rangle_{l^2}$ the l^2 inner product in the finite dimensional space. Then we have

$$\langle Qdm, \phi\rangle_{l^2} = \langle dm, Q^*\phi\rangle_s = \langle \Lambda^s dm, \Lambda^s Q^*\phi\rangle = \langle dm, \Lambda^{2s} Q^*\phi\rangle$$

Thus the adjoint operator of Q in the s-inner product sense is

$$Q^* = \Lambda^{-2s} Q^T.$$

Using the s-inner product, the gradient of $\tilde{J}$ becomes

$$\nabla \tilde{J}[m_l] = \Lambda^{-2s} D^2 \bar{F}[m_l]^T [\delta\bar{m}_k, D\bar{F}[m]\delta\bar{m}_k - \delta d]. \tag{11}$$

We will test how accurate is this gradient numerically in the Example section.

EXAMPLE: EXTENDED 2D CONSTANT DENSITY ACOUSTICS

The simplest useful model for simulation of seismic data is the constant density acoustic wave equation. The model space is a set of velocities, or more conveniently, squared velocities: $M = \{c^2(\mathbf{x})\}$. For this problem, we use shot coordinate $\mathbf{x}_s$ as the extended coordinate and the extended model space is $\bar{M} = \{\bar{c}^2(\mathbf{x}, \mathbf{x}_s)\}$. The pressure field is causal, and solves the constant density acoustic wave equation. The right-hand side represents an isotropic point radiator source with time-dependence $w(t)$:

$$\begin{aligned} \left(\frac{\partial^2}{\partial t^2} - c^2(\mathbf{x})\nabla^2_{\mathbf{x}}\right) u(\mathbf{x}, \mathbf{x}_s, t) &= \delta(\mathbf{x} - \mathbf{x}_s) w(t), \\ u(\mathbf{x}, \mathbf{x}_s, t) &= 0,\ t << 0. \end{aligned} \tag{12}$$

The value of both the extended and non-extended forward map for the model $c^2(\mathbf{x})$ is the result of sampling the pressure field at a prescribed set of receiver points for the various source positions in the survey:

$$\bar{F}[c^2] = F[c^2] = \{u(\mathbf{x}_r, \mathbf{x}_s, t)\}.$$

Linearization, or extended Born approximation, results from sampling a pressure field perturbation δu of a background pressure field u_l, the solution of (1) with the background squared velocity $c^2 = c_l^2$, resulting from an extended squared-velocity perturbation $\delta\bar{c}^2(\mathbf{x}, \mathbf{x}_s)$. Thus the extended Born approximation model space is $M = \{c_l^2(\mathbf{x}), \delta c^2(\mathbf{x}, \mathbf{x}_s)\}$. The perturbational pressure δu solves

$$\begin{aligned} \left(\frac{\partial^2}{\partial t^2} - c_l^2(\mathbf{x})\nabla^2_{\mathbf{x}}\right) \delta u(\mathbf{x}, \mathbf{x}_s, t) &= \delta\bar{c}^2(\mathbf{x}, \mathbf{x}_s)\nabla^2 u_l(\mathbf{x}, \mathbf{x}_s, t), \\ \delta u(\mathbf{x}, \mathbf{x}_s, t) &= 0,\ t << 0. \end{aligned} \tag{13}$$

So the Born approximation is $D\bar{F}[c_l^2]\delta\bar{c}^2 = \{\delta u(\mathbf{x}_r, \mathbf{x}_s, t)\}$. Note that $D\bar{F}$ is linear in $\delta\bar{c}^2$, but nonlinear in c_l^2.

The second order derivative of $\bar{F}[c^2]$ comes from sampling the perturbed pressure field $\delta^2 u$ of the above Born pressure field δu, with an extended perturbation $\delta_2\bar{c}^2(\mathbf{x}, \mathbf{x}_s)$. The pressure field $\delta^2 u$ solves

$$\begin{aligned}\left(\frac{\partial^2}{\partial t^2} - c_l^2(\mathbf{x})\nabla^2\right)\delta^2 u(\mathbf{x}, \mathbf{x}_s, t) &= \delta\bar{c}^2(\mathbf{x}, \mathbf{x}_s)\nabla^2\delta_2 u(\mathbf{x}, \mathbf{x}_s, t) + \delta_2\bar{c}^2(\mathbf{x}, \mathbf{x}_s)\nabla^2\delta u(\mathbf{x}, \mathbf{x}_s, t), \\ \delta^2 u(\mathbf{x}, \mathbf{x}_s, t) &= 0,\ t << 0, \end{aligned} \tag{14}$$

with $\delta_2 u$ solves equation (13) with $\delta\bar{c}^2$ replaced by $\delta_2\bar{c}^2$. So $D^2\bar{F}[c_l^2][\delta\bar{c}^2, \delta_2\bar{c}^2] = \{\delta^2 u(\mathbf{x}_r, \mathbf{x}_s, t)\}$. Note that $D^2\bar{F}$ is linear in both $\delta\bar{c}^2$ and $\delta_2\bar{c}^2$, but nonlinear in c_l^2.

Finite difference method is used to discretize the wave equation. We use 2-nd order accuracy in time and 4-th order accuracy in space. We refer to a C function, that compute the next time step wavefield from current and past time step wavefields, as the time step function. The time step function of the implementation for both $D\bar{F}[c_l^2]$, $D^2\bar{F}[c_l^2]$ and their adjoints $D\bar{F}[c_l^2]^T$, $D^2\bar{F}[c_l^2]^T$, are generated using automatic differentiation tool TAPENADE (Hascoët and Pascual, 2004) with the original finite difference wave propagation code of $\bar{F}[c_l^2]$ provided. These time step functions are implemented in IWAVE, which is a framework that provides i/o, job control and parallelization. All results in this section is obtained using IWAVE framework and RVL optimization software (*https://svn.code.sf.net/p/rsf/code/trunk/trip/*).

Since physical models do not depend on $\mathbf{x}_s$, a feasible choice of annihilator for this shot coordinate model extension is (Kern and Symes (1994))

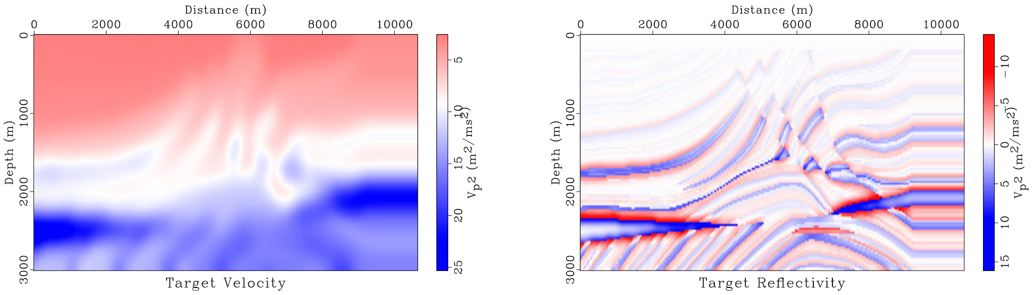
$$A = \frac{\partial}{\partial \mathbf{x}_s}.$$

In fact, as shown by Stolk and Symes (2003), this is essentially the *only* choice of annihilator that will lead to a smooth reduced objective function.

The Marmousi model (Bourgeois et al., 1991) is used in the next two subsections. This model is separated into smoothed long scale background model m_t and δm (see figure 1(a) and 1(b)).

Property of the Normal Operator $N[m_l]$

We want to show the key feature of the normal operator $N[m_l]$ in this subsection using numerical examples. With the choice of differential semblance operator $A = \dfrac{\partial}{\partial x_s}$, $N[m_l]$ is closely related to a *pseudo-differential operator*. The relation is closest when $\omega(t) = \delta(t)$, but holds more generally. $D\bar{F}[m_l]^T$ is usually called *reverse time migration operator* (RTM) or an *imaging operator*. It takes data as an input and produce



(a) (b)

Figure 1: (a) smoothed Marmousi model m_t; (b) the reflectivity δm. lewi/Fig/marm csqsm,dcsq

an image of δm, which is referred to as *migrated image* and shows the approximate positions of singularities in δm with differences in amplitude and some spatial frequency components. The accuracy of the approximation depends on how close is m_l to the true velocity which generates the data.

Figures 2 illustrates the key feature of pseudo-differential operators, namely that *they do not move reflectors (singularities in the model function)*. Panels (a) - (g) of Figure 2 shows $N[m_l]\delta m$, with fixed δm given in Figure 1(b), and $m_l = \sigma m_t$, where m_t is given in Figure 1(a) and σ changes from 0.7 to 1.3.

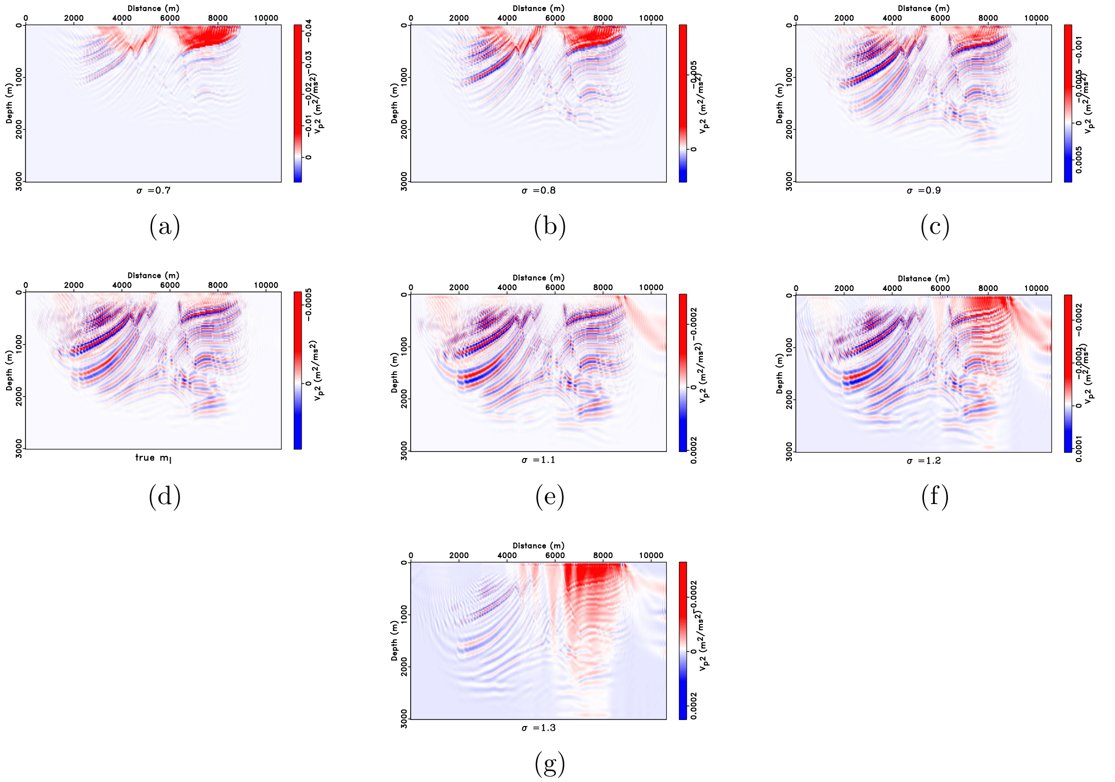
Figure 3 shows $N[m_t]\delta m$ at the correct velocity m_t with different source frequencies: Ricker1 wavelet with peak frequency 5 Hz for (a), 10 Hz for (b) and 15 Hz for (c). Notice that the equal phase surfaces in the migrated image, representing approximate reflector locations, do not move: only the amplitude changes.

This feature of the Hessian is also responsible for the success of approximate inversion by scaling, see for example (Symes, 2008a).

Figures 2 and 3 lend plausibility to the assertion that $N[m_l]\delta m$ is smooth as a function of m_l. Changing the frequency content of the data (or source) will change the spatial frequency of the reflector images, but the amplitude scaling between δm and its image remains the same, independent of frequency. Therefore the size of the derivatives of $N[m_l]$ with respect to m_l also remains the same independent of frequency.

Scan Tests of the Reduced Objective Function

The data are computed with 60 shots starting from 3 km, with spacing 100 meters and 12 meters below the sea surface. 96 receivers are placed behind each shot, with offset 200 meters between the first receiver and a shot , 25 meters spacing between

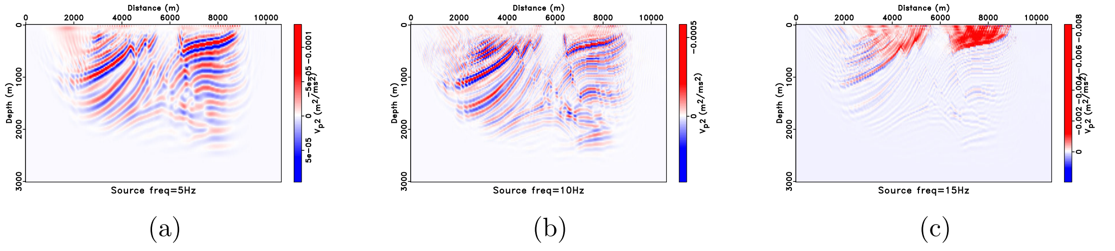


(a) (b) (c)

(d) (e) (f)

(g)

Figure 2: $N[m_l]\delta m$ with $m_l = \sigma m_t$ (a) $\sigma = 0.7$, (b) $\sigma = 0.8$, (c) $\sigma = 0.9$, (d) $\sigma = 1.0$, (e) $\sigma = 1.1$, (f) $\sigma = 1.2$, (g) $\sigma = 1.3$. lewi/Fig/marm mcsq07,mcsq08,mcsq09,mcsq,mcsq11,mcsq12,mcsq13



(a) (b) (c)

Figure 3: $N[m_t]\delta m$ with peak source frequency, (a) 5 Hz, (b) 10 Hz, (c) 15 Hz. lewi/Fig/marm mcsq5Hz,mcsq10Hz,mcsq15Hz

each receiver and 8 meters below the surface. We use Ricker1 wavelet with 10Hz peak frequency as the source. Figure 4 shows the data from 5.8 km shot and 6 km shot. Data is recorded along a line segment in this case and the end of the line accounts for high frequency change and will results in edge effects in images. This model contains a shallow water layer on the top and thus the computed Born data contains direct waves (waves travel directly from source to receivers without any reflection) and diving waves (refracted waves), which will be difficult to fit during inversion. To mitigate edge effects, the data is tapered and to mitigate the influence of direct and diving waves, the data is muted. Both *mute* and *taper* are smooth varying cutoff functions that are applied to each shot of the data. NOTE: the same tapering and muting are applied both to the input data and during inversion process. If denote by M the mute and taper operator, that is equivalent to solve the following optimization problem

$$\tilde{J}[m_l] = \min_{\delta\bar{m}} J[m_l, \delta\bar{m}] = \frac{1}{2}\|MD\bar{F}[m_l]\delta\bar{m} - M\delta d\|^2 + \frac{\alpha^2}{2}\|A\delta\bar{m}\|^2,$$

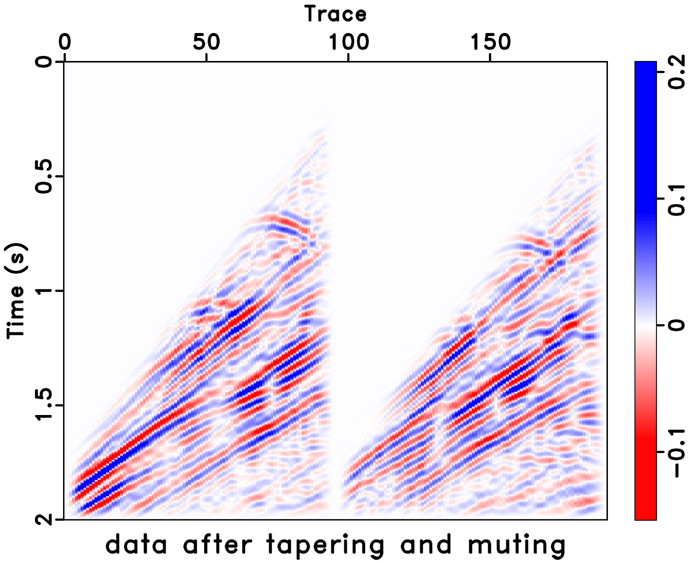
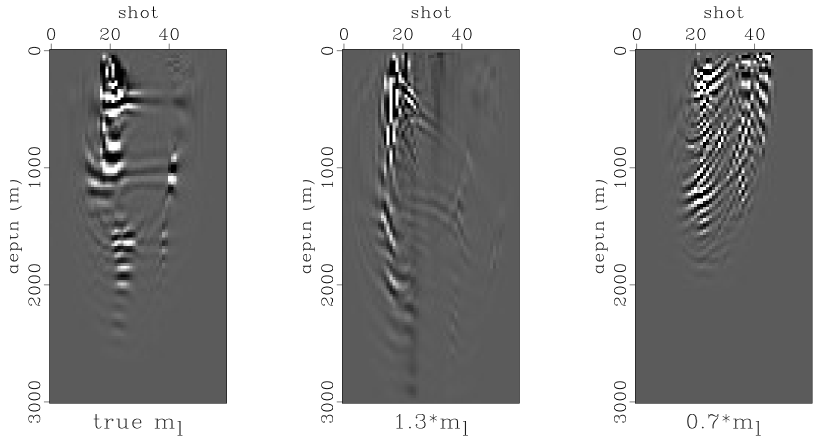


Figure 4: Data is computed using the true velocity and reflectivity as in figure 1(a) and 1(b) with $\mathbf{x}_s = 5.8$ km, 6 km. Data shown in this figure has been applied tapering and muting. lewi/Fig/marm born2shot-tnm

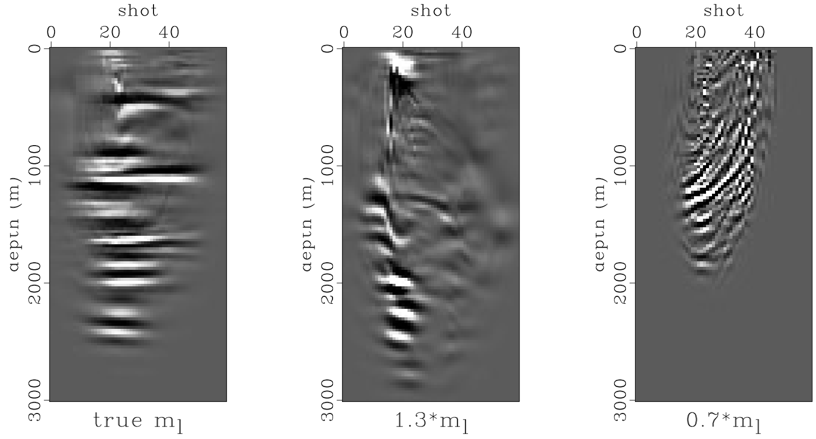
The extended approach to inversion draws inspiration from migration velocity analysis. In principle, migrated shot record image volumes $D\bar{F}[m]^T\delta d$ should be "flat" along the shot axis, i.e. independent of $\mathbf{x}_s$ for correct velocity. In practice, amplitude anomalies may obscure this effect, as is illustrated in figure 5(a).

Instead, we follow the mathematical path laid out above, and base our construction of an objective function on the linearized inversion volume ($\delta\bar{m}$, solution of equation (7)), instead of the migrated image volume. We use conjugate gradient iteration (Nocedal



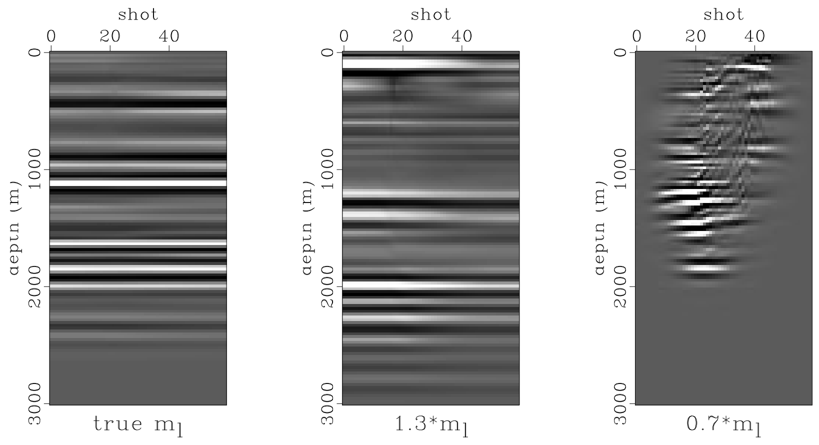
(a) (b) (c)

Figure 5: RTM image gathers for (a) correct velocity, (b) 1.3 times of correct velocity, (c) 0.7 times of correct velocity at $x = 5088$m. lewi/Fig/marm igt-extrtm,igt-extrtm-lg,igt-extrtm-sl



(a) (b) (c)

Figure 6: Inversion image gathers for (a) correct velocity, (b) 1.3 times of correct velocity, (c) 0.7 times of correct velocity for $\alpha = 0.01$. lewi/Fig/marm igt-tgtvel-DS001,igt-lgvel-DS001,igt-slvel-DS001



(a) (b) (c)

Figure 7: Inversion image gathers for (a) correct velocity, (b) 1.3 times of correct velocity, (c) 0.7 times of correct velocity for $\alpha = 0.1$. lewi/Fig/marm igt-tgtvel-DS01,igt-lgvel-DS01,igt-slvel-DS01

and Wright, 1999) to approximately minimize $J[m_l, \delta\bar{m}]$ over $\delta\bar{m}$, solve the normal equation (7) and thus compute $\tilde{J}[m_l]$ (equation 6).

Figure 6 shows image $(\mathbf{z}, \mathbf{x}_s)$ gathers for the same horizontal position as in figure 5. We use weight $\alpha = 0.01$, and perform 100 steps of conjugate gradient iterations. The gradient (normal residual) is reduced 5% of its original value for the true low frequency velocity, and 9% of its original value for other velocities. We can see clearly the flatness of the inverted gathers for correct velocity, and the systematic tendency to slope one way or the other when the velocity is incorrect.

Increasing α will force the inverted velocity to be more $\mathbf{x}_s$-independent, and the objective to behave more like the ordinary least-squares objective. Figure 7 shows the same image gathers as figure 6, but this time with $\alpha = 0.1$. Now the requirement of $\mathbf{x}_s$-independent has largely overwhelmed the kinematic information in the gathers.

Figure 8 displays the values of the approximate $\tilde{J}[m]$ along the line segment

$$m = \sigma m_t,$$

with 11 evenly spaced points of $\sigma \in [0.6, 1.4]$, for several values of α (0.01, 0.1, and 1.0), and less (20 iterations) and more (100 iterations) application of the conjugate gradient algorithm. Here m_t is the background velocity displayed in Figure 1(a). Small α tends to give flat valley near the global minimum, while with large α, the valley is deep and narrow, and stationary points other than global minima appear.

Figure 9 shows a similar sampling of $\tilde{J}[m]$ values for the line segment

$$m = (1-\sigma)m_t + \sigma m_0, m_0(\mathbf{x}) = 1500\text{m/s},$$

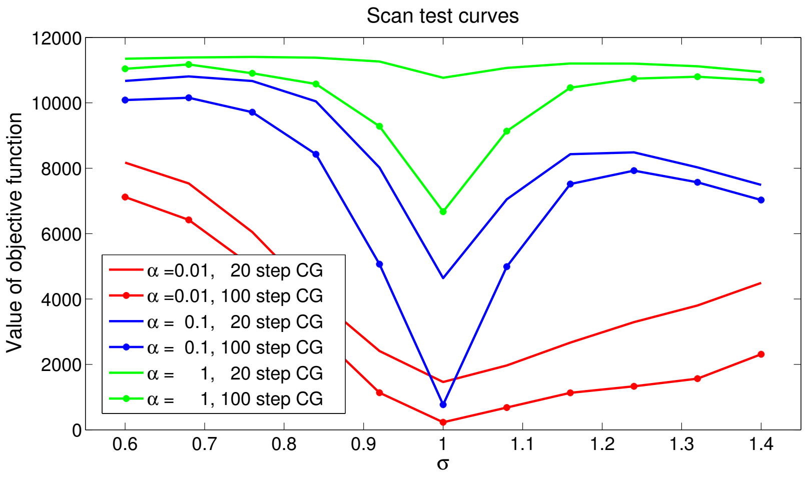


Figure 8: Values of $\tilde{J}[m]$ for $m = \sigma m_t$ with $\sigma \in [0.6, 1.4]$: several values of α, and 20 or 100 conjugate gradient iterations. $\boxed{\text{lewi/Fig/marm scantapermute}}$

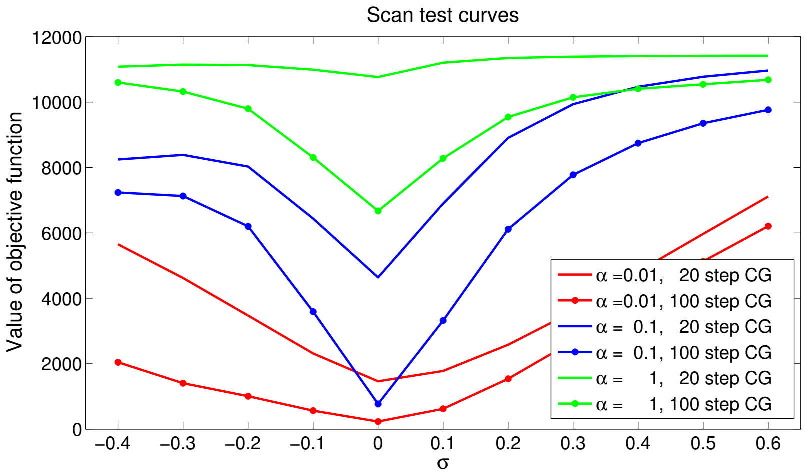


Figure 9: Values of $\tilde{J}[m]$ for $m = (1-\sigma)m_t + \sigma m_0$ with $\sigma \in [-0.4, 0.6]$ and $m_0 = 1500$ m/ms: several values of α, and 20 or 100 conjugate gradient iterations. $\boxed{\text{lewi/Fig/marm scanhomtapermute}}$

and 11 evenly spaced choices of $\sigma \in [-0.4, 0.6]$.

Tomographic Operator Test

In this section, we test our implementation of the tomographic operator. I would like to thank Exxonmobil Upstream Research Company for permissions to use these results in this proposal. These results in this subsection and the next two subsections are obtained when I did my internship there during summer 2014. The Gaussian anomaly model I used for tests is supplied by Yaxun Tang.

Results shown in this section are obtained with model extension, but only the stacked version of them is shown. Tomographic operator

$$D^2F[m_0]^T[\cdot,\cdot]$$

is important since it is the operator we use for gradient computation. Thus the successful implementation of it is really important for the convergence of the inversion.

In order to perform the test, we need three models: background velocity model $m_0 =$ 2km/s, reflectivity model δm (Figure 10(a)), velocity perturbation dm (Figure 10(b)).

Then we compute the Born data perturbation (Figure 10(e))

$$\delta^2 d = D^2F[m_0][\delta m, dm] \approx DF[m_0 + dm]\delta m - DF[m_0]\delta m$$

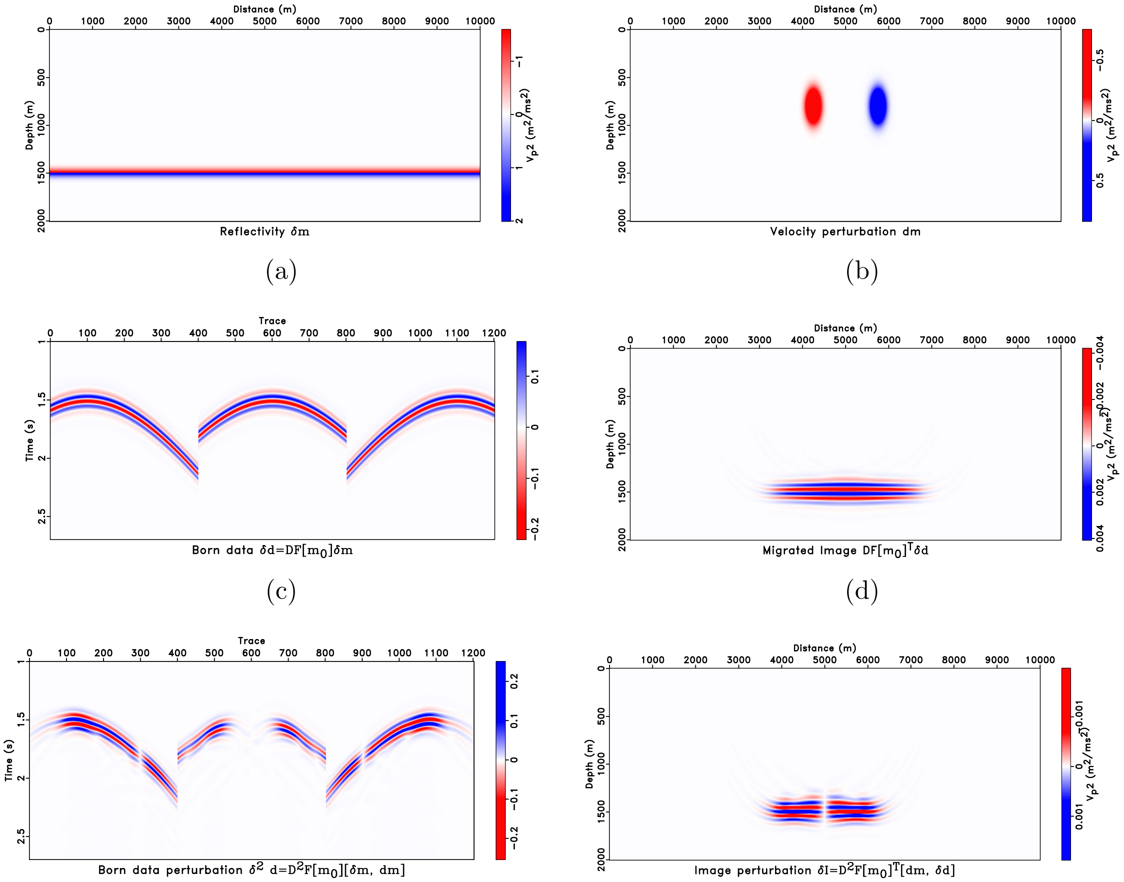
and image perturbation (Figure 10(f))

$$\delta I = D^2F[m_0]^T[dm, \delta d] \approx DF[m_0 + dm]\delta d - DF[m_0]\delta d$$

due to the velocity perturbation dm.

Tomographic operator is then applied to the perturbed Born data and perturbed image (Figure 11(a), 11(b)).

From Figure 12, we see that the tomographic operator could indicate the correct position of these two anomalies with correct reflectivity and data perturbation or image perturbation and data as inputs. Quantitative measurements are also necessary: adjoint relation tests are effective to test a correct implementation of an operator and its adjoint. Table 1 shows adjoint relations of both first order derivative and second order derivative of operator F. For $A = DF[m_0]$, fixed input $x = \delta m$ as shown in Figure **??** and $y = Ax$. For $A = D^2F[m_0][\delta m, \cdot]$, fixed input $x = dm$ as shown in Figure 10(b) and $y = Ax$. Since the four values are all close to or smaller than $100 * macheps$, we could say our tomographic operator is correctly implemented.

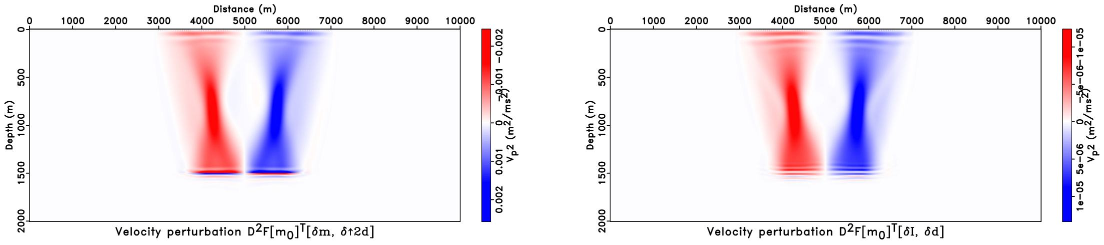


(a) (b)

(c) (d)

(e) (f)

Figure 10: (a) reflectivity δm, (b) Gaussian anomaly velocity perturbation dm, (c) Born data $\delta d = DF[m_0]\delta m$, (d) Image $I = DF[m_0]^T\delta d$, (e) Born data perturbation due to velocity perturbation $\delta^2 d = D^2F[m_0][\delta m, dm]$, (f) Image perturbation due to velocity perturbation $\delta I = D^2F[m_0]^T[dm, \delta d]$. lewi/Fig/gauss-tomo dcsq-flat,dcsqsm,born3shot-flat,mcsq3-b1,d2f3shot-flat,dimage



(a) (b)

Figure 11: (a) tomographic operator output with migrated image and Born data perturbation $D^2F[m_0]^T[\delta m, \delta^2 d]$, (b) tomographic operator output with image perturbation and Born data $D^2F[m_0]^T[\delta I, \delta d]$. lewi/Fig/gauss-tomo mcsq3-b2,tcsq

		random x, y	fixed input x, y
$A = DF[m_0]$	$\langle Ax, y\rangle$	1.320123e+11	4.218034e+16
	$\langle x, A^T y\rangle$	1.320405e+11	4.218104e+16
	$\dfrac{\langle Ax, y\rangle - \langle x, A^T y\rangle}{\|Ax\|\|y\|}$	2.384397e-09	1.659730e-05
$A = D^2F[m_0][\delta m, \cdot]$	$\langle Ax, y\rangle$	1.522127e+11	9.053059e+15
	$\langle x, A^T y\rangle$	1.522677e+11	9.053160e+15
	$\dfrac{\langle Ax, y\rangle - \langle x, A^T y\rangle}{\|Ax\|\|y\|}$	5.247307e-09	1.114891e-05

Table 1: Adjoint relation results and 100**macheps*=1.19209290e-05 in our system.

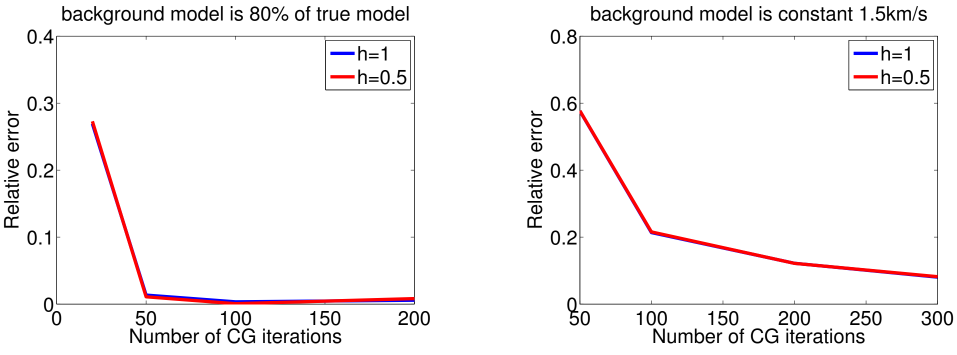
Gradient Accuracy Test

In this subsection, we test how accurate is the approximate gradient in equation (11). We compare the following two values for different values of m

$$a1 = \langle \nabla \tilde{J}[m], dm\rangle_s$$

$$a2 = \frac{\tilde{J}[m + h * dm] - \tilde{J}[m - h * dm]}{2h}.$$

Relative error is defined as $\dfrac{|a2 - a1|}{a1}$.



(a)

(b)

Figure 12: (a) gradient accuracy curve for velocity that is near the true velocity model, (b) gradient accuracy curve for velocity that is far away from the true velocity model. lewi/Fig/marm GradAccuracy80,GradAccuracy15

In the tests, the true background model is the smoothed Marmousi model. Tapering, muting are applied to the data. Windowing operator is applied to the computed reflectivity. Figure 12(a) shows the relative error at m which is 80% of the true velocity model. Figure 12(b) shows the relative error at constant velocity $m = 1.5$km/s which is far away from the true velocity model.

We see that when velocity is close to true velocity, 50 steps of conjugate gradient iteration could give good enough approximate gradient. For velocity that is very far away from the true velocity, 300 steps of iteration is not enough to give a good approximate gradient for this problem. These indicate two things: first, without any knowledge of the true velocity mode, we may expect slow convergence rate at the beginning of the inversion; second, a method that could deal with non-accurate gradient is preferred.

Inversion of Gaussian Anomaly Model

In this subsection, I will show preliminary inversion results for 15 percent Gaussian anomaly velocity model.

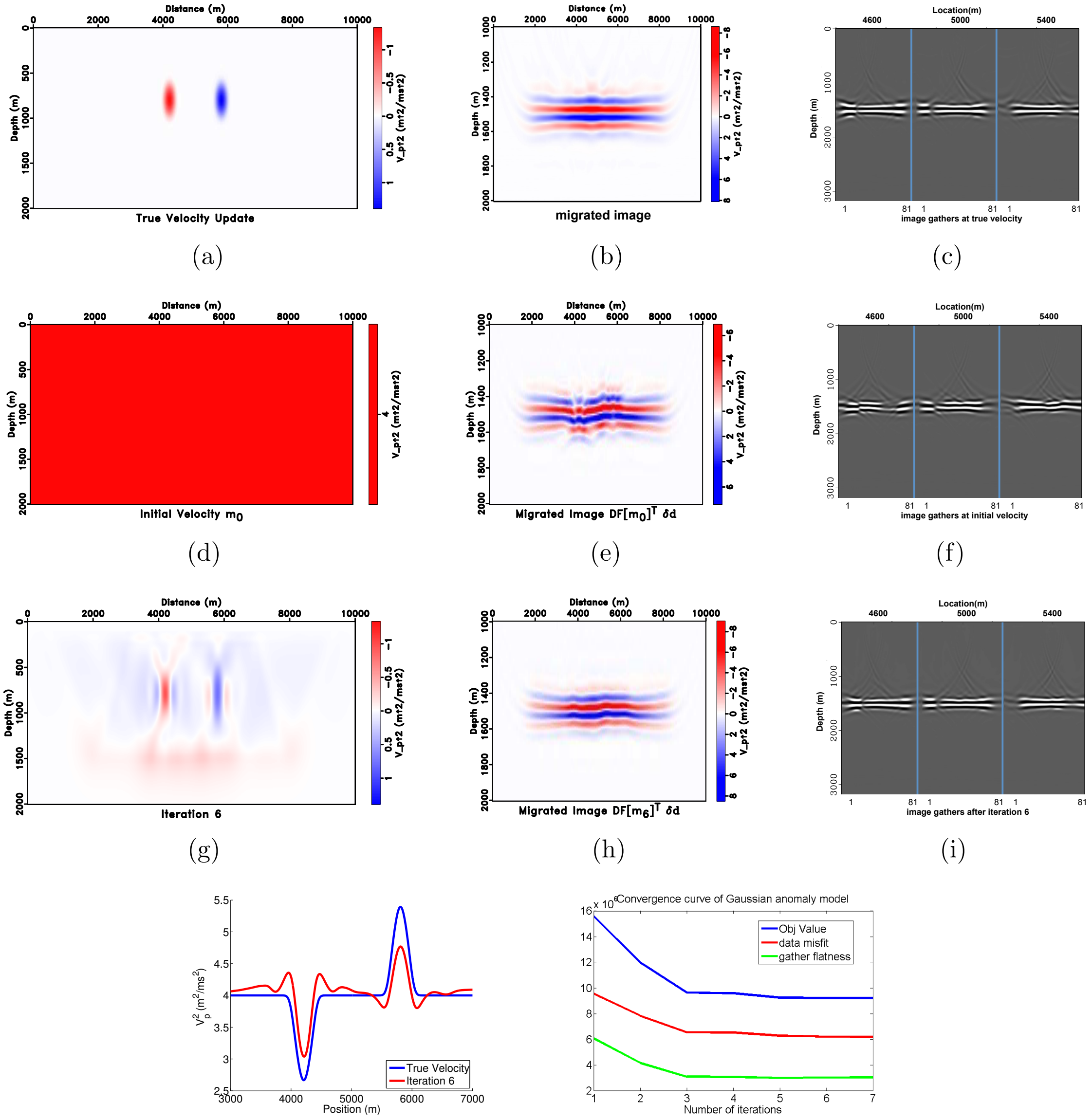
We use $(2, 4)$ finite difference scheme: 2-nd order accurate in time and 4-th order accurate in space. Grid size is 10 m in both z and x directions. We use Ricker wavelet with central frequency 8 Hz as a source wavelet.

The true velocity model is 2 km/s plus the velocity update in Figure 10(b). 81 sources with 8 Hz peak frequency are placed between 3km to 7km with 50m spacing. 999 receivers are placed on the top starting from 10m to 9990m and fixed for each shot.

Inversion starts with constant initial velocity $m_0 = 2$km/s. For each iteration of inversion, we run 10 steps conjugate gradient iterations to get the reflectivity and then run backtracking line search with first step length=1 and back tracking factor=0.5. Limited memory BFGS method with last 5 iterations information is used. If we look at Figure **??**, although there are some unwanted non-zero values around and below two anomalies, positions of them are correctly detected and the values of them tends to be correct. If we could afford run more steps of conjugate gradient iterations, these values will become less noticeable. Figure 13(k) plots the convergence curve of the objective function. The blue curve is the value of the whole objective function. The red curve is the value of the data misfit term and the green curve is the value of the second term which measures how flat are these gathers. This indicates that the two terms are both decreasing through iterations.

CONCLUSION

A natural objective function for the linearized extended waveform inversion combines least squares data misfit and a differential semblance penalty for non-physical dependence on the model extension coordinates. We have shown numerically that



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k)

Figure 13: (a) real velocity update, (b) real image, (c) shot gathers at real velocity, (d) initial velocity model 2 km/s, (e) image at initial velocity model, (f) shot gathers at initial velocity, (g) velocity update after 6 iterations, (h) migrated image after 6 iterations, (i) shot gathers at velocity after iteration 6, (j) velocity comparison at depth 800m, (k) Convergence curves of two terms in the objective function.

lewi/Fig/gauss-inv dcsqsm2,mcsq-flat2,cigvtrue,csqinit,mcsqinit,cigv0,dsestcsq10-6,mcsq10-6,cig6,Vel6 the normal operator is a pseudo-differential operators with the differential semblance operator as an annihilator and it is smooth as a function of background velocity. This property is crucial for the success of this linearized extended waveform inversion method.

We examined this objective for constant density acoustic modeling of reflected waves. As suggested by Kern and Symes (1994), the reduced objective (with short scale components eliminated via a quadratic optimization) tends to be smooth and unimodal in the background (velocity) model parameters, with proper choice of penalty weight and sufficiently precise solution of the inner minimization. Stolk and Symes (2003) showed that the reduced objective has these properties only for the differential measure of semblance, up to inessential modifications.

Numerical results from Gaussian anomaly model show that this method is capable of the reconstruction of Gaussian anomalies.

PROPOSED PROJECT

As shown before, the choice of penalty weight and the correctness of the solution to the extended least squares migration are very important to the smoothness and unimodality of our reduced objective function. This observation suggests several possible topics for further work.

Inversion Velocity Analysis

The inversion velocity analysis could avoid choosing the penalty weight. IVA reformulate the problem (4) as the following optimization problem:

$$\begin{array}{ll} \min\limits_{\delta\bar{m}} & \frac{1}{2}\|A\delta\bar{m}[m_l]\|^2 \\ \text{subject to} & \delta\bar{m}[m_l] = (D\bar{F}[m_l]^T D\bar{F}[m_l] + \epsilon^2 I)^{-1} D\bar{F}[m_l]^T \delta d \end{array}$$

Although we still have to choose ϵ in this formulation, its values is less critical than the penalty weight.

The $\delta\bar{m}[m_l]$ is solved again by some Krylov subspace method, say conjugate gradient method. Thus, numerically $\delta\bar{m}[m_l]$ is only solved approximately, which results in that the gradient is only an approximation. There are several ways to approximate the gradient of the objective function. Symes derived the following form of approximate gradient together with an error estimate.

$$\nabla\tilde{J}[m_l] = D^2\bar{F}[m_l]^T[\delta\bar{m}_a[m_l], D\bar{F}[m_l]\bar{q}_a[m_l]] + O(\epsilon^2\|\delta d\|^2)$$

with $\delta\bar{m}_a[m_l]$ the solution of the constraint, and $\bar{q}_a[m_l]$ the solution of the same constraint with $D\bar{F}[m_l]^T\delta d$ in the right hand side replaced by $A^T A\delta\bar{m}_a[m_l]$.

It is known that we can control $O(1)$ error in solutions $\delta\bar{m}_a, \bar{q}_a$ by controlling normal residual. The gradient equation implies that we can make gradient relative error $O(\epsilon^2)$.

Based on the approximate gradient formulation and what we see in gradient accuracy tests, we need to find methods that could converge in presence of gradient error. The optimization algorithm that couples convergence to gradient relative error (Heinkenschloss and Vicente, 2001) will be implemented and tested.

Preconditioners

The feature of the the linear problem indicates that an iterative method should be employed. And based on my experience on numerical examples, the main cost is solving the "extended least squares migration", that is solving equation

$$\left(D\bar{F}[m_l]^T D\bar{F}[m_l] + \alpha^2 A^T A + \epsilon^2 I\right)\delta\bar{m} = D\bar{F}[m_l]^T \delta d, \tag{15}$$

for $\delta\bar{m}$ and with $\alpha = 0$ for IVA. Preconditioned Krylov space methods seem natural for this problem, and many preconditioners have been suggested in the recent literature. It remains to evaluate them in the context of extended waveform inversion.

In the equation (7) for $\delta\bar{m}$, we use $N[m_l]^\dagger$, since $N[m_l]$ is not invertible in some cases. The principal symbol of pseudo-differential operator $D\bar{F}[m_l]^T D\bar{F}[m_l]$ is of order 1 (Rakesh, 1988; Symes, 1998). And for our choice of $A = \dfrac{\partial}{\partial \mathbf{x}_s}$ for shot coordinate extension, $A^T A$ is a differential operator of order 2. These lead to very large condition number when we discretize the normal operator and result in an ill-conditioned problem when solve for $\delta\bar{m}$ by "extended least squares migration" method.

One option of preconditioning is to use

$$\Lambda = (I - L)^{\frac{1}{2}}, \text{ or just } \Lambda = L^{\frac{1}{2}}$$

which are introduced in the last section and have principal symbol

$$(1 + \sum_{i=1}^{n} \omega_i \xi_i^2)^{\frac{1}{2}} \text{ and } (\sum_{i=1}^{n} \omega_i \xi_i^2)^{\frac{1}{2}}$$

respectively, n is the dimension of the space (Taylor, 1981). Thus Λ is a pseudo-differential operator of order 1 and Λ^{-1} is a pseudo-differential operator of order -1.

Preconditioned "extended least squares migration" becomes

$$(\Lambda^{-\frac{T}{2}} D\bar{F}[m_l]^T D\bar{F}[m_l]\Lambda^{-\frac{1}{2}} + \alpha^2 \Lambda^{-T} A^T A \Lambda^{-1} + \epsilon^2 I)\delta\bar{m} = \Lambda^{-\frac{T}{2}} D\bar{F}[m_l]^T \delta d. \tag{16}$$

By the composition theorem of pseudo-differential operators, the first term inside the parentheses is of order 0 and the second term inside the parentheses is also of order 0. This results in a much better conditioned problem.

Define block operator

$$B[m_l] = \begin{pmatrix} D\bar{F}[m_l]\Lambda^{-\frac{1}{2}} \\ \alpha A\Lambda^{-1} \\ \epsilon I \end{pmatrix},$$

and $\mathbf{d} = \left(\delta d^T, 0, 0\right)^T$. In operator $B[m_l]$, $D\bar{F}[m_l]\Lambda^{-1}$: $L^2 \mapsto L^2$ is bounded and $A\Lambda^{-1}$ as an oder 0 operator is also bounded, which implies that operator $B[m_l]$ is bounded. For any function $v \in L^2$, $\|B[m_l]v\|^2 \geq \epsilon\|v\|^2$, i.e. $B[m_l]$ is coercive. Thus B^TB is also bounded and coercive. By Lax-Milgram theorem, there exists a unique solution to equation (16). This explains when we define the reduced objective function

$$\tilde{J}[m_l] = \min_{\delta\bar{m}} \frac{1}{2}\|B[m_l]\delta\bar{m} - \mathbf{d}\|^2,$$

minimum is used instead of infinum.

Another option is the approximate linearized optimal scaling (Symes, 2008a).

With non-multipathing assumption on the model and flat spectrum assumption on the source, the normal operator without model extension is decomposed as a power of Laplace operator compositing with a scaling operator

$$DF[m_l]^T DF[m_l]\delta m \simeq L^{\frac{n-1}{2}} V^2 \delta m$$

with n the space dimension and V^2 the multiplication by $\sigma_0(x, \nabla\psi(x))$. The scaling factor σ_0 is a function of position and dip, with $\nabla\psi(x)$ indicating the dip direction of reflectivity model δm. Then the preconditioner we want to use in the "extended least squares migration" is defined as

$$P = L^{-\frac{n-1}{4}} W,$$

with $W = V^\dagger$ the multiplication by $(\sigma_0)^\dagger$, which is obtained by solving an optimization problem.

Pseudo-differential operators commute to leading order. Thus

$$P^2 \simeq \left(DF[m_l]^T DF[m_l]\right)^\dagger.$$

Since $\sigma_0 \geq 0$, it is represented as $\sigma_0^\dagger = e^\tau$. Bicubic spline is used to represent τ:

$$\sigma_0^\dagger = \exp\left(\sum_{i=1}^{N} \tau_i \psi_i(x)\right).$$

Let's review the algorithm stated in (Symes, 2008a):

- Compute $\delta m_{\text{mig}} = DF[m_l]^T d$;
- Compute $\delta m_{\text{filt}} = L^{-\frac{n-1}{2}} DF[m_l]^T DF[m_l] \delta m_{\text{mig}}$;
- Minimize
$$J[\tau] = \sum_{x \in K} \left[\left(\exp \left(\sum_{i=1}^{N} \tau_i \psi_i(x) \right) \right) \delta m_{\text{filt}} - \delta m_{\text{mig}} \right]^2 + \mu \sum_{i=1}^{N} \tau_i^2 .$$
  Here K is a window and images have no reasonable value everywhere outside the window.

We run the above algorithm separately for each shot data, and result in different operator P for different shot data. We denote the shot coordinate dependent operator as $\bar{P}$. And
$$\bar{P}(x_s) = L^{-\frac{n-1}{4}} W_{x_s}.$$

Thus the preconditioned extended least squares migration problem has the following form
$$\left(\bar{P} D\bar{F}[m_l]^T D\bar{F}[m_l] \bar{P} + \alpha^2 \bar{P}^2 A^T A \bar{P}^2 + \epsilon^2 I \right) y = \bar{P} D\bar{F}[m_l]^T \delta d,$$
and the resulting
$$\delta \bar{m} = \bar{P} y.$$
NOTE that this preconditioner depends on background velocity model. The computation of $\bar{P}$ is needed each time after updating the background velocity. We may also compute x_s-independent P by using all data in the computation of P and then define $\bar{P}(x_s) = P$. Performance difference of using these two different $\bar{P}$ remains to be evaluated. This optimal scaling preconditioner may be used together with the square root of Helmholtz operator to control the condition number of the least squares problem.

Finally, correct gradient computation is the key for a successful inversion, as we discussed before. Many technics should be included in the calculation of gradient, for example using a different weighted Sobolev norm in the objective function.

Then these methods will be applied to complex synthetic examples and also simple field data to explore the true capacity of them. There are lots of examples of the comparison between migration velocity analysis and FWI already in the literature. Examples that illustrate that FWI and linearized FWI (without model extension) suffer from cycle skipping, while the method in this proposal could seccesfully reconstruct the model, will also be generated.

ACKNOWLEDGMENTS

I would like to think Dr Anatoly Baumstein and Dr Yaxun Tang for a lot of help and inspiring discussions when I did my internship at Exxonmobil and thank them for allowing me to include the work I did there in this proposal.

We wish to thank the sponsors of The Rice Inversion Project for their continued support.

The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper (`http://www.tacc.utexas.edu`).

REFERENCES

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.

Beylkin, G., and R. Burridge, 1990, Linearized inverse scattering problem of acoustics and elasticity: Wave Motion, **12**, 15–22.

Biondi, B., and A. Almomin, 2012, Tomographic full waveform inversion (TFWI) by combining full waveform inversion with wave-equation migration velocity analysis: 82nd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, SI9.5.

———, 2014, Simultaneous inversion of full data bandwidth by tomographic full-waveform inversion: Geophysics, **79**, WA129–WA140.

Biondi, B., and P. Sava, 2004, Wave-equation migration velocity analysis - I: Theory, and II: Subsalt imaging examples: Geophysics, **52**, 593–623.

Biondi, B., and Y. Zhang, 2013, Moveout-based wave-equation migration velocity analysis: Geophysics, **78**, U31–U39.

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

Bourgeois, A., P. Lailly, and R. Versteeg, 1991, The Marmousi model, *in* The Marmousi Experience: IFP/Technip.

Brandsberg-Dahl, S., M. V. De Hoop, and B. Ursin, 2003, Focusing in dip and ava compensation on scattering-angle/azimuth common image gathers: Geophysics, **68**, 232–254.

Burridge, R., M. V. De Hoop, D. Miller, and C. Spencer, 1998, Multiparameter inversion in anisotropic elastic media: Geophysical Journal International, **134**, 757–777.

Chauris, H., and M. Noble, 2001, Two-dimensional velocity macro model estimation from seismic reflection data by local differential semblance optimization: applications synthetic and real data sets: Geophysical Journal International, **144**, 14–26.

Chauris, H., and R.-E. Plessix, 2013, Differential waveform inversion - a way to cope with multiples?: Presented at the 75th Conference EAGE, European Association for Geoscientists and Engineers. (Expanded Abstract).

Ehinger, A., and P. Lailly, 1993, Prestack imaging by coupled linearized inversion.

Gauthier, O., A. Tarantola, and J. Virieux, 1986, Two-dimensional nonlinear inversion of seismic waveforms: Geophysics, **51**, 1387–1403.

Hascoët, L., and V. Pascual, 2004, TAPENADE 2.1 user's guide: Technical Report 0300, INRIA.

Heinkenschloss, M., and L. Vicente, 2001, Analysis of inexact trust-region SQP algorithms: SIAM J. Optimization, **12(2)**, 283–302.

Jannane, M., W. Beydoun, E. Crase, D. Cao, Z. Koren, E. Landa, M. Mendes, A. Pica, M. Noble, G. Roeth, S. Singh, R. Snieder, A. Tarantola, D. Trezeguet, and M. Xie, 1989, Wavelengths of earth structures that can be resolved from seismic reflection data: Geophysics, **54**, 906–910.

Kern, M., and W. Symes, 1994, Inversion of reflection seismograms by differential semblance analysis: Algorithm structure and synthetic examples: Geophysical Prospecting, **99**, 565–614.

Liu, Y., W. W. Symes, Y. Huang, and Z. Li, 2013, Linearized extended waveform inversion via differential semblance optimization in the depth oriented extension: 83rd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 4869–4874.

Mulder, W., and F. ten Kroode, 2002, Automatic velocity analysis by differential semblance optimization: Geophysics, **67**, 1184–1191.

Nocedal, J., and S. Wright, 1999, Numerical Optimization: Springer Verlag.

Nolan, C., and W. Symes, 1997, Global solution of a linearized inverse problem for the wave equation: Communications on Partial Differential Equations, **22**, 919–952.

Prucha, M., B. Biondi, and W. Symes, 1999, Angle-domain common image gathers by wave-equation migration: 69th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 824–827.

Rakesh, 1988, A linearized inverse problem for the wave equation: Communications on Partial Differential Equations, **13**, 573–601.

Santosa, F., and W. Symes, 1989, An Analysis of Least-Squares Velocity Inversion: Society of Exploration Geophysicists, volume **4** *of* Geophysical Monographs.

Shan, G., and Y. Wang, 2013, RTM based wave equation migration velocity analysis: 83rd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 4726–4731.

Shen, P., 2012, An RTM based automatic migration velocity analysis in image domain: 82nd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, SVE2.2.

Shen, P., and W. Symes, 2008, Automatic velocity analysis via shot profile migration: Geophysics, **73**, VE49–60.

Stolk, C. C., 2000, Microlocal analysis of a seismic linearized inverse problem: Wave Motion, **32**, 267–290.

Stolk, C. C., and W. Symes, 2003, Smooth objective functionals for seismic velocity inversion: Inverse Problems, **19**, 73–89.

Sun, D., 2012, A nonlinear Differential Semblance algorithm for waveform inversion: PhD thesis, Rice University, Houston, Texas, USA.

Symes, W., 1986, Stability and instability results for inverse problems in several-dimensional wave propagation, *in* Proc. 7th International Conference on Computing Methods in Applied Science and Engineering: North-Holland.

———, 1998, Mathematics of reflection seismology. (available at http://www.trip.caam.rice.edu).

———, 2008a, Approximate linearized inversion by optimal scaling of prestack depth migration: Geophysics, **73**, R23–R35.

———, 2008b, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790.

Symes, W., and J. J. Carazzone, 1991, Velocity inversion by differential semblance optimization: Geophysics, **56**, 654–663.

Tang, Y., and B. Biondi, 2011, Target-oriented wavefield tomography using synthesized born data: Geophysics, **76**, WB191–WB207.

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

Taylor, M., 1981, Pseudodifferential Operators: Princeton University Press.

ten Kroode, F., D. Smit, and A. Verdel, 1998, A microlocal analysis of migration: Wave Motion, **28**, 149–172.

van Leeuwen, T., and W. A. Mulder, 2009, A variable projection method for waveform inversion: Presented at the 71st Conference EAGE, European Association for Geoscientists and Engineers. (Expanded Abstract).

Virieux, J., and S. Operto, 2009, An overview of full waveform inversion in exploration geophysics: Geophysics, **74**, WCC127–WCC152.

Weibull, W., and B. Arntsen, 2013, Automatic velocity analysis with reverse time migration: Geophysics, **78**, S179–S192.

———, 2014, Anisotropic migration velocity analysis using reverse time migration: Geophysics, **79**, R13–R25.

IWAVE Structure and Basic Use Cases

William. W. Symes

ABSTRACT

The IWAVE control structure facilitates construction of wave simulators with flexible specification of input and output. This document describes synthesis of seismograms and wavefield movies from initial data and from single and multiple sources (right-hand sides), and linearized ("Born") and linearized adjoint (reverse time migration) modeling. The choice of physical model and simulation method - constant density acoustics with Dirichlet boundary conditions and $(2, 2k)$ finite difference schemes - is the simplest possible, but the framework accommodates any regularly gridded stencil-based discretization of arbitrary wave physics in the same way.

INTRODUCTION

IWAVE combines minimal physics- and scheme-specific code with a common base of memory allocation, SPMD communication and other forms of parallelism, i/o, and job control code to produce complete wave modeling applications. This paper explains what code must be written to implement a wave modeling application in IWAVE, and how IWAVE makes various modeling options available via a set of examples. It ends with a discussion of various other extensions and capabilities, implemented or planned, including various forms of inversion.

The next section describes the general organization of the IWAVE core code and the additional code that must be added to form an application. Following this generic structural description, I explain how to implement it in the case of the simplest useful example, constant density acoustics.

Some design aspects of IWAVE arise from the intended use cases not discussed here: parallel processing of shots, subdomains, and loops, and interface with an object oriented optimization package, the Rice Vector Library, to facilitate inversion applications. These use cases and their design implications will form the subjects of other reports.

STRUCTURES AND INTERFACES

Three components and their relations define a time-stepping simulator based on regular gridding and stencil field updates:

- a collection of (discrete) dynamic and static (coefficient) fields;
- model-specific functions implementing time step stencils, determining space and time grids and stencil shape, and sanity-checking coefficient fields;
- a list of i/o procedures to populate the fields at initial and intermediate times, and to extract results at final and intermediate times.

Definitions and interfaces for these three components form the core of IWAVE. The critical functions and data structures are static members of the `IWaveInfo` class; definining them completely defines the application. Declarations appear in `iwave/core/include/iwaveinfo.hh`, which should be consulted for function signatures etc. Applications must supply definitions, each unique (as is requlred for static global data), as described in the final paragraphs of this section.

Fields

The top level data structure characterizing an application implemented in IWAVE is a list of field keywords and other attributes. Each field gets a keyword, acting as a mnemonic index for internal and external reference. Fields are dynamic or static, and may be primal or dual in each coordinate axis: "dual" here refers to the grid, so dual fields represent values assigned to edges, faces, or volumes, rather than grid vertices (nodes). IWAVE encodes these boolean attributes as 0's or 1's. Finally, a bit of information about the scheme intrudes: each dynamic field is updated in one of the (sub)steps of a (possibly) multistep method, and for that substep, and that substep only, needs ghost cell data exchanged.

The `FIELD` struct containing this information is organized in the order

- (string) keyword;
- (int) dynamic flag;
- (int) substep number;
- (int array) duality flag for each axis.

For instance, a (z, x) stress component field for a staggered grid elastic finite difference method is a dynamic fielld, representing a grid sampled centered in the (z, x) faces of the grid cells, and updated in second substep of each time step Moczo et al. (2006). The `FIELD` struct

```
{"szx", 1, 1, {1, 1, 0}}
```

captures this information (here coordinate order is (z, x, y)).. Note that the only arbitrary choice here is the keyword string. The choice of keyword for each field must be consistent throughout the code.

The distinction between static and dynamic fields is more than a convenience. Static fields, representing the coefficients of the model differential equations, must exist in the simulation environment prior to simulation, in some form of persistent store. Simulation input and output fields, defined below, must also exist as persistent store prior to simulation. The current implementation of IWAVE presumes that “persistent store” is a synonym for “disk file”, however the logic is simply that the data for these fields should exist outside of the simulation scope. Future versions of IWAVE may accommodate distributed data as persistent store, for instance. Dynamic fields exist only within the scope of the simulation: IWAVE creates and destroys them in the course of a run.

By convention, the first field listed in the application `FIELD` array is the source of the primary simulation grid, to which all other grids are referenced. Since this information will need to enter the compuation via i/o, this first field should be static (i.e. a coefficient), which will exist in the simulation environment prior to execution of the application.

The static array `iwave_fields` data member of the `IWaveInfo` class lists the static and dynamic fields of an IWAVE application and their top-level attributes using the `FIELD` struct explained above. Being static, `iwave_fields` must be initialized once, and only once, somewhere in global namespace. That is, the application author must supply a definition

```
iwave_fields = { ... };
```

somewhere - the preferred location for this definition is in the model definition header file containing the other required definition to be reviewed below. This model definition file should be included *only* in main program source files, as its static declarations must appear only once in program text.

Field allocation

IWAVE stores all fields as *rarrays* (or RARRs). RARR is an intelligent array type that defines gridpoints by reference to global coordinates in $\mathbf{Z}^d$ for d-dimensional simulations, provides both one- and multi-dimensional access to array data, and includes many utiliities for data manipulation. The time step function implementing the dynamic field updates will be written in terms of RARR utilities and data members. The RARRs corresponding to the application's fields together form an rdomain (RDOM), which is simply an array of RARR, indexed in the order that they appear in the iwave_fields array.

Derivatives (and adjoint derivatives) of the simulation map involve multiple copies of the basic application fields: the first derivative needs two copies, a set of reference fields and a corresponding set of perturbation fields. Since the kth derivative is the derivative of the $k-1$st derivative, the kth derivative requires 2^k sets of fields in its definition. The IWaveTree data structure of order k is a vector of 2^k RDOMs suitable for defining the kth derivative; the first 2^{k-1} RDOMs comprise the reference IWaveTree, the second form the perturbation.

Functions

The implementation of an IWAVE application depends on a half-dozen functions, which the application author must supply. The most obvious of these is the time step, or dynamic field update, function, and it is described here. All six essential functions are documented in

```
RSFSRC/trip/iwave/core/include/iwinfo.hh
```

The time step interface accommodates (in principle) the update functions for all orders of derivative, and adjoint (reverse mode) as well as forward time stepping. Its signature is encapsulated in a typedef:

```
typedef void (*FD_TIMESTEP)(std::vector<RDOM *> dom,
                            bool fwd,
                            int iv,
                            void* fdpars);
```

The first argument is the RDOM array extracted from an IWaveTree. The order of derivative to be computed is the base 2 log of dom.size(). The implementation should take the form of a case list, one major case for each order of derivative implemented. Each case (except the simulation itself, or order 0) should be divided into two subcases, one each for forward and adjoint modes, switched by the second `bool` argument `fwd`. Each subcase is further divided into sub-sub-cases according to the *substep index* `iv`. This third refinement permits implementation of multistep methods in this framework. To take a prominent example, staggered grid methods for elastodynamics, in their basic form, use leapfrog time stepping, a two-step methods: velocity fields are updated from stress fields in the first substep, stress fields from velocity fields in the second. This subdivision of time steps in this fashion allows for the simplest coding and reduces the amount of data in ghost cell exchange for domain decomposition, an advantage for low-latency systems. The final `void *` pointer is the usual dodge for faking private class data members in "object oriented C": the opaque object passed by address should be of a type defined for the given application and encapsulating all information needed to compute the time step, such as Courant numbers, auxiliary damping arrays for absorbing layers, and so on. One of the other five functions initializes this object.

I/O

The `FIELD` array member of the `IWaveInfo` class defines the physical fields of a modeling application. The function data members describe the memory allocation and updating of the arrays representing these fields. It remains to describe initialization and finalization, that is, how external data is read to and written from these arrays.

The basic principle of IWAVE i/o is that *the structure of a data item determines the manner in which it is read/written.* Absent self-describing i/o formats, this principle implies that the relevant part of the code implements a case list. Persistent data is presumed to exist in disk files, or in any case be identified by single strings. Accordingly the case switch is the *filename suffix.* At this writing, two file data structures are realized: SEGY minus reel header, or Seismic Unix (SU) format, suffix `su`, and Regularly Sampled Function format, suffix `rsf`. [Other file or network data structures may be added to IWAVE later.]

Since the details of i/o are implicit in the choice of data structure, it remains only to connect external data units with the data arrays with which they communicate. Since the data arrays are indexed by keyword, and since several external data objects may communicate with the same data array, it follows that a keyword must be introduced to stand as a proxy for each external data object in the program text. At runtime, the keyword appears as an index into an associative array describing program inputs and ouputs; the value associated to each key is a path.

Accordingly, the final data member of the `IWaveInfo` class is the `IOKEY` array `iwave_iokeys`. `IOKEY` is a typedef for a struct consisting of

- an external data item keyword

- a field index
- a flag to indicate input (1) or output (0)
- an activity flag, which is key to defining the RVL Operator interface (see discussion below) - active arrays (1) define components of the operator domain (input) or range (output), whereas non-active arrays (0) represent auxiliary inputs/outputs.

For example, an input to the pressure Cauchy data for pressure-velocity acoustics, regarded as auxiliary data, might be represented by the `IOKEY`

```
{"init_p", 1, 1, 0}
```

(assuming that the pressure array has index 1) whereas the output of the same array data, regarded as defining part of the range of the simulation operator, might correspond to

```
{"movie_p",1, 0, 1}
```

Note that any sampling in space and/or time may be represented in this way, and is determined at the time of interaction with the external data unit. Some details of file structure controlling IWAVE i/o are discussed in later in this paper.

The indirection afforded by the external data unit keyword allows the the same IWAVE-based command to be used for many different combinations of inputs and outputs. Parameter pairs `keyword = filename` passed from a driver serve to connect specific disk files or other data repositories to external data keywords, hence to read/write operations.

The IOKEYS array defines i/o info for the basic simulation. Keywords for perturbation fields, used in derivatives and adjoint derivatives, are generated automatically. For example, if the keyword for density is `rho`, then the keyword for the first perturbation density (input to the first derivative) is `rho_d1`, The adjoint output density is `rho_b1`. The choices of suffix correspond to those generated by the *Tapenade* automatic differentiation package Hascoët and Pascual (2013). These keywords would be used in as parameter keys in pairs passed to driver routines, for example, `rho_d1 = my_density_pert.rsf`.

IWaveInfo

Definition of an IWAVE application amounts to initialization, in global namespace of the static public data members of the `IWaveInfo` class:

```
static FIELD iwave_fields[];
static IOKEY iwave_iokeys[];
static FD_MODELINIT minit;
static FD_MODELDEST mdest;
static FD_TIMESTEP timestep;
static FD_TIMEGRID timegrid;
static FD_STENCIL createstencil;
static FD_CHECK check;
```

These initializations must occur precisely once for each application. The recommended procedure is to assign all members except `iwave_iokeys` in a *model definition* header file, which includes both the declarations of the static data types (via `#include "iwinfo.hh"`) and of the function data members (via `#include` of the main model header file). The `iwave_iokeys` array, on the other hand, should be assigned at the top of each command file, after `#include` of the model definition file.

This organization allows the model application library (object files defining various functions) to be linked without alteration to a variety of drivers (main program files) implementing different input/output choices. I have identified two main use cases:

- standalone command: a standalone driver such as `acd` (next section) must have keys provided for all inputs and outputs contemplated. I/O keys for which key=value pairs don't appear amongst the parameters passed to the application at runtime are simply ignored. So this command can be used for all possibilities covered by the chosen I/O keys without any alteration. See the next section for illustration.
- RVL Operator interface, for use in inversion applications: the constructor builds its domain and range spaces using the supplied I/O keys - this is the reason for the "active" flag. I/O keys flagged as active define components of domain and range (product) spaces, with geomtry metadata taken from the files which the keys point to via the parameter list. Data sources corresponding to these keys must be present in the environment and correspond to parameter values. This use case will be described in more detail in another report.

A negative consequence of IWAVE's reliance on static global definitions is that only one IWAVE modeling application can be active in a given process. This disadvantage is offset by two considerations: (1) in the use cases for which IWAVE was designed, only one modeling application per process is required; (2) the creation of a new IWAVE application is as simple as one can imagine, requiring only the definition of a couple of arrays and half-a-dozen functions with specified signatures, and their assignment to static data of `IWaveInfo`. If in the future interesting use cases arise in which truly distinct modeling packages must be combined in the same process, then simple modificiations of the IWAVE overall structure will answer, with slightly increased burden on the programmer.

CASE: CONSTANT-DENSITY ACOUSTICS

A simple example illustrating the framework described above is the IWAVE implementation of the constant-density acoustic wave equation with Dirichlet (pressure-free) boundary conditions, connecting the acoustic potential field $u(\mathbf{x},t)$ and a right-hand side $f(\mathbf{x},t)$ representing a source of mechanical energy, defined in a spatial domain Ω over a suitable time interval,

$$\frac{\partial^2 u}{\partial t^2} - c^2\nabla^2 u = f,$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}),\ \frac{\partial u}{\partial t}(\mathbf{x},0) = v(\mathbf{x})\ \mathbf{x}\in\Omega,$$

$$u(\mathbf{x},t) = 0,\ \mathbf{x}\in\partial\Omega.$$

The examples to be discussed use the centered difference approximation (Kelly et al., 1976)

$$u^{n+1} = 2u^n - u^{n-1} + \Delta t^2 c^2 L u^n + \Delta t^2 f^n$$

in which L is a regular grid difference approximation to the Laplacian, and u^n represents the array of acoustic potential samples for time $n\Delta t$. The choice of L used below is a sum of centered coordinate second difference operators of order $2k$, $k = 1, 2, 4, ...$ resulting in a scheme of formal order 2 in time and $2k$ in space. Lax-Wendroff extension to higher order time approximation fits this pattern also.

Since each array element in u^{n-1} appears exactly once in a loop through the array, it is possible to store only the two arrays for time indices $n-1$ and n, represented by RARRs up and uc respectively, and store c^2 in the RARR csq. The the three-level scheme above becomes

```
up = 2 * uc - up + dt2 * csq .* L uc + dt2 * f^n
[swap up, uc]
```

With the type of discrete finite difference Laplacian described above, the grids for uc, up, and csq are all primal, and commensurable. Since csq must exist along with all of its metadata (its grid information, basically) in the scope of the simulation, it is natural to read the primal grid geometry from it. Thus an appropriate iwave_fields array for acoustic constant density modeling is

```
FIELD IWaveInfo::iwave_fields[]
= {
  {"csq",    0,    0,  {0, 0, 0}},
  {"uc",     1,    0,  {0, 0, 0}},
  {"up",     1,    0,  {0, 0, 0}},
  {"",       0,    0,  {0, 0, 0}}
};
```

The last line functions the same way as the traling NUL for C strings, that is, to signal the end of the structure.

Inspection of the pseudo-code above reveals that up and csq need to be available at precisely the same gridpoints, whereas uc must store additional gridpoints around the boundary ("halo" or "ghost" points) in order that the Laplacian can be built at the csq gridpoints (the "physical" grid). So the memory allocations for uc and up differ, and the "swap" mentioned in the algorithm exchanges only values of the two fields at physical grid points. The algorithm must be completed with a boundary loop which updates the non-physical gridpoints of uc, for example with odd reflection implementing a Dirichlet condition.

The design described in the preceding paragraphs is realized in the IWAVE acoustic constant density package, RSFSRC/trip/iwave/acd. The standalone executable implementing the various options provided by IWAVE is also called acd. It can be built as part of a Madagascar top-down build, in which case it shows up as RSFROOT/bin/sfacd and can be referenced as acd in Madagascar Flows, or standalone via invocation of scons in either RSFSRC/trip or RSFSRC/trip/iwave. In the latter case, the dependency RSFSRC/trip/rvl must be built first. The standalone build has the virtue of permitting local control of build environment. The RSFSRC/trip/admin directory includes a number of example configuration scripts for build options - to use, copy one of these, or create a similar file, as config.py in the root build directory.

The standalone build results in the executable command

```
  RSFSRC/trip/iwave/acd/main/acd.x.
```

In the remainder of this paper, I will refer to this command as acd.x. The SConstruct file in the project subdirectory of the paper directory is also configured to use this standalone-built command.

The IWAVE acoustic constant density implementation includes code for the acoustic simulator and its derivatives (with respect to velocity-squared) of orders 1 and 2, and their adjoints, built from a simple numerical kernel (or set of kernels) of truncation orders in space 2, 4, and 8, and truncation order 2 in time. My research group has used the Tapenade (Hascoët and Pascual, 2013) automatic differentiation package to produce the code for derivatives and adjoints. For example, the signature of an implementation of the $(2, 2k)$ scheme for 3D is

```
acd_3d_[2k](float *** uc3,
            float *** up3,
            float *** csq3,
            int * s,
            int * e,
            ...,
            int * lbc,
            int * rbc);
```

in which uc3 is the 3-dimensional array view of the array uc, and so on; s and e are the vectors of start and end indices for the loop over gridpoints; and ... stands in for a list of difference formula coefficients, the the number and value of which depend on the order ($2k$). The integer arrays lbc and rbc flag whether the left and right boundaries of the computational domain, delimited by s and e, are external (physical) boundaries or internal boundaries. In the former case, phsical boundary conditions must be applied; these are also part of the code.

Tapenade produces similar code for the first derivative of this stencil (with respect to the uc, up, and csq arguments, with signature

```
void acd_3d_[2k]_d(float *** uc,
                   float *** ucd,
                   float *** up,
                   float *** upd,
                   float *** csq,
                   float *** csqd,
                   int * s,
                   int * e,
                   ...,
                   int *lbc,
                   int *rbc);
```

in which ucd, upd, and csqd are the perturbations of the arrays without the d's.

These kernels can be folded into an obvious case list, switched by the inputs to the timestep interface described above.

Tapenade output is not entirely suitable for immediate use: some minor cleanup is necessary, and any serious optimizations (vectorization, for instance) will need to be applied in a tuning phase. However the code as it comes from the package is correct and reasonably readable, and can serve as a baseline with which to verify tuned versions.

Definition of a command based on the fields and functions described above requires one more piece of information: the connection of fields to external data sources and sinks, intermediated by i/o keywords. Many choices are possible; one reasonable choice for the standalone command option and constant density acoustics (acd.x) is:

```
IOKEY IWaveInfo::iwave_iokeys[]
```

```
= {
  {"csq",    0, true,  true },
  {"data",   1, false, true },
  {"source", 1, true,  false},
  {"movie",  1, false, false},
  {"initc",  1, true,  false},
  {"initp",  2, true,  false},
  {"",       0, false, false}
};
```

Clearly the velocity (or rather velocity-squared) must be made available. Two outputs from `uc` are identified, `"data"` and `"movie"`: while nothing about the specs demands this usage, the first is intended for trace output, the second for time slices, as the keywords choices are intended to suggest. Since the precise mechanism of I/O is inherent in the data unit (file structure, for instance) rather than the directed by the code, in fact these mnemonic suggestions could be ignored, and `"data"` used to store a movie, for example. However it is an intended use case that movies might be generated at a byproduct of trace generation, so two output slots are provided. Similarly, several input keywords suggest a right-hand side input (time dependent force divergence traces) (`"source"`) and Cauchy data (`"initc"`, `"initp"`).

Note that the discrete Cauchy data represent pressure at two successive time levels, whereas the natural Cauchy data for the wave equation would provide presure and its time derivative. An application accepting this natural Cauchy data would need to pre-process it into discrete Cauchy data as indicated above. I have elected to "unbundle" this type of pre-processing, that is, it is not included in the IWAVE code itself. Similarly, the natural SEGY representation of the RHS source traces needs to be pre-processed to code the source positions as receiver coordinates, as reviewed below.

Single Shot Examples

The examples presented here are built in the `project` subdirectory of the paper directory, and organized in standard Madagascar fashion with SConstruct (Fomel, 2009). Details of the build are best appreciated by reading the `SConstruct` script in `./project`. All use the standalone build of the acoustic constant density command `acd.x` (in `RSFSRC/trip/iwave/acd/main`). This command self-docs: all of the parameters described below, and others not used in these examples, are described in the self-doc, which the reader should consult in conjunction with this discussion.

All of these examples use the simple layered velocity (-squared) model depicted in Figure 1. This data derives from a corresponding velocity model, fetched from and squared to create a velocity-squared field.

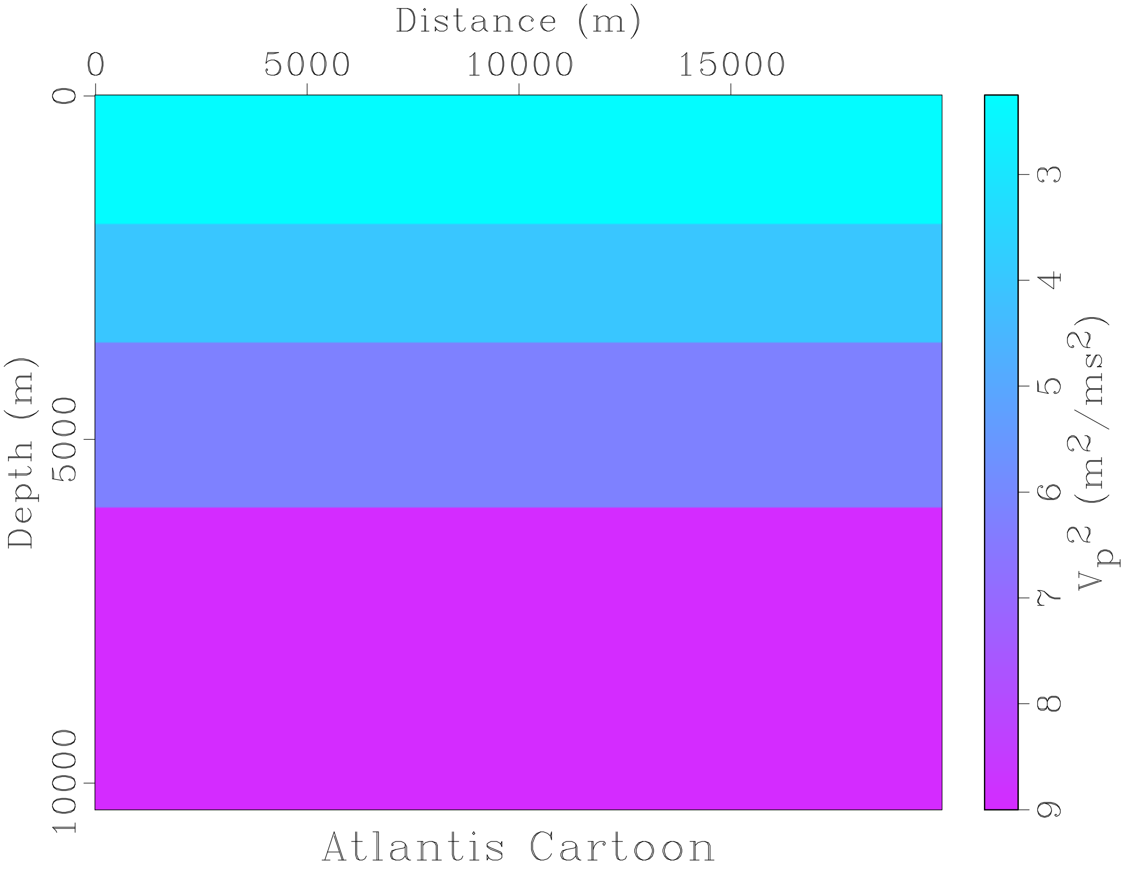


Figure 1: Four layer deep-water cartoon - quantity plotted is velocity squared. struct/project csq-4layer

Movie from Initial Data. Figure 2 displays a Gaussian initial field, input as the initial potential field `uc`. Since `up` is not input, it remains at its default value of zero. The resulting IWAVE solution is not convergent to the solution of any particular Cauchy problem with full convergence order (2), but produces a reasonable-looking movie. Another (minor) application would produce a correction to be added to the `up` field to restore 2nd order convergence.

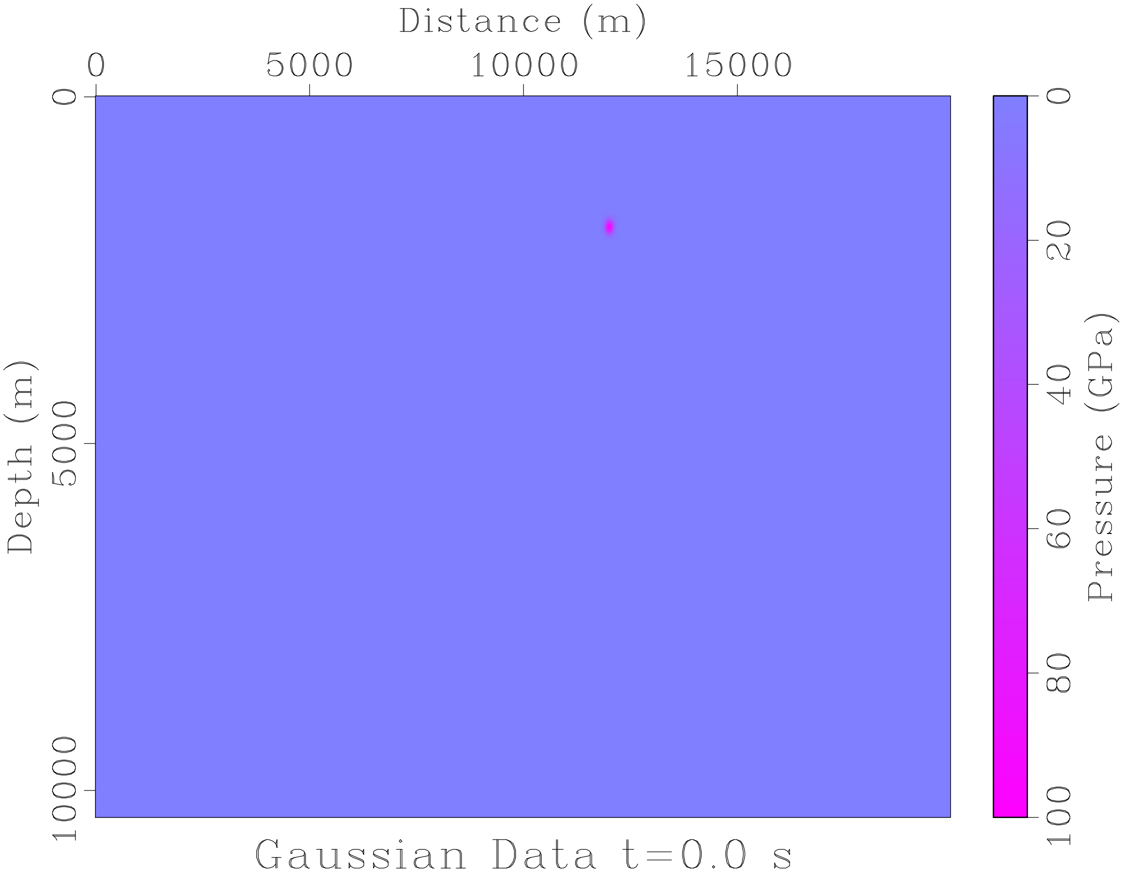


Figure 2: Gaussian initial datum struct/project gauss

The data depicted in Figures 1 and 2 is input to the simulation, so clearly must exist prior to simulation. However the output must also exist: IWAVE I/O, both reads and writes, is driven by the target data structure. Therefore the movie output file must be constructed before the simulation fills it with data. The `SConstruct` script contains an invocation of the `sfmakevel` command which creates a 3D `rsf` file `movieinit.rsf`. On completion of the command, this file holds the movie output.

Perusal of this command reveals some customization of the `rsf` file format, as compared to its standard use (Fomel (2009)). The duration of the movie determines the duration of the simulation: the initial simulation time is the time of the initial movie frame, and similarly for the final time. Thus IWAVE must be able to determine which axis specified in `movieinit.rsf` is the time axis. Three additional header word categories, beyond those of the `rsf` standard, make this feat possible:

- `dim` = spatial dimension - should be same as spatial dimension of the primary static file (keyword `csq` in this case).
- `gdim` = global dimension - of the data array, at least `dim`
- `id1, id2, id3,...`: identification tags of axes 1, 2, 3,...

By IWAVE convention, axis with tag = `dim` is the time axis, and axes 0, 1,..., `dim`-1 are the spatial axes. More than `dim` axes are permitted - they are used to represent other acquisition or modeling parameters, see section on multi-shot simulation, below.

In this example, the space dimension is 2, so `id3=2` indicates that the 3rd axis is the time axis.

After propagating 5.12 s and interacting with both the reflecting (Dirichlet) boundaries and the interfaces in the model, the potential field becomes that depicted in Figure 3.

Parameters passed to the command `acd.x` included

```
  csq = ../csq_4layer.rsf
initc = ../init.rsf
movie = ../movieinit.rsf
```

Keywords `data` and `source` were ignored. Many other paramters were required; a brief description of these is included in the self-doc of the command `acd.x`.

Note that the pathnames refer to the directory level above the working directory. IWAVE produces various diagnostic output at runtime, switched by various flags passed as parameters. These outputs, and possibly other auxiliary outputs of commands built upon IWAVE (eg. the data residual in an inversion) vary with application and data, so are inconvenient to specify individually as cleanup targets. Instead, the SConstruct script creates a working subdirectory and executes (and dumps its auxiliary output) there. The entire directory is cleaned up by `scons -c`. So the correct parameter specification for archival input and output files is one directory level up.

Movie from Point Source. The pulse in Figure 2 the derivative of a 5 Hz Gaussian; it is embedded in a space-time field via the `iwave/trace/main/towed_array.x` utility, and used as the RHS in the wave equation. The resulting field (starting from homogeneous Cauchy data) at 5.12 s is depicted in Figure 5.

In this case, the parameters passed to `acd.x` (in addition to the other required numerical parameters) are

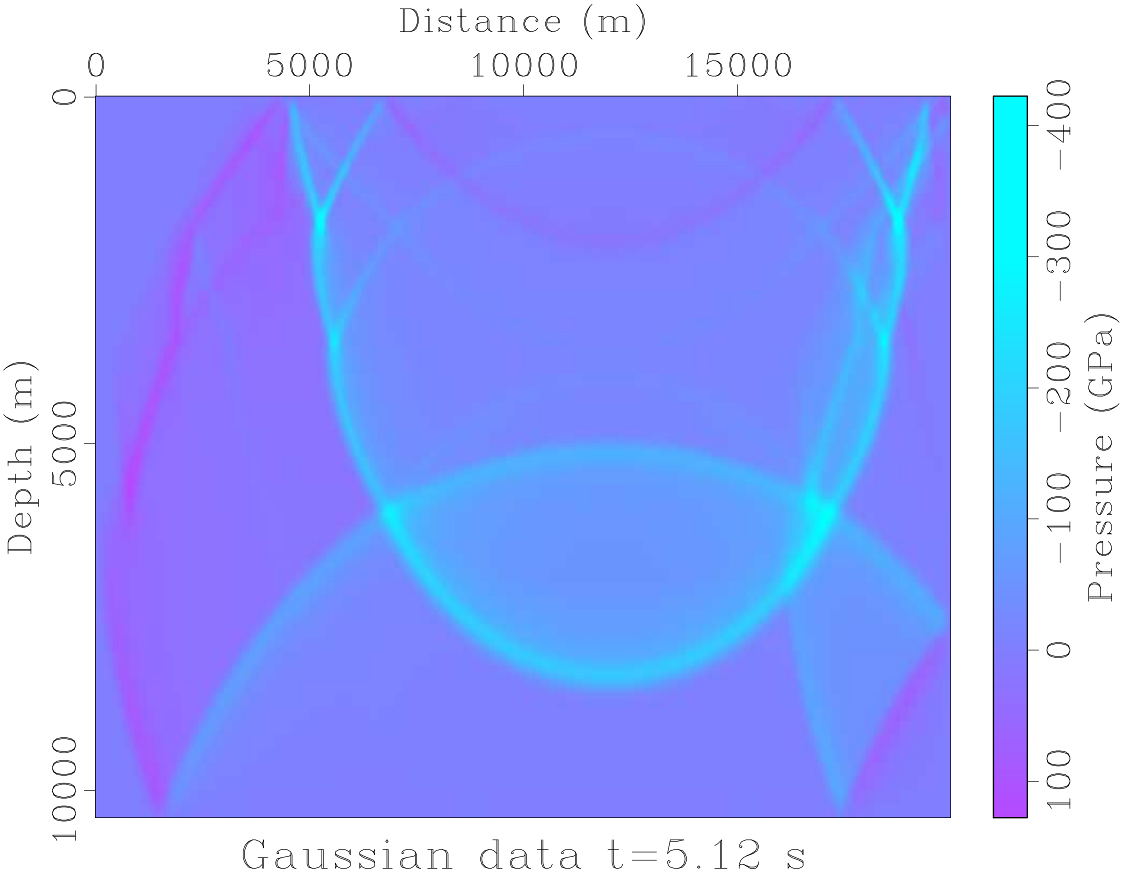


Figure 3: Acoustic potential field at 5.12 s, resulting from Gaussian initial data struct/project frameinit

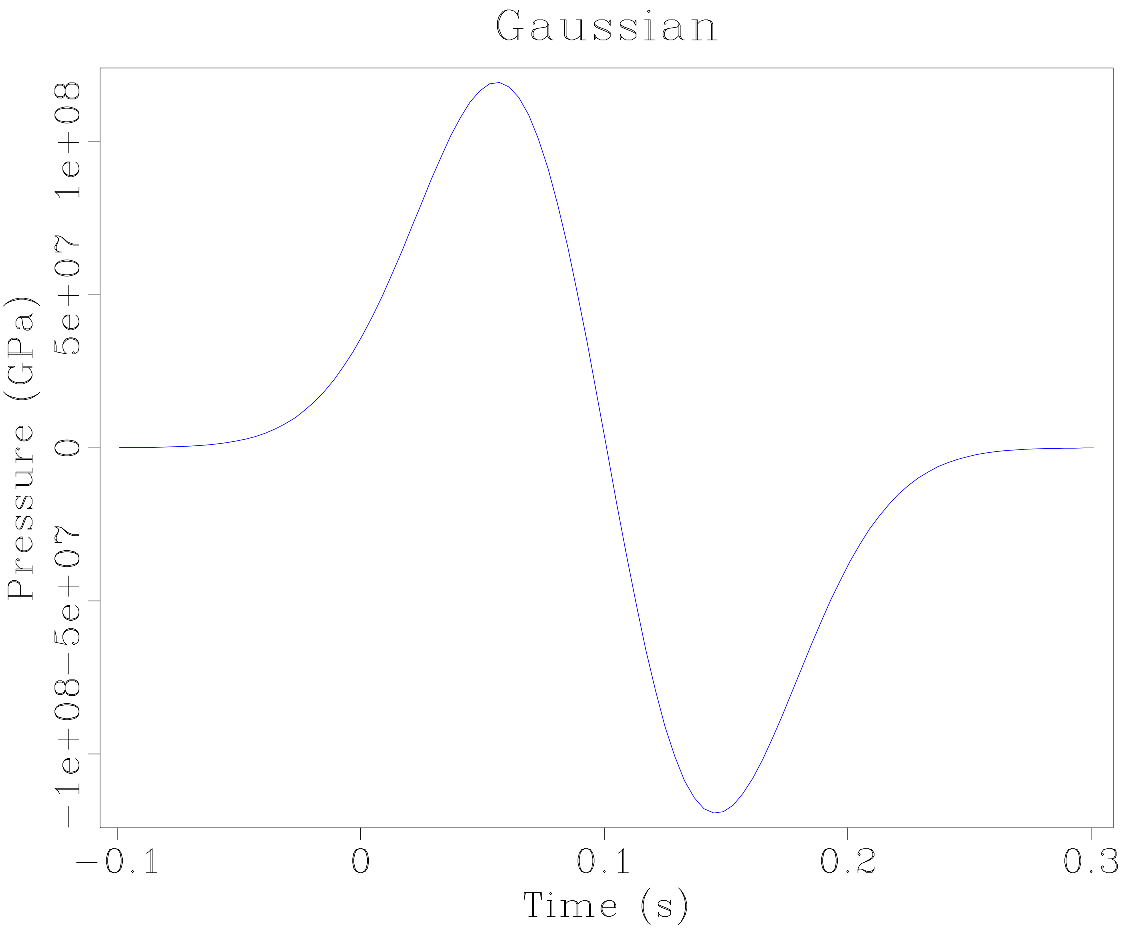


Figure 4: Derivative of 5 Hz Gaussian $\boxed{\text{struct/project wavelet}}$

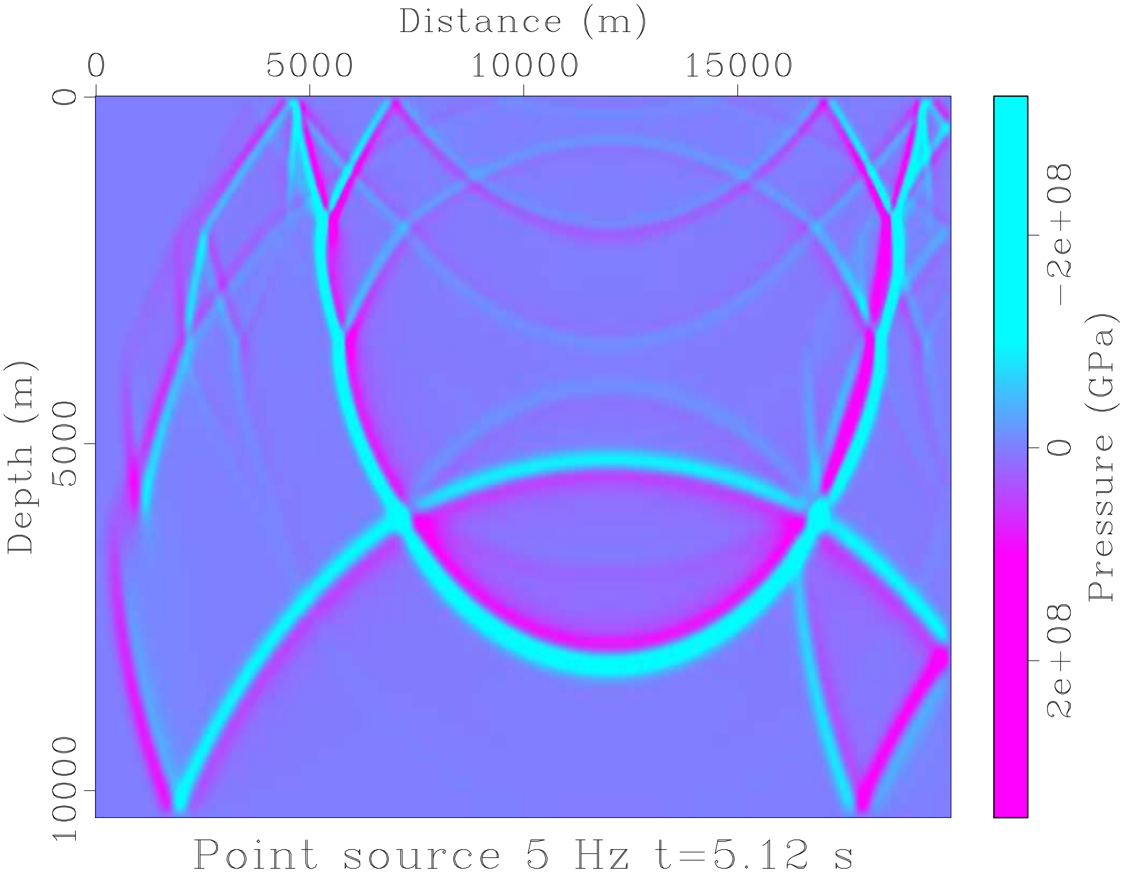


Figure 5: Acoustic potential field at 5.12 s, resulting from Gaussian derivative point source pulse located at x=12 km, z = 1.875 km. struct/project framesrc

```
   csq = ../csq_4layer.rsf
source = ../wavelet12000.su
 movie = ../movieinit.rsf
```

The point source data are presented as SEGY traces, rather than as an RSF data file; the latter would in principle also be possible, but at the time of this writing RSF trace I/O is not enabled.

Trace Data from Point Source The same point source as in the previous example produces sthe traces at 6 m depth depicted in Figure 6. The sampling is also pure point, that is, does not filter the potential field in any way. So this trace data could be interpreted as the 2D pressure traces of a point source with 5 Hz Gaussian pulse, for example. In this case, the required parameters are

```
   csq = ../csq_4layer.rsf
source = ../wavelet12000.su
  data = ../shot12000.su
```

As noted before, there is nothing sacred about the string `data` - any keyword identified in the I/O key struct as output from `uc` would do just as well. This key is provided for mnemonic convenience. Note that the file data structure (`shot12000.su`) entirely determines the *way* in which the data is written - sampling, source and receiver geometry are all taken from this file.

Born Approximation The linearization of the acoustic field with respect to the wave velocity is the solution δu of

$$\frac{\partial^2 u}{\partial t^2} - c^2\nabla^2 u = f, \tag{1}$$

$$\frac{\partial^2 \delta u}{\partial t^2} - c^2\nabla^2 \delta u = \frac{2\delta c}{c^3}\frac{\partial^2 u}{\partial t^2}, \tag{2}$$

$$u(\mathbf{x},0) = u_0(\mathbf{x}),\ \frac{\partial u}{\partial t}(\mathbf{x},0) = v_0(\mathbf{x})\ \mathbf{x} \in \Omega,$$

$$\delta u(\mathbf{x},0) = 0,\ \frac{\partial \delta u}{\partial t}(\mathbf{x},0) = 0\ \mathbf{x} \in \Omega,$$

$$u(\mathbf{x},t) = \delta u(\mathbf{x},t) = 0,\ \mathbf{x} \in \partial\Omega.$$

As explained above, the IWAVE framework provides approximations for computing the linearization (widely called the "Born approximation", even though this is strictly a misnomer), along with its adjoint and higher derivatives.

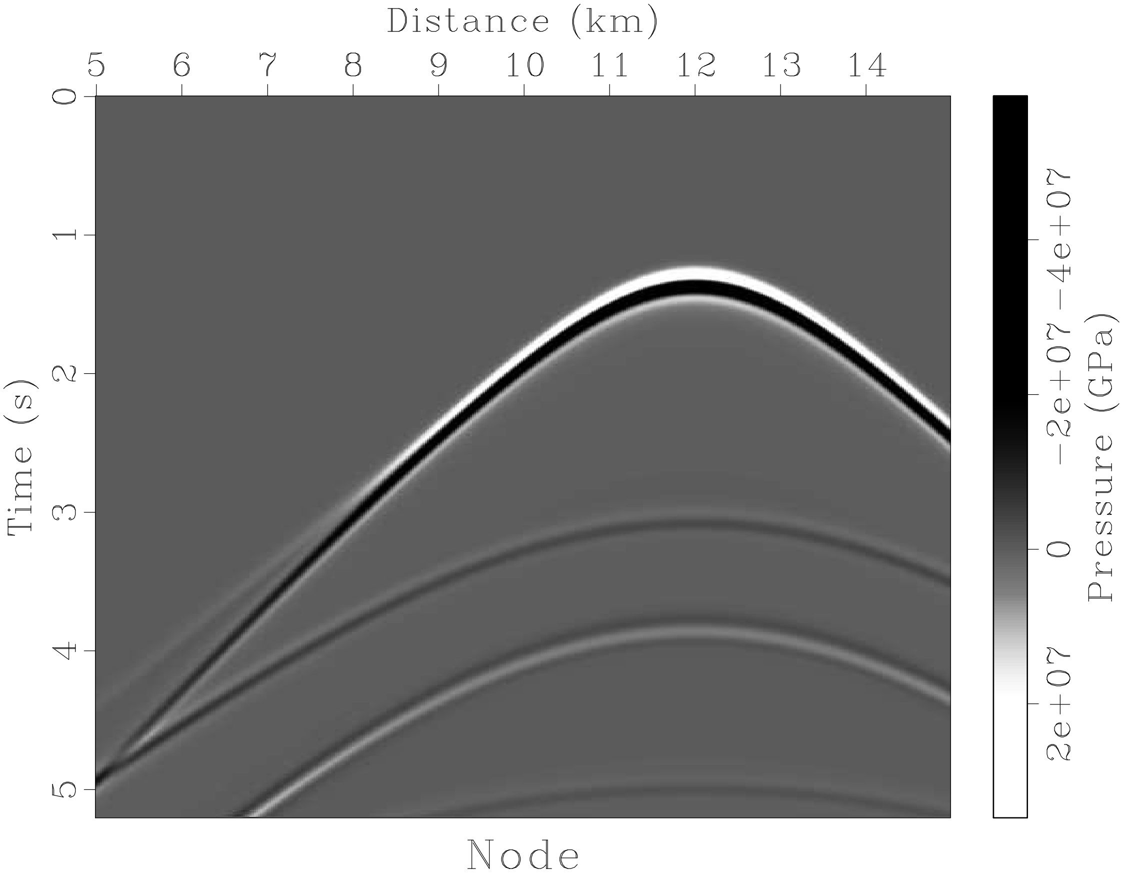


Figure 6: Response of seafloor pressure sensor at 1.875 km depth, to 400 shots spaced 25 m apart at 6 m depth, source is isotropic point radiator, source pulse is Gaussian derivative with peak frequency 5 Hz, struct/project shot12000

The parameter key `deriv` flags the computation, or not, of derivatives. The value assigned is the order of derivative, with default 0. Each input perturbation (representing quantities such as δc in the linearized system of PDEs, above) is assigned a key equal to the key for the unperturbed quantity with `_d1` appended (for the first derivative - higher derivatives require multiple input perturbations, keys for which have `_d2`, `_d3`,... appended). Output keys remain the same as for the reference computation.

For the acoustic constant density application, Born approximation requires reference and perturbation square velocity fields. Figures 7 and 8 show perturbation and reference square velocity fields, respectively, that will generate Born data roughly corresponding to the preceding example. The required parameters are

```
  deriv = 1
    csq = ../csq_4layer.rsf
 csq_d1 = ../dcsq_4layer.rsf
 source = ../wavelet12000.su
   data = ../born12000.su
```

The linearized response (Born modeling) corresponding to Figure 6 appears as Figure 9.

Reverse Time Migration One version of Reverse Time Migration (RTM) is simply the adjoint of Born modeling. IWAVE provides adjoint computations for every derivative mapping (first, second,...) using the *optimal checkpointing* method of reverse time propagation (Griewank, 2000; Blanch et al., 1998; Plessix, 2006; Symes, 2007). Other approaches to time reversal can be more efficient in special cases, especially when the interior dynamics are conservative (acoustics, elasticity) (Dussaud et al., 2008; Clapp, 2009). However none are more effective in general, in particular when energy attenuation is significant part of wave dynamics, as is the case for all realistic models of seismic wave motion.

Figure 10 displays the migration of the single Born "shot" gather (really, OBS receiver gather) located at $x_s = 12000$ m from the left edge of the model. No effort has been made to remove the low-frequency noise caused by the sea bottom reflection.

The parameters required for this job are

```
  deriv = 1
adjoint = 1
 nsnaps = 10
    csq = ../csq_4layer.rsf
 csq_b1 = ../migr12000.rsf
 source = ../wavelet12000.su
   data = ../born12000.su
```

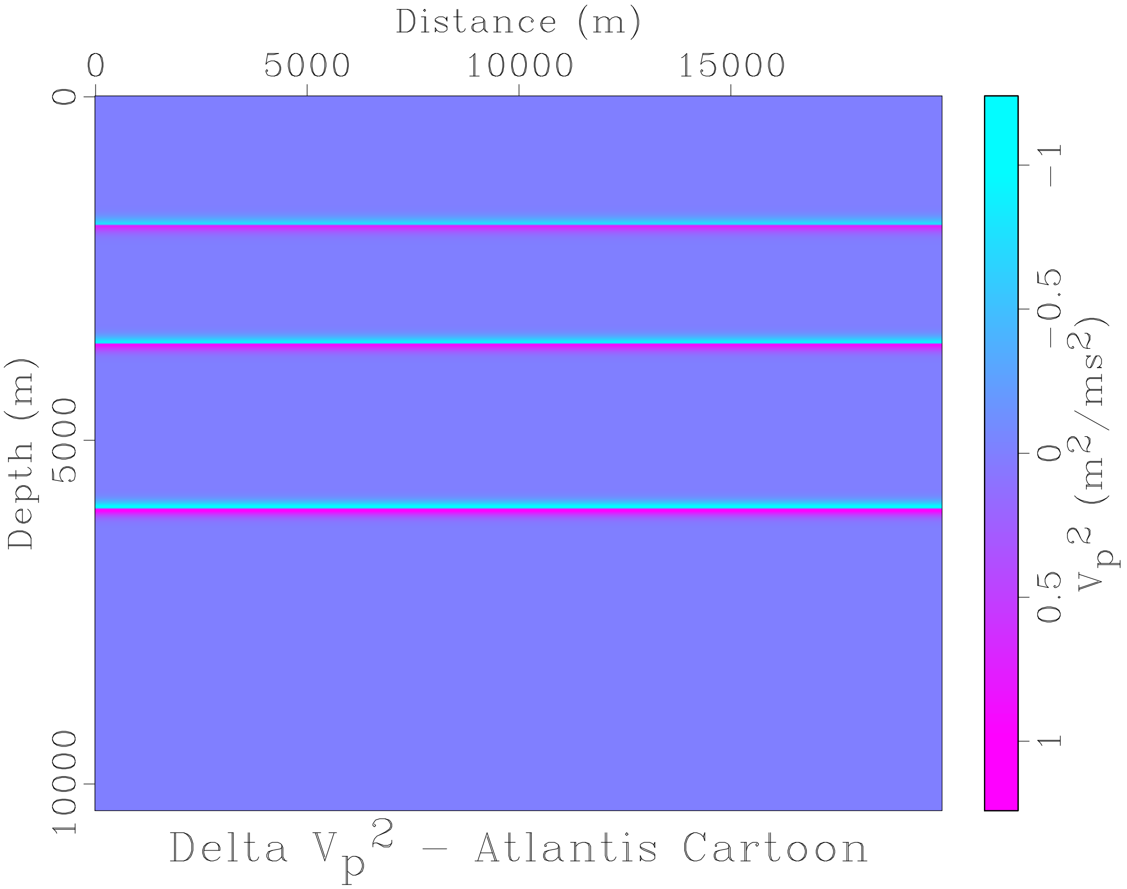


Figure 7: Velocity-squared perturbation - localized oscillations at layer boundaries struct/project dcsq-4layer

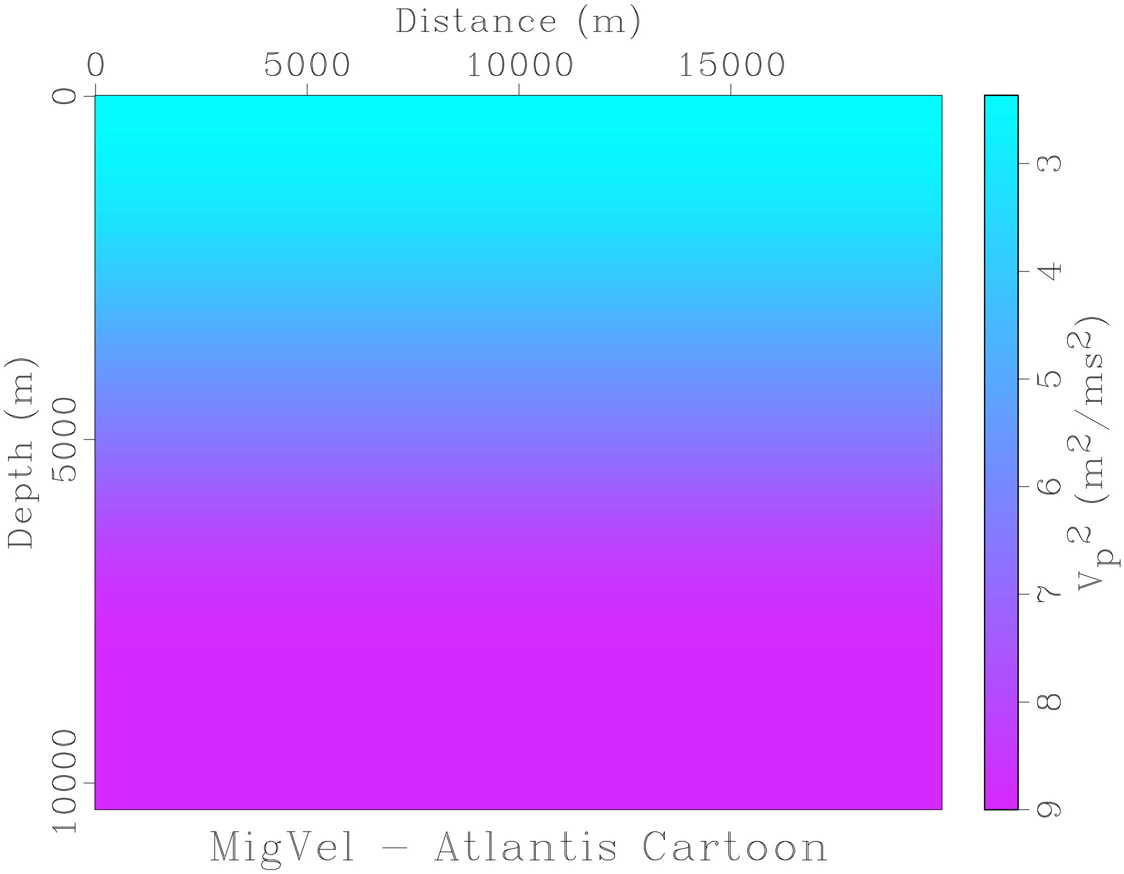


Figure 8: Smooth velocity-squared obtained from velocity of Figure 1 by filtering with a cubic spline window. struct/project csq-4layersm

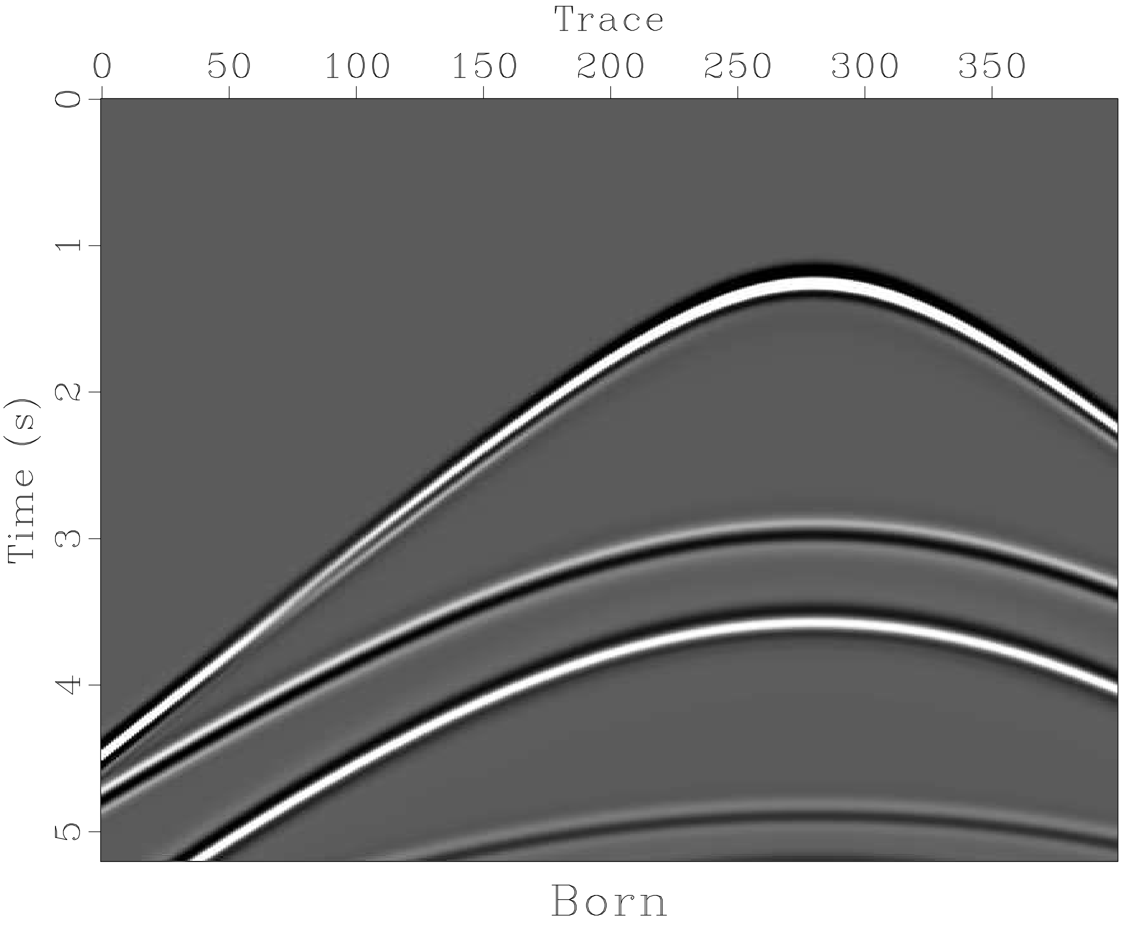


Figure 9: Linearized response ("Born modeling") of seafloor pressure sensor, due to perturbation (Figure 7) about smooth background (Figure 8); other parameters as in Figure 6. struct/project born12000

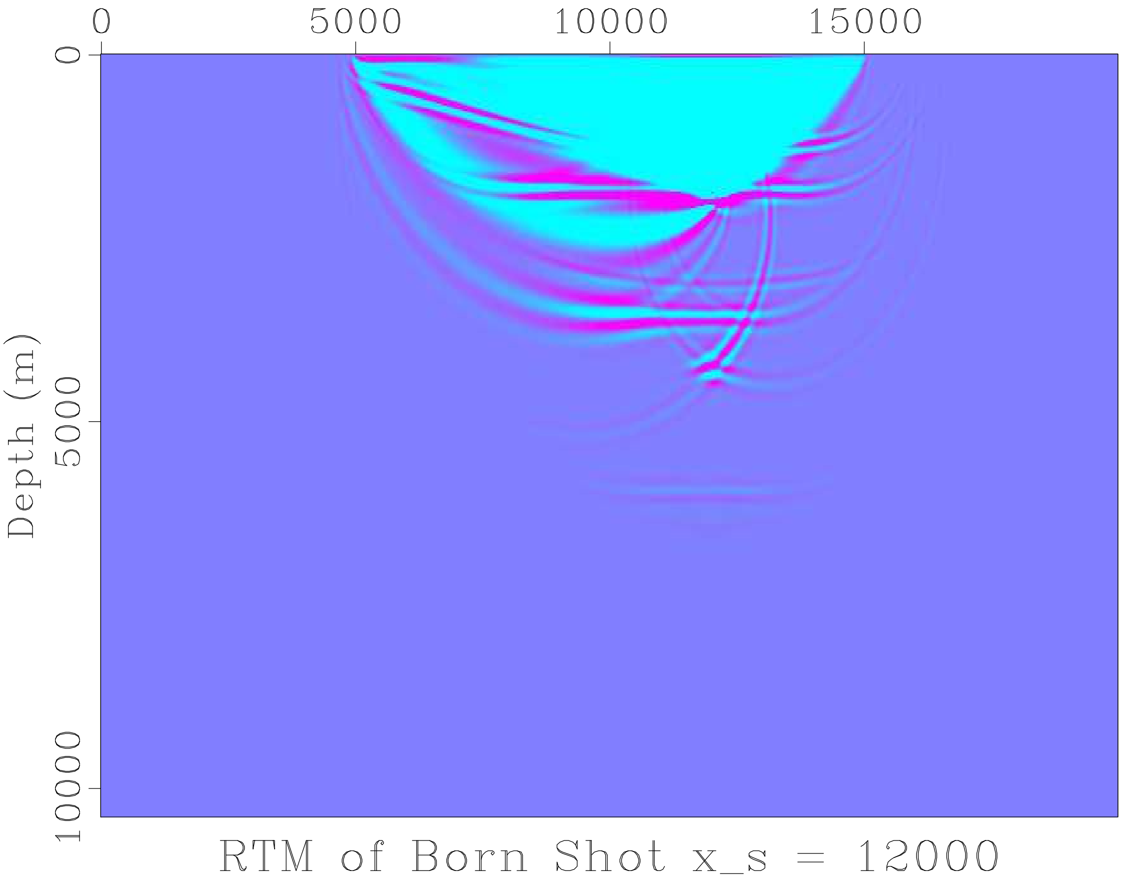


Figure 10: Reverse-time migration of Born data from Figure 9. struct/project migr12000

The `adjoint` key flags the adjoint computation. The checkpointing algorithm requires allocation of workspace for checkpoints (copies of wavefield Cauchy data, consisting of all dynamic arrays). The number of checkpoints allocated is the value for key `nsnaps`. The appropriate number of checkpoints depends on the number of time steps. Reasonable numbers to achieve a cost ratio of adjoint to forward computations of around 5 are

- up to 1000 time steps: 5 checkpoints
- up to 5000 time steps: 10 checkpoints
- up to 10000 time steps: 20 checkpoints
- up to 20000 time steps: 30 checkpoints

Interlude: The Internal Grid

The basic control structure of IWAVE is its *internal grid*, which describes the global state space of IWAVE simulations. This rectangular grid is logically equivalent to a tuple of axes; each axis is regularly sampled, hence described by the usual (n, d, o) triple familiar from RSF. [In common with conventional usage for this subject, "axis" in this discussion will mean "finite regularly sampled interval on an axis".]

Like RSF, the restriction to regular sampling implies some workarounds for cases in which data sampling is actually not regular - the notable instance being trace acquisition geometry. Irregular sampling parameters find a natural home in the attributes of `IWaveSampler` i/o objects, which encapsulate all of the information required for i/o of a given type or format. A reference to the internal grid is an key data member of every `IWaveSampler` instance.

The spatial axes are obvious members of the internal grid, as is the time axis. The `IWaveSim` constructor creates a version of the time axis that combines the various time axes of associated data objects that have them, and resets the time step to one appropriate for the simulation (returned by the `TIMESTEP` member function of `IWaveInfo`). Thus for example the source time axis may even end earlier than the output trace time axis begins; the internal time axis contains the convex hull the two, resampled to the simulation time step.

Other axes represent other aspects of simulation - most important, other sampled parameters such as source positions that parametrize individual shots within a simulation. The internal grid allocates additional axes to record these parameters. For instance, SEGY data is organized shot coordinates, on a *shot axis*. All coordinates are recorded in the `IWaveSampler` for SEGY data, but only one additional axis with stride 1 is introduced to describe the various shots.

Additional axes may represent internal degrees of freedom available to the simulation, for instance space- or time-shift simulation (Symes, 2008). The use of these internal degrees of freedom will be described in another report.

With many axes interacting, it is essential to provide some method to determine their roles. Other data handling systems either carry out this task in a similar way to that described here (eg. DDS,...) or ignore it (not an option for IWAVE!). The additional information is arranged as additional keywords in RSF data files, mentioned earlier in the discussion of movie output:

- `dim` = spatial dimension of the simulation
- `gdim` = global dimension = dimension of internal grid
- `id1, id2, id3,...` = axis identifiers, coded as follows:
  - `0,...,dim-1`: spatial dimensions corresponding to these choices in SEGY coordinates: 0=z, 1=x, 2=y;
  - `dim`: time
  - `dim+1,...,99`: additional axes for various simulation parameters (eg. shot index)
  - `100,...,dim+99`: internal degrees of freedom (extended modeling)

The labeling of spatial axes avoids external data transpose: if the data for coefficient arrays is organized with x as the fast variable, for instance (x meaning the coordinate with that name in the SEGY data standard, for example), then setting `id1=1` etc. enables correct sampling of traces. The default settings are `id1=0, id1=2, id2=3`. [Note that axis keywords are numbered Fortran-style, whereas axis values are C-style (beginning with 0). This is done to maintain compatibility with RSF header keywords, which appear to be inherited from SEPlib.]

Multi-shot (survey) simulation

`IWaveSim` loops over any axes beyond the time axis (signified by `idxxx=dim` through `idxxx=99`) that `IWaveSampler` objects add to the internal grid. In particular, SEGY data file identified as output or input adds a simulation axis with `idxxx=dim+1` to the internal grid. The loop over this axis increments when a the keywords `sx`, `sy`, or `sz` change from one trace to the next, signifying a new shot.

From the user point of view, this means that multi-shot simulation is automatic: if multiple shots are part of the output data structure, then all shots will be simulated.

We show several examples that illustrate this data-flow feature of IWAVE. The examples are identical to those reviewed above for single shots, except that the additional computation load of multi-shot simulation suggests the use of parallelism. The parallel features of IWAVE (parallel shots, parallel subdomains, parallel loops) will be the subject of a subsequent report. For now, we note that the `partask` keyword indicates the number of shots to run in parallel. The `SConstruct` files for the several multishot simulations include a line (near the top) to set the variable `NP`. If `NP=1`, then the simulations described below are run in serial mode. If `NP` is set to a value larger than one, then this value indicates the number of shots to process in parallel, via a collection of MPI communicators. Running several shots in parallel requires that IWAVE be installed with MPI enabled `IWAVE_USE_MPI` defined as a compiler parameter, see the `README_INSTALL` file in the top-level directory. The number of MPI processes assigned (via `mpirun -np` can be fewer than the number of shots to be simulated - in that case the simulations run in batches until all shots are completed. Any unnecessary processes at the terminal stage of the simulation are simply left idle, so there is no necessary relation between the number of MPI processes and the number of shots. The `SConstruct` script in the project directory for this paper uses `mpirun -np NP` to initiate MPI and assign the number of processes to be used. The follow-on report will describe the use of IWAVE in a batch environment, for both parallelization over shots and via domain decomposition.

The next few examples are large enough that completion single-threaded execution requires perhaps half an hour on a typical (circa 2014) desktop CPU. The data displayed were obtained on a typical multicore desktop machine, using MPI with `NP=6`. These results are precisely the same as those that are obtained with a single process, but required less than 4 minutes walltime.

Simulation Recall that output files for IWAVE applications must exist prior to execution. Therefore, to simulate 12 shots at intervals of 400 m over the model depicted in Figure 1, the first step is to create a suitable SEGY data set to hold the output. Therefore the `SConstruct` script creates the output file of zero SEGY traces (`line8-12km.su`), subsequently filled with samples by IWAVE, all as part of the same `Flow` command. As note above, the source must be properly embedded in a space-time field (collection of SEGY traces) for each shot location. The utility `trace/main/towed_array.x` provides a convenient method for combining the position data of a prototype SEGY trace file with a choice of source array to produce translations of a source array to correct shot positions. Other parameters are as in the single-shot case. The result is displayed in Figure 11.

Perusal of the `project/SConstruct` script shows that in the invocation of `acd.x`, only the filenames have changed. The dataflow design of IWAVE, in which data objects determine the way in which they are read/written, implies that the information about changed source/receiver geometry and data volume does not need to be explicitly passed to the simulation command.

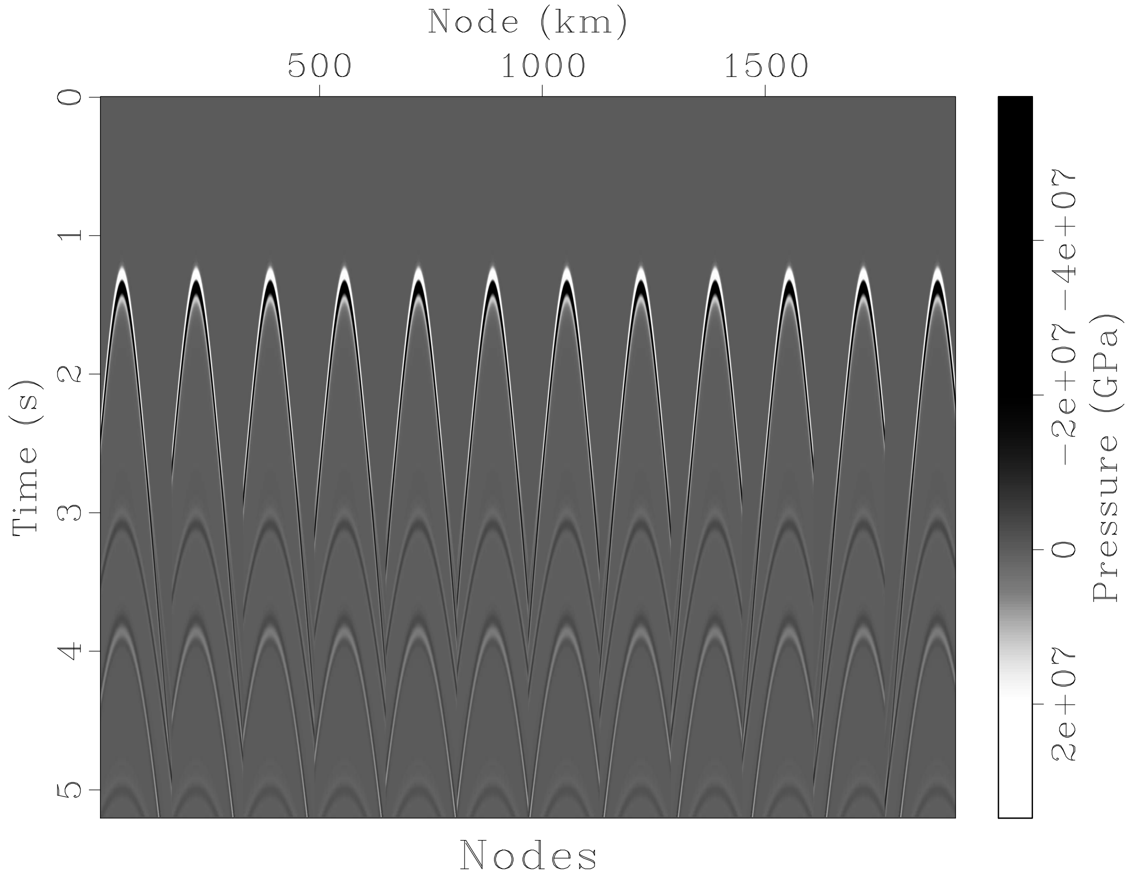


Figure 11: Response of 13 seafloor pressure sensors spaced 400 m apart at 1.875 km depth, to 400 shots spaced 25 m apart at 6 m depth, sources are isotropic point radiators, source pulse is Gaussian derivative with peak frequency 5 Hz. Velocity-squared model depicted in Figure 1. struct/project line8-12km

Linearized Simulation Linearized simulation (Born modeling) works the same way. The data predicted from the perturbation in Figure 7 of the smooth background velocity (squared) in Figure 1, same source-receiver geometry as in the preceding example, is displayed in Figure 12.

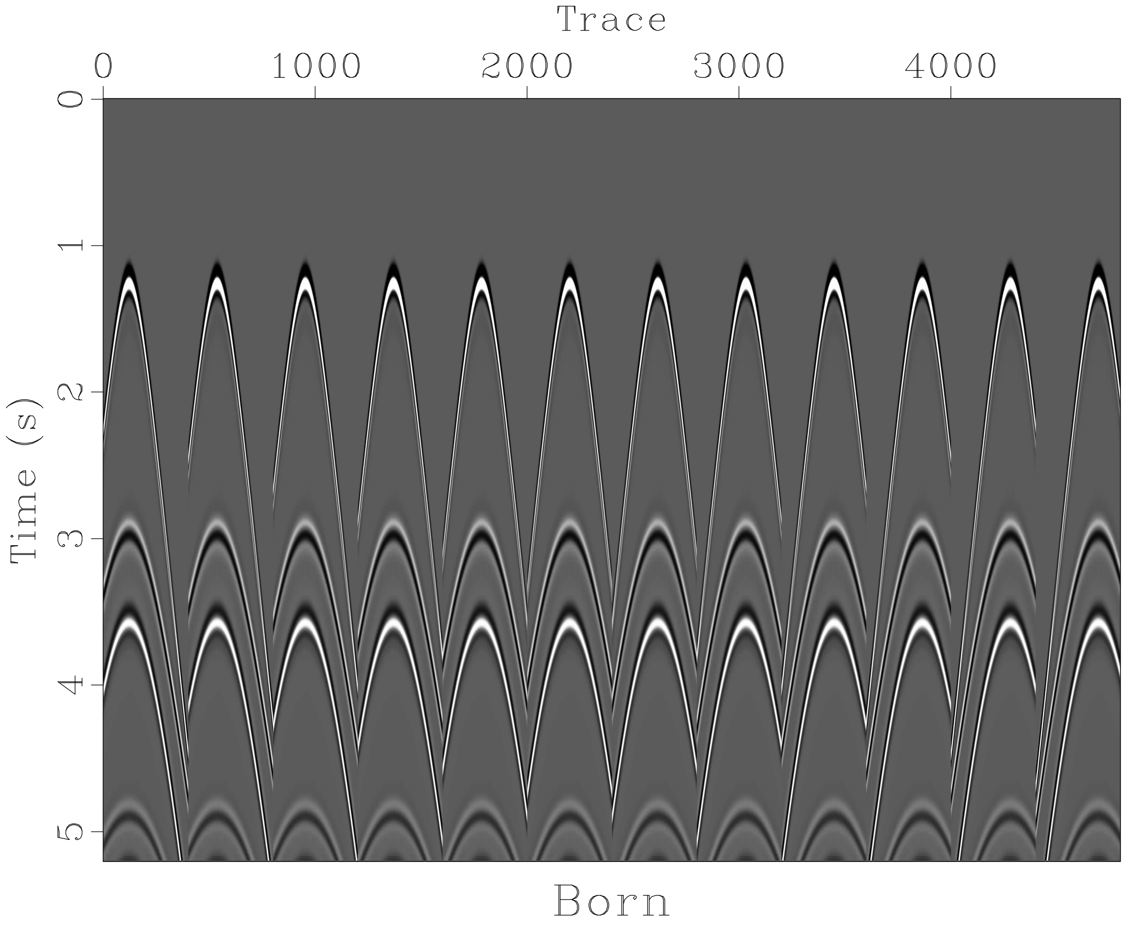


Figure 12: Linearized response of 13 seafloor pressure sensors space 400 m apart at 1.875 km depth, to 400 shots spaced 25 m apart at 6 m depth, sources are isotropic point radiators, source pulse is Gaussian derivative with peak frequency 5 Hz, Background or reference model as in Figure 8, perturbation as in Figure 7. struct/project born8-12km

Adjoint Linearized Simulation Adjoint linearized simulation (reverse time migration) of the 12 shot linearized data (Figure 12) produces the image displayed in Figure 13.

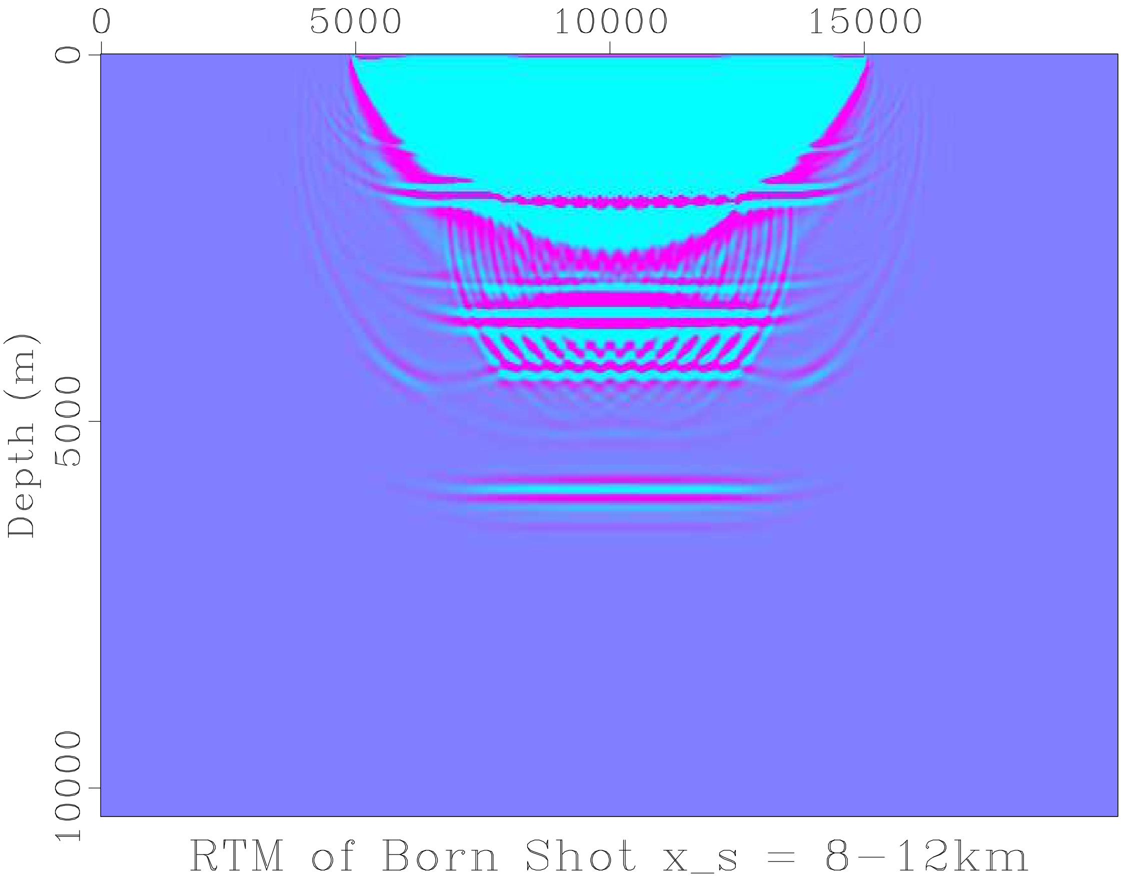


Figure 13: Reverse-time migration of Born data from Figure 12. Reflectors are correctly positioned and clearly visible, along with migration artifacts characteristic of raw OBS migration, for example a free-surface multiple image slightly above 5000 m depth. struct/project migr8-12km

CONCLUSION

This paper has described the overall structure of the IWAVE framework, and illustrated a few of its capabilities.

The release of IWAVE described in this paper is 2.0. It implements many use cases not described here:

- multipole source representation - essential to simulate the directional nature of field source and receiver arrays;
- plane wave and other extended sources, including randomized;
- PML absorbing boundary conditions for acoustic constant density simulation, following Grote and Sim (2010);
- shot record extension of the acoustic constant density model. (Kern and Symes, 1994)

All of these simulation modes are as accesssible as easily as those illustrated in the preceding seciton.

Earlier releases (1.x) of IWAVE included implementations of staggered grid schemes for acoustics and isotropic linear elasticity (Moczo et al., 2006). See Fehler and Keliher (2011) for an account of the role played by the IWAVE acoustic staggered grid code as a QC tool in the SEAM Phase I project. Revival of these applications in the IWAVE 2.0 infrastruction is intended for a future release.

The QC role in the SEAM project heavily influenced the design of IWAVE, particularly the requirement that its performance scale well to very large simulations and large numbers of threads. All versions of IWAVE have offered parallelism via domain decomposition: SEAM simulations typically split simulations involving eight fields and 3000 × 3000 × 1500 grid points amongst 1000-4000 processes. The user interface specifies the subdivision of the domain with a few parameters. Another report will detail domain decomposition, parallelization over simulations (mentioned above), and multithreaded parallel loop execution aspects of IWAVE, as well as performance enhancements implemented in the finite difference kernels.

Another ground-up design goal is to provide a simple interface to inversion software. Symes et al. (2011) describe the concept behind the data interchange mechanism: it is file-based, and motivates the dataflow design mentioned several times already. Since data objects, represented as files or collections of files, determine their own i/o modalities, inversion software can simply communicate pathnames to the IWAVE interface, rather than some more complex data structure. The current release couples to (and in fact depends on) the Rice Vector Library ("RVL") (Padula et al., 2009), a framework for linear algebra and optimization. IWAVE 2.0 includes full waveform inversion, linearized inversion ("least squares migration"), and shot record extended linearized inversion or differential semblance optimization. These and other inversion applications of IWAVE will be discussed in other reports.

ACKNOWLEDGEMENTS

IWAVE has been a team effort: the original design of the core framework is due to Igor Terentyev, and Tanya Vdovina, Dong Sun, Marco Enriquez, Xin Wang, Yin Huang, Mario Bencomo, Lei Fu, and Muhong Zhou have each made major contributions. Development of IWAVE was supported by the SEG Advanced Modeling (SEAM) project, by the National Science Foundation under awards 0620821 and 0714193, and by the sponsors of The Rice Inversion Project. The IWAVE project owes a great deal to several open source seismic software packages (Seismic Un*x, SEPlib, Madagascar), debts which we gratefully acknowledge. The author wishes to record his special gratitude to Sergey Fomel, the architect of Madagascar, for his inspiring ideas and his generous and crucial help in the integration of IWAVE into Madagascar.

REFERENCES

Blanch, J., W. Symes, and R. Versteeg, 1998, A numerical study of linear inversion in layered viscoacoustic media, *in* Comparison of Seismic Inversion Methods on a Single Real Dataset: Society of Exploration Geophysicists.

Clapp, R. E., 2009, Reverse time migration with random boundaries: 80th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2809–2813.

Dussaud, E., W. Symes, P. Williamson, L. Lemaistre, P. Singer, B. Denel, and A. Cherrett, 2008, Computational strategies for reverse-time migration: 79th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2267–2271.

Fehler, M., and J. Keliher, 2011, SEAM Phase 1: Challenges of subsalt imaging in Tertiary basins, with emphasis on deepwater Gulf of Mexico: Society of Exploration Geophysicists.

Fomel, S., 2009, Madagascar web portal: http://www.reproducibility.org, accessed 5 April 2009.

Griewank, A., 2000, Evaluating derivatives: Principles and techniques of algorithmic differentiation: Society for Industrial and Applied Mathematics (Frontiers in Applied Mathematics 19).

Grote, M., and I. Sim, 2010, Efficient pml for the wave equation: Technical Report arXiv1001.0319v[math.NA].

Hascoët, L., and V. Pascual, 2013, The Tapenade Automatic Differentiation tool:

Principles, Model, and Specification: ACM Transactions On Mathematical Software, **39**.

Kelly, K. R., R. W. Ward, S. Treitel, and R. M. Alford, 1976, Synthetic seismograms: A finite-difference approach: Geophysics, **41**, 2–27.

Kern, M., and W. Symes, 1994, Inversion of reflection seismograms by differential semblance analysis: Algorithm structure and synthetic examples: Geophysical Prospecting, **99**, 565–614.

Moczo, P., J. O. A. Robertsson, and L. Eisner, 2006, The finite-difference time-domain method for modeling of seismic wave propagation: Advances in Geophysics, **48**, 421–516.

Padula, A. D., W. Symes, and S. D. Scott, 2009, A software framework for the abstract expression of coordinate-free linear algebra and optimization algorithms: ACM Transactions on Mathematical Software, **36**, 8:1–8:36.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

Symes, W., 2007, Reverse time migration with optimal checkpointing: Geophysics, **72**, SM213–222.

———, 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790.

Symes, W. W., D. Sun, and M. Enriquez, 2011, From modelling to inversion: designing a well-adapted simulator: Geophysical Prospecting, **59**, 814–833. (DOI:10.1111/j.1365-2478.2011.00977.x).

Planewave Modeling and Migration with IWAVE

William. W. Symes

ABSTRACT

Since IWAVE accepts any forcing term, or source, modeling plane waves simply involves creating a plane wave source. This paper describes the `planewave` utility: it creates both plane wave source files and prototype output (header) files for plane wave data. The examples show both how to perform ordinary plane wave modeling and migration, and also how to create an extended plane wave image volume for use in velocity analysis and AVO.

INTRODUCTION

This paper describes the IWAVE implementation of plane wave modeling and migration. The examples use constant density acoustic modeling, but the same principles apply to plane wave modeling based on any wave physics implemented in IWAVE.

IWAVE is a framework for solving time-dependent partial differential equations by Finite Element or Finite Difference methods. The current implementation focuses exclusively on uniform grid FD methods. A simple example of the target problem class is constant density acoustics, governed by the acoustic wave equation

$$\left(\frac{\partial^2 u}{\partial t^2} - c^2\nabla u\right)(s;t,\mathbf{x}) = f(s;t\mathbf{x}). \tag{1}$$

together with appropriate initial and boundary conditions, defining a family of fields depending on a source parameter s. Plane wave modeling results from the choice

$$\begin{aligned} s &= (p_x,p_y), & (2)\\ f(s;t,z,x,y) &= \delta(z-z_s)w(t-p_xx-p_yy) & (3) \end{aligned}$$

or, for 2D,

$$\begin{aligned} s &= (p_x), & (4)\\ f(s;t,z,x) &= \delta(z-z_s)w(t-p_xx) & (5) \end{aligned}$$

The 2D plane wave source (4), for example, produces a planar wave at a source point (z_s,x) propagating at an angle of $\theta = \arcsin c(z_s,x)p_x$ to the vertical. Plane waves propagate only in the region in which $|cp_x| < 1$. Thus a plane wave will transit the entire region only if $|c_{\max}p_x| < 1$. This criterion places an intrinsic limit on the image region for migration.

The basic acoustic constant density driver `acd` computes approximate solutions to (1) for arbitrary right-hand side families $f(s; t, \mathbf{x})$, so plane wave simulation requires another utility to produce appropriate right hand side families. As the notation in (1) suggests, the field also depends on the parameter s, so storage of the simulation output requires a data structure including parametrization by s. The utility `planewave` does both tasks, producing a compatible pair of SEGY files for input to `acd` (and, with appropriate modification, to other IWAVE modeling tools).

Any solution of the acoustic or elastic wave equations in a homogeneous material model may be decomposed into propagating plane waves: this observation is fundamental to a basic understanding of these equations (Courant and Hilbert, 1962; Achenbach, 1973). Plane wave modeling has a long history in seismology: classic references include Aki and Richards (1980), Diebold and Stoffa (1981), Stoffa et al. (1981), Treitel et al. (1982), and Carrion et al. (1984). Plane waves are a persistent waveform for wave propagation in stratified or layered media, and accordingly there is a large literature on analysis and inversion for layered models based on this observation. More recently, Dong Sun used plane wave modeling in his pathbreaking investigation of nonlinear image domain inversion (Sun and Symes, 2009, 2012). For imaging (or inversion), plane wave sources have the advantage over point sources of illuminating a substantial part of the model, rather than a relatively narrow beam: as the examples to be presented in this paper will show, plane wave images consist largely of image, with relatively small volume occupied by edge diffraction, unlike point source images. The chief disadvantage of plane wave sources is the necessary extension of the simulation time axis: some parts of the plane wave are activated earlier than others (see (4)!), whereas time-stepping methods must initiate when any of their dynamic fields become nonzero. Possibilities for ameliorating this added simulation expense exist, for instance broken and time-shifted plane waves or spatial sinusoid modulation (Soubaras and Gratacos, 2007).

The `SConstruct` file in the `project` subdirectory contains complete annotated scripts for producing the examples shown here. The reader should consult this script, along with the self-docs for `acd` and `planewave` and the basic IWAVE white paper (Symes, 2014) to fill in the usage details not mentioned here. The examples build in a few minutes on any modern workstation or laptop via `scons` in `project` - the reader will need to do this, for example, to view the movie files `movie1p.rsf` and `movie1pm01.rsf`. The examples require that the Madagascar revision number at least 13459.

PLANEWAVE

Since IWAVE permits source input at any point in the spatial modeling domain, there is no need to restrict the source points composing a plane wave emitter to simulation grid points. The `planewave` utility assumes uniformly spaced source and receiver points, arranged in horizontal arrays. This is OK for synthetic experiments. On the TO-DO list for `planewave'` is addition of a facility to take arbitrary source and receiver locations from SEGY data files.

The salient characteristic of each plane wave is slope, or slowness (the vectors (p_x) and (p_x, p_y) in (4), (2)). Since planewave sources are synthetic in any case, there seems little harm in limiting the sampling in slowness to uniform in some sense. Current implementation samples uniformly in slowness. Analysis of layered medium kinmatics suggests that uniform sampling of slowness-squared may be more useful - on the TO-DO list.

Due to a bug in the SU utility `suplane`, to which `planewave` delegates plane wave construction, all plane waves must pivot around the midpoint of the array - that is, the time=0 point in each plane wave will occur at the source array midpoint.

Upshot: `planewave` requires parameters to determine several axes:

- receiver time axis - just `nt` and `ot`, as `dt` is taken from the source pulse header
- receiver horizontal axis (`nx`, `ox`, `dx`)
- receiver depth (`zr`)
- source horizontal axis (`nxs`, `oxs`, `dxs`)
- source depth (`zs`)
- slowness axis (`np`, `op`, `dp`)

Because the `Flow` does not automatically inherit the ambient environment, it is also necessary to pass the path to the SU root directory, `CWPROOT`, via a parameter of the same name.

The source time axis is computed - the user does not specify it. The computation uses the extreme slownesses and the extreme x coordinates of the source array, to compute an interval containing the support of every trace in the entire impulse (spike) plane wave for every slowness. Then `suconv` adds this to the time axis of the pulse to create a time axis containing the support of every trace in every plane wave. As described in (Symes, 2014), IWAVE creates a simulation time axis containing the convex hull of the source time axis and the output data time axis.

EXAMPLES

This section presents several examples of plane wave modeling and migration. I'll describe the key parameter selections; the reader can consult the annotated `SConstruct` file in the project directory for precise details. I use the acoustic constant density driver `acd` in all of the examples below - for usage direction, see the self-doc, or for more extensive discussion (Symes, 2014).

The examples in this section use a subsampled version of the Marmousi model, Figure 1, with the water depth increased to 400 m and a horizontal extension on the left side of the model. The extended geometry has 444 points in the x direction, and extends from $x = 0$ m (the left side) to $x = 10632$ m (the right side). The depth range is $[0, 3000]$ m, so to accommodate the deeper water layer some of the original model is dropped off the bottom. The spatial sample rate is 24 m in each direction. Horizontal locations are measured from the left edge, from 0 to 10632 m.

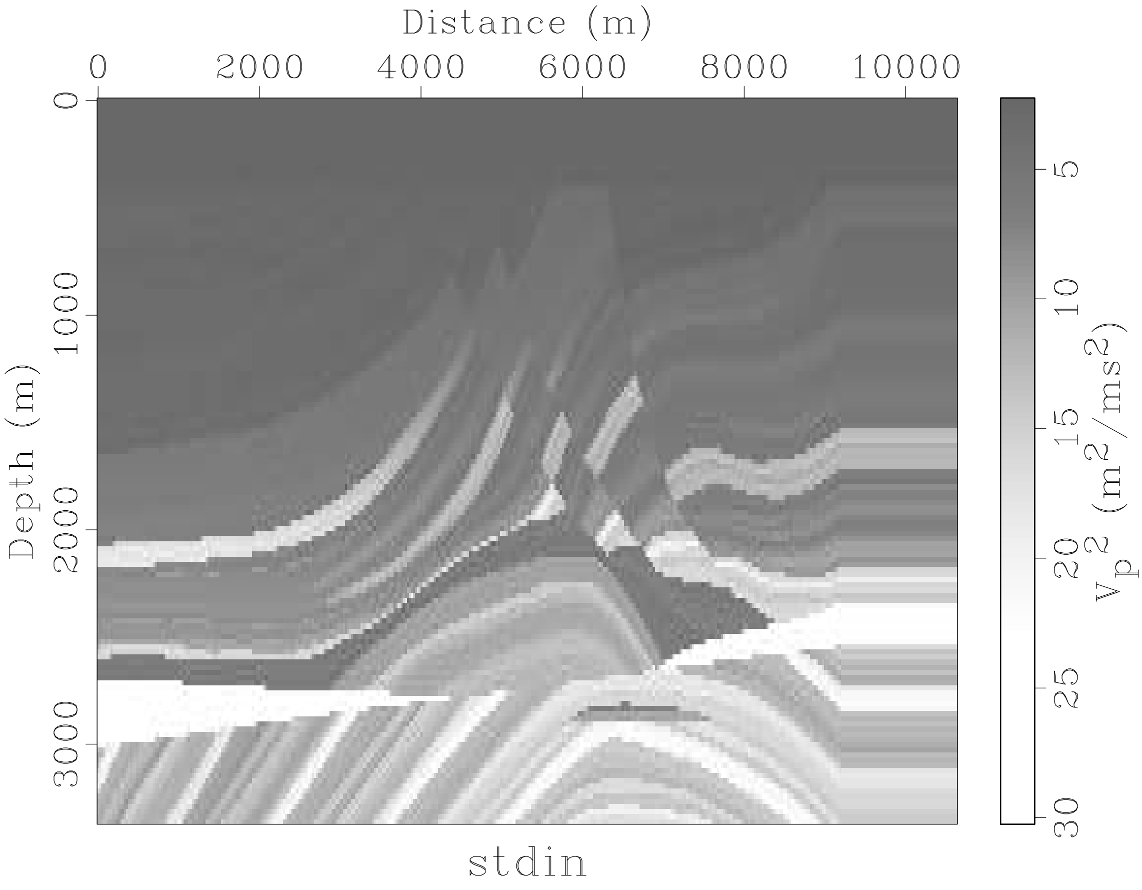


Figure 1: Marmousi model, sampled at 24 m in z and x, layered extension on right to 10632 m. pw/project csq24

The choice of modeling algorithm, the venerable (2,4) scheme, requires at least 5 gridpoints per wavelength for reasonable accuracy over a few seconds' propagation time, so the maximum frequency adequately modeled in the water layer is 12.5 Hz. The source wavelet for these simulations is a $[1, 3.5, 10, 12.5]$ Hz trapezoidal bandpass filter (Figure 2). Free surface boundary conditions are imposed on all boundaries.

The recording time interval is 2 s, short enough that few boundary reflections will be observed, and those only for the larger slownesses.

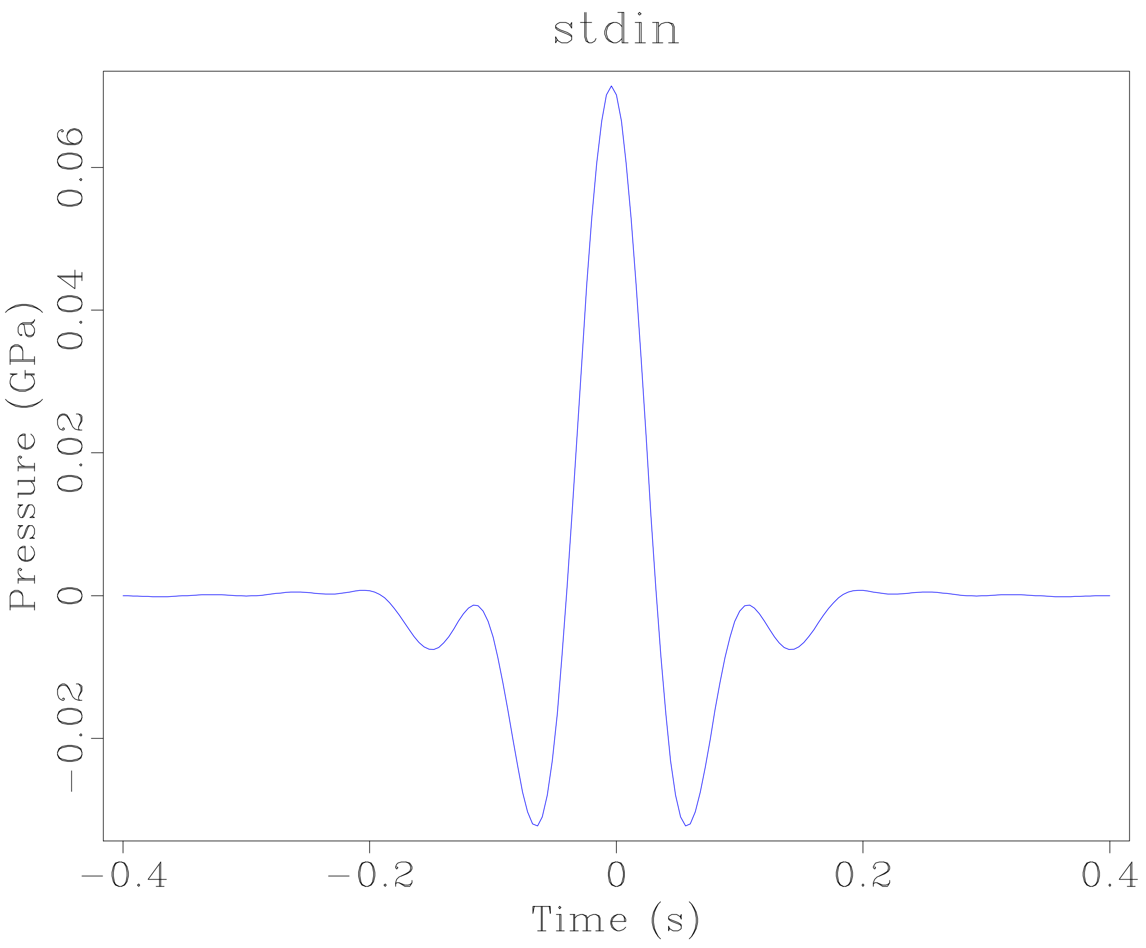


Figure 2: Source pulse: [1.0, 3.5, 10, 12.5] Hz trapezoidal bandpass filter. pw/project wavelet

Source points are placed at all 444 horizontal grid locations (multiples of $dx = 24$ m) at a depth of 12 m. [IWAVE evaluates fields at non-grid points by multilinear interpolation, and inserts sources by multilinear adjoint interpolation.]

The receiver array occupies 241 contiguous horizontal locations spaced at 24 m and beginning at 3000 m, at a depth of 12 m.

Single plane wave at normal incidence

The parameter choice for this example is `op=0, np=1, dp=` whatever.

A movie of the wavefield shows the plane wave starting near the surface (the ghost occurs at a short enough time lag that the effect is visible only as a change of wavelet shape). The sea floor creates a sizeable reflection, of which the first seafloor multiple is within the time range of the simulation - it is however of such low amplitude that it is very difficult to distinguish from other, primary reflections. Figure 3 shows the final frame. View the entire movie as follows:

```
scons movie.rsf
sfgrey clip=2 < movie.rsf|xtpen
```

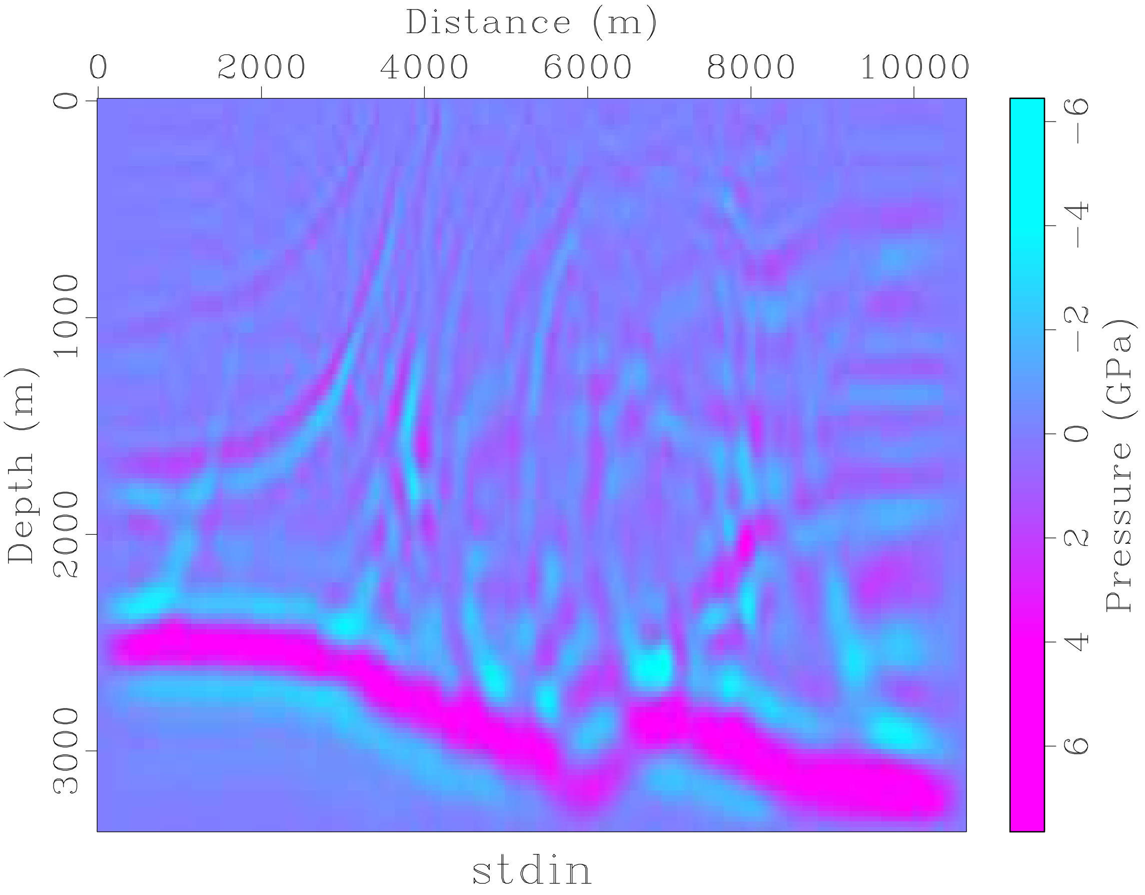


Figure 3: Pressure field simulation at t = 2 s, normal incidence plane wave source. pw/project movie1p

Trace data for this example appears as Figure 4.

Born (linearized) simulation requires definition of a background model and a reflectivity (perturbational model). Figure 5 shows a smoothed version of the Marmousi model, which serves as a macromodel for this example. The difference of the base

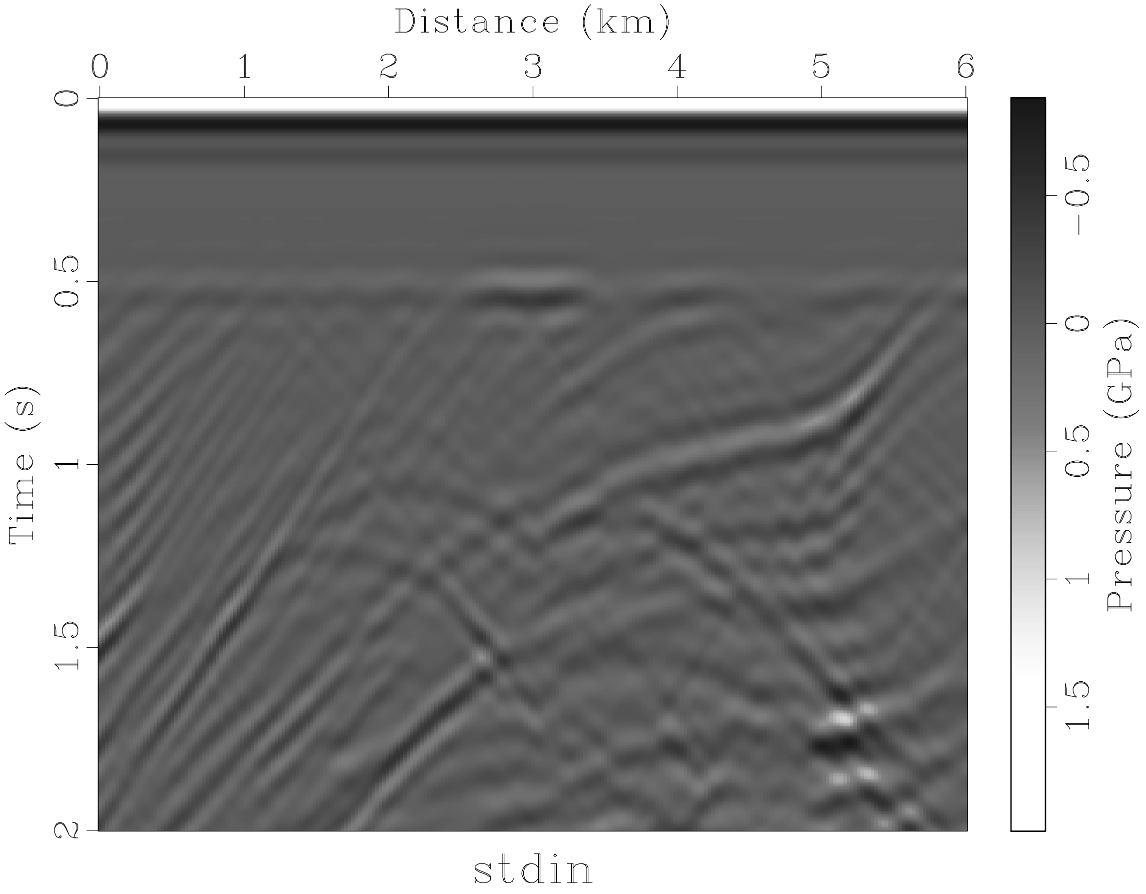


Figure 4: Plane wave gather, normal incidence. pw/project shot1p

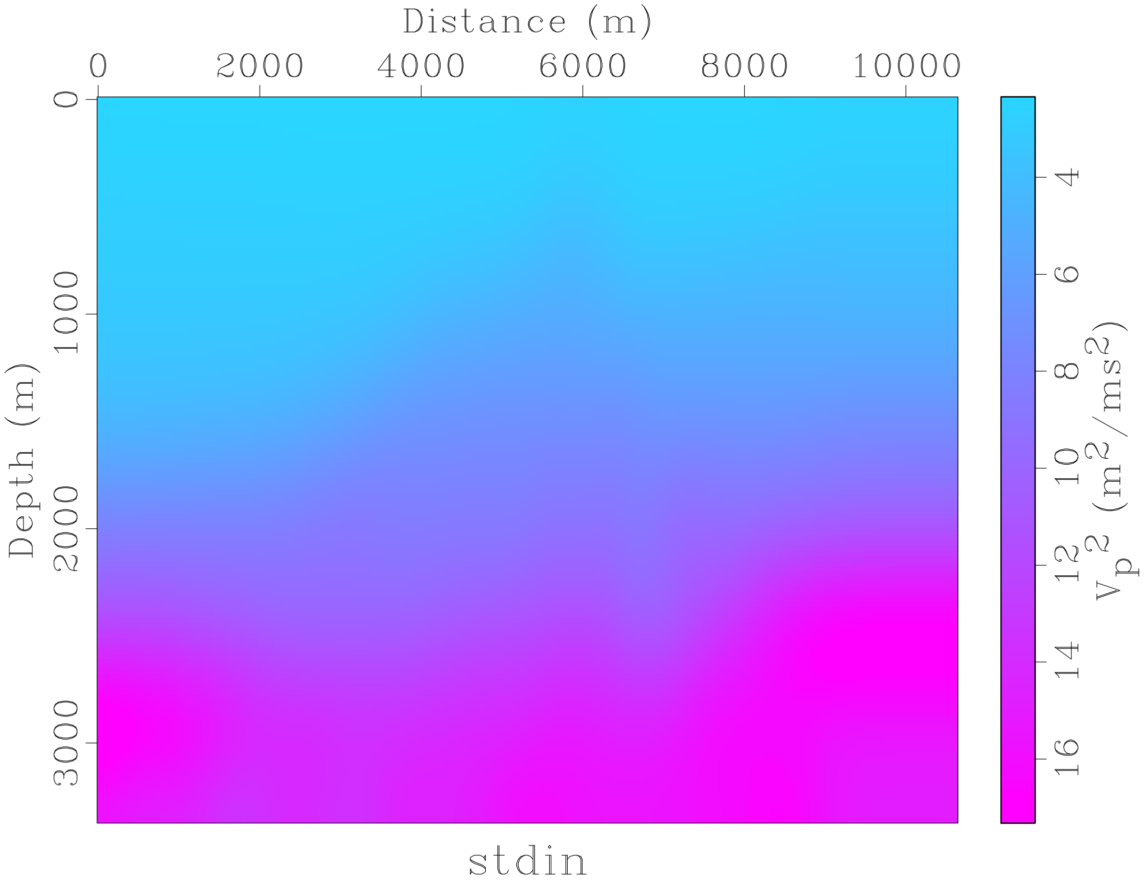


Figure 5: Smoothed Marmousi c-squared model: moving average on 240 m window, iterated 10 times. pw/project csq24big

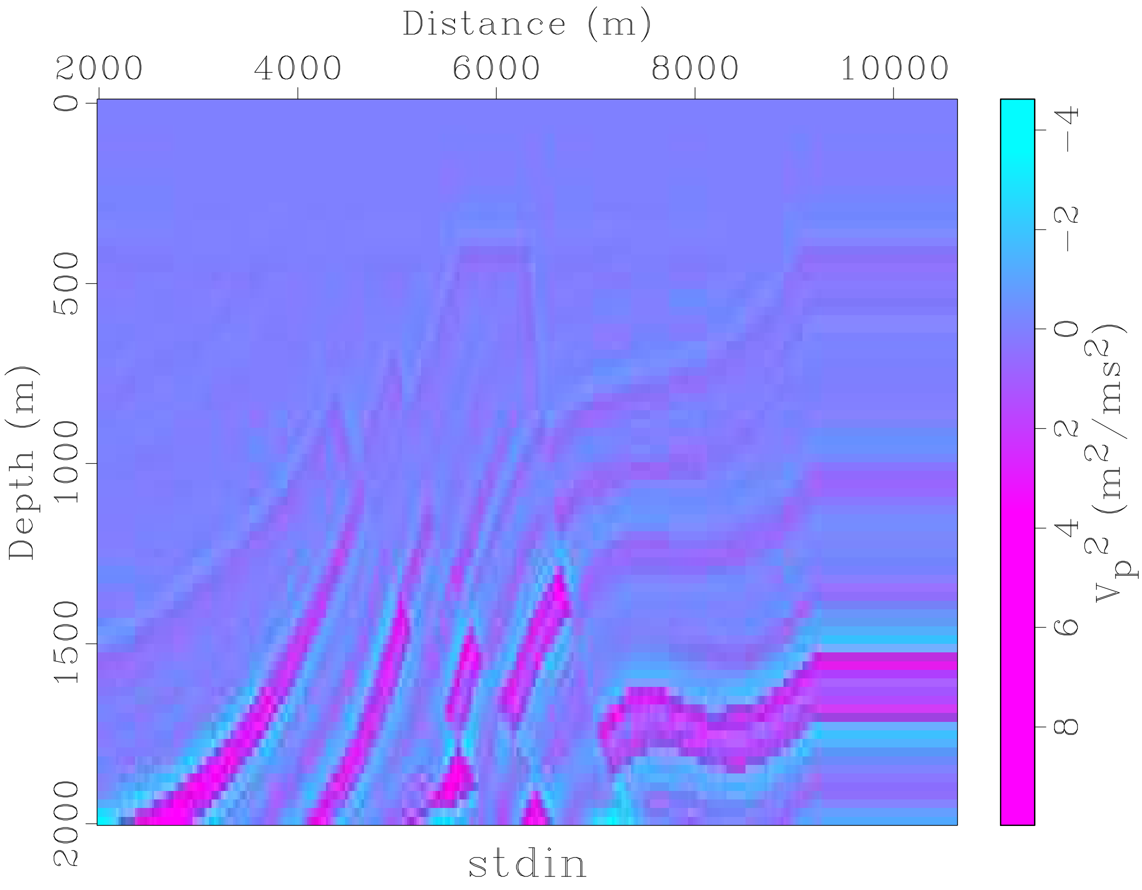


Figure 6: Reflectivity: difference of base model (Figure 1) and moving average on 120 m window, iterated twice. Plot window chosen to match image zone for migration. pw/project dcsq24

model and a smoothing on a shorter length scale gives a suitable, scale-separated reflectivity model (Figure 6).

Trace data for this example appears as Figure 7.

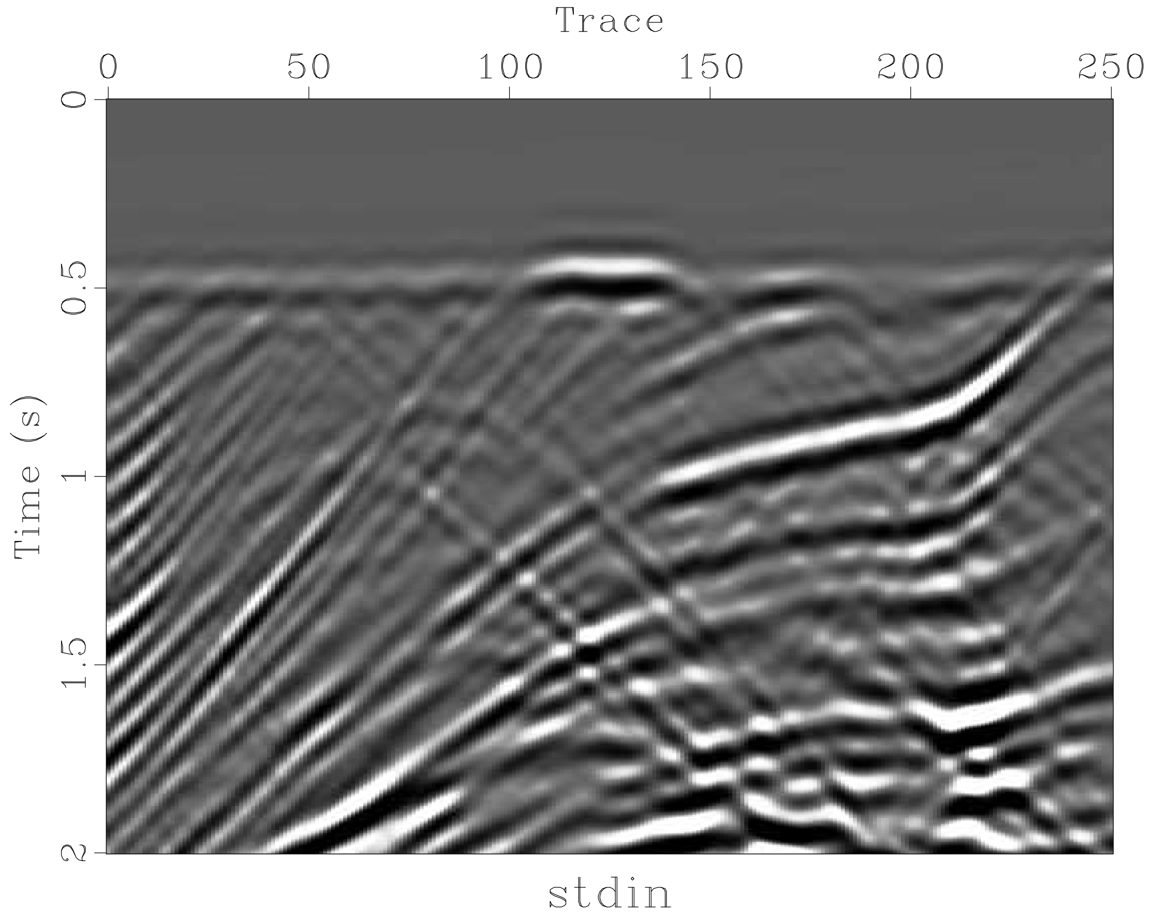


Figure 7: Born plane wave gather, normal incidence. pw/project born1p

Migration of Born data amounts to application of the adjoint linearized modeling operator, a kinematic inverse in the high frequency limit provided that the background model is transparent, as it is in this well-scale-separated example. Compare the migrated image, Figure 8, to the reflectivity model, Figure 6.

The FWI gradient is equivalent to a (reverse time) migration of the simulation residual data, that is, the difference of the modeled and observed data. To illustrate this construction, take for the "observed" data the Marmousi normal incidence gather (Figure 4), and for the current FWI iterate a homogeneous model with $c = 1.5$ m/ms at all locations. The difference is plotted in Figure 9, and the migration = FWI gradient in Figure 10.

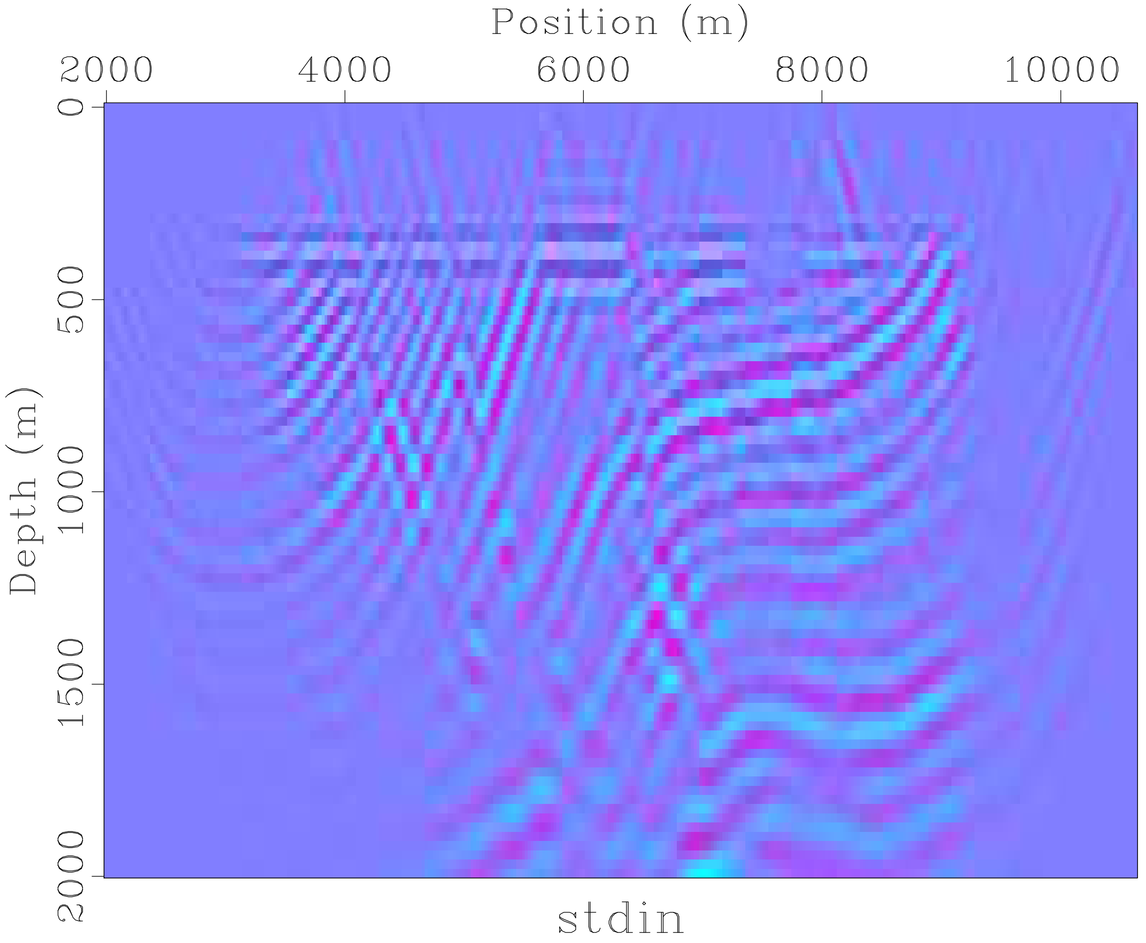


Figure 8: Migration of normal incidence Born data using correct background model.
pw/project mig1p

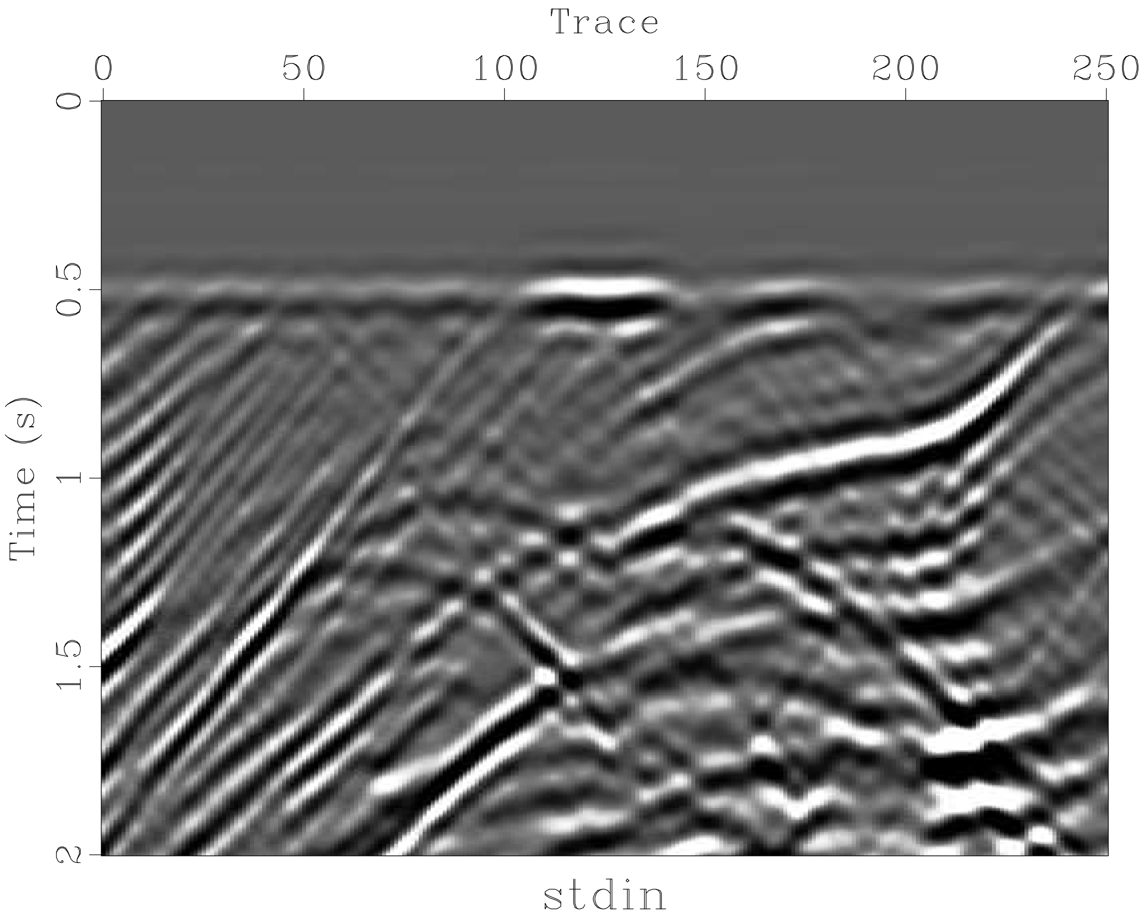


Figure 9: Difference of modeled data (Figure 4) and data from a homogeneous model, which contains only the incident wave. Like Born data (Figure 7), the incident wave is missing (because the difference cancels it); unlike Born data, the difference contains all nonlinear effects (eg. multiple reflections) present in the modeled data. pw/project diff

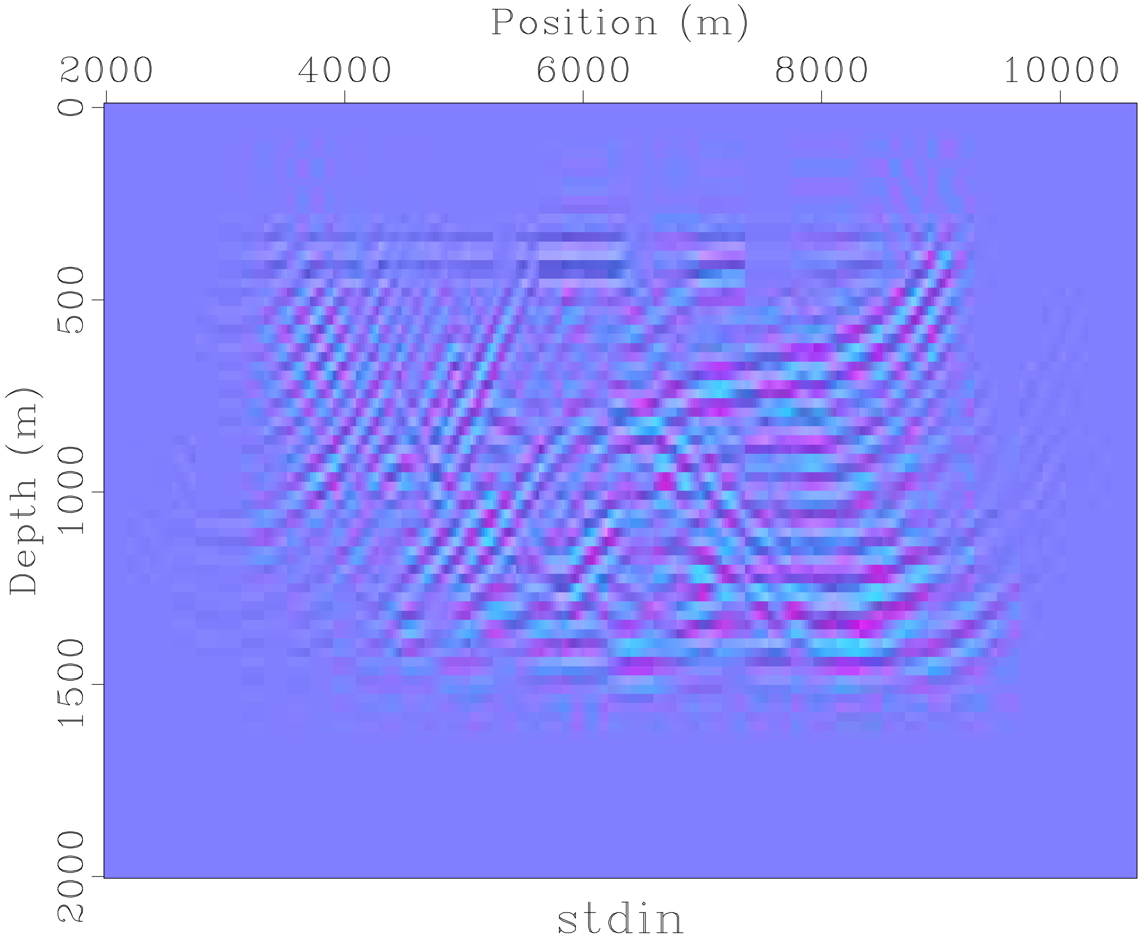


Figure 10: Migration of the difference data of Figure 9, in homogenous background model (c=1.5 m/ms). Note kinematic distortion compared to migration of Born data in consistent background (Figure 8). pw/project migshot1p

Multiple plane waves, oblique incidence

As explained in (Symes, 2014), IWAVE implicitly loops over axes labeled `dim`+1 and greater, in which `dim` is the dimension of the simulation domain (= 2, in the examples presented here). By rigid convention, axes 0,...`dim`-1 are the spatial grid axes, `dim` is the time axis, and all axes with labes `dim` + 1 and greater represent additional axes over which the simulation should loop.

For SEGY, the shot axis is implicitly axis `dim` +1. A change in either `sx` or `sy` increments this axis. Therefore to simulate a line of plane wave "shots", the user merely need construct trace headers containing the requred number of traces for each slowness, and encode the slowness in `sx`. Similarly, the plane wave source should contain one gather for each slowness, with precisely the same slownesses and number of slownesses as the occur in the trace headers.

The `planewave` utility does this job for the user, creating a matching set of plane wave headers and source gathers suitable for IWAVE input. The example plane wave source displayed in Figure 11 shows the plane wave source gathers for 11 evenly spaced slownesses between $p = -0.1$ and 0.1, corresponding to propagation angles of approximately -6 to 6 degrees. Figure 12 shows the computed pressure field at 0.5 s for p=-0.1. It is plotted at approximately 1:1 aspect ratio; clearly the propagation angle in the water later is indeed about 6°. Figure 13 shows the 11 plane wave gathers created by the source array depicted in Figure 11, with the same receiver array as in Figure 4.

Born modeling and migration work the same way. Figure 7 shows the result of linearized modeling with the background of Figure 5 and reflectivitly of Figure 6, with the source gathers of Figure 11. Figure 15 show the result of migrating these 11 gathers. This image is a bit crisper than the normal incidence migration of Figure 8, with migration swings suppressed somewhat by stacking and some of the more steeply dipping features towards the bottom better imaged.

Shot-record extended modeling

Stacking of individual shot images is responsible for artifact suppression in line images. To understand how this happens, exposure of the shot images is useful. The full volume of shot images is also underlies one approach to migration velocity analysis. IWAVE can output this full volume.

In fact, IWAVE interprets the shot record volume as the output of the adjoint linearized map of an *extended* model, in which each shot uses an independent copy of the coefficient fields (c^2, for the constant density acoustics system). The extended modeling concept underlies much recent work on "image domain" inversion. For an

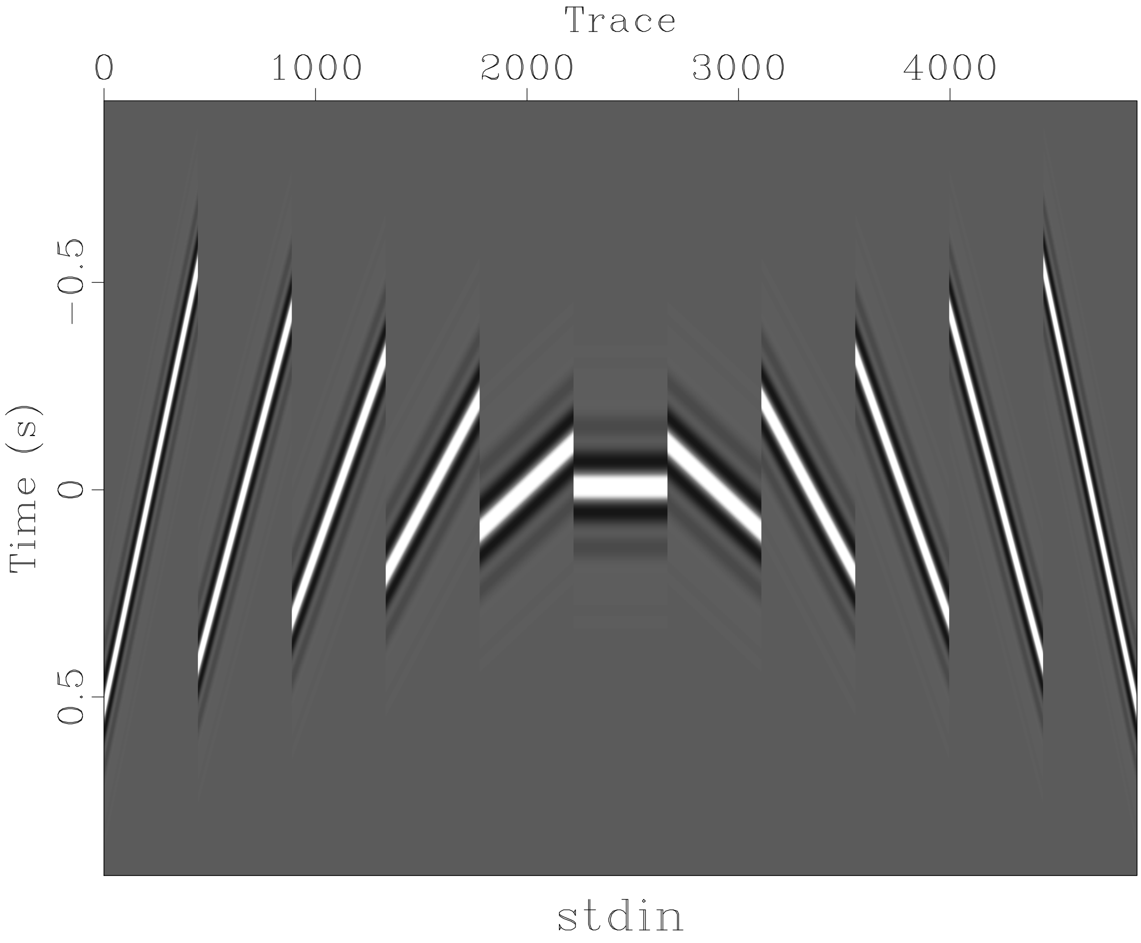


Figure 11: Multiple slowness source gathers: $p \in [-0.1, 0.1]$, $\Delta p = 0.02$ ms/m. pw/project wav11p

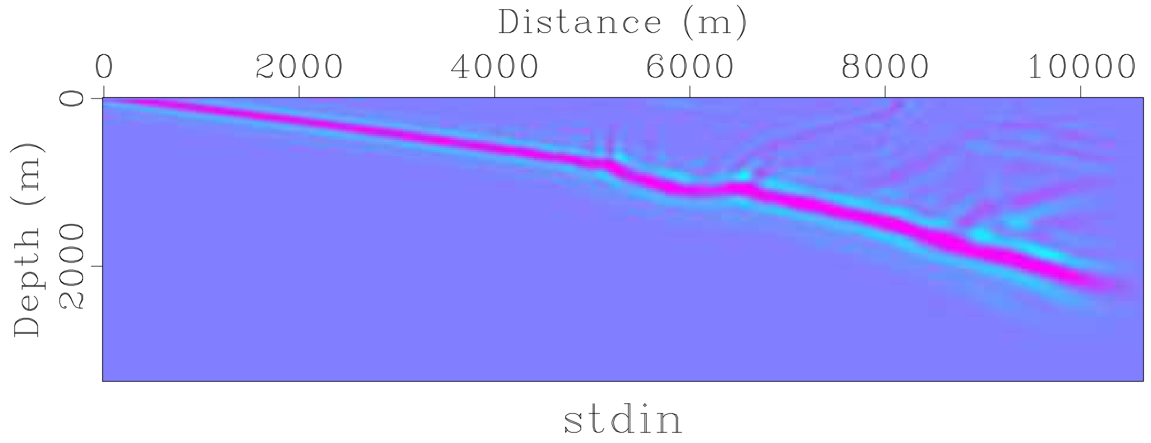


Figure 12: Pressure field response to plane wave source at $p = -0.1$ ms/m, $t = 500$ ms. pw/project movie1pm01

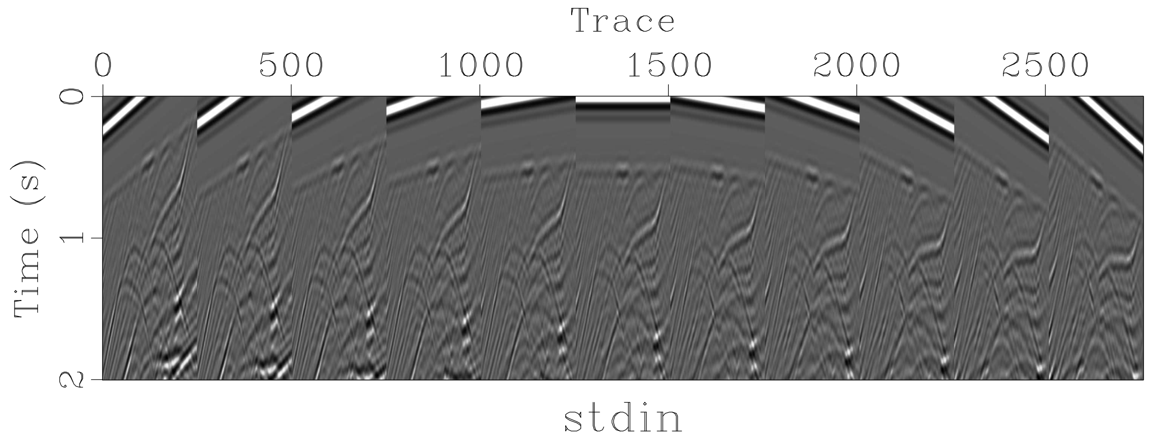


Figure 13: Simulation for 11 plane wave sources in Figure 11. pw/project shot11p

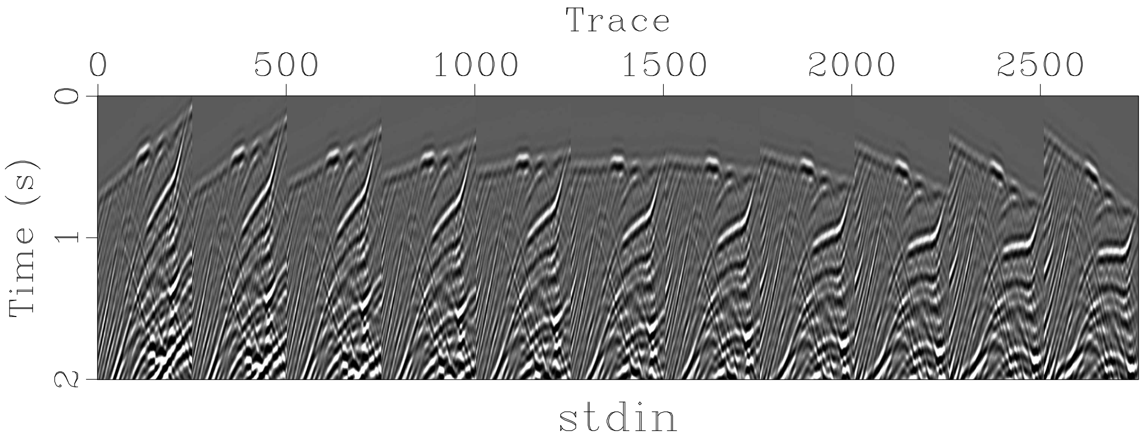


Figure 14: Linearized simulation for 11 plane wave sources, same model parameters as Figure 7. pw/project born11p

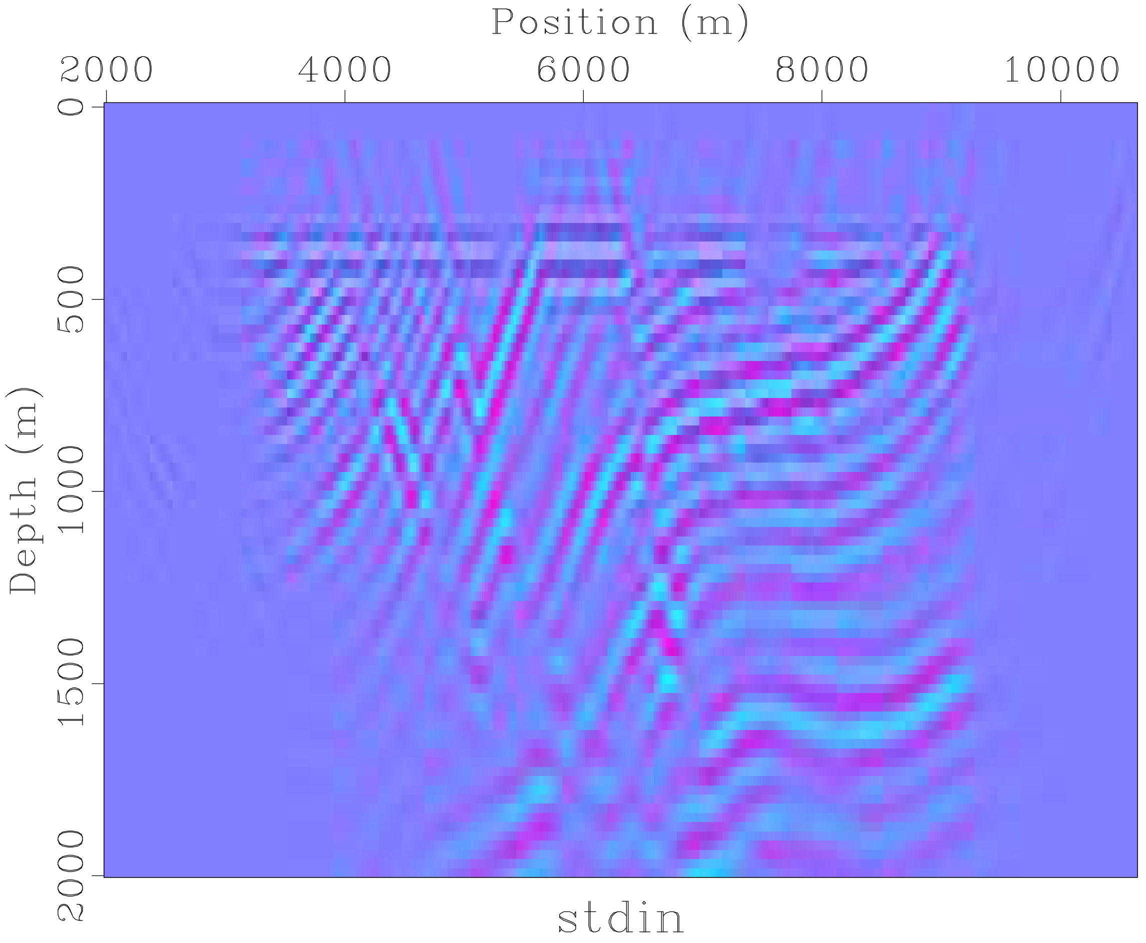


Figure 15: Migration of data in Figure 14. pw/project mig11p

extensive discussion of this idea, see (Symes, 2008).

IWAVE requires the definition of an extended model space, with one "panel" or dim-dimensional model for each shot (in this form of extended modeling - there are others!). Thus as a first step, one must create data files exhibiting the extended structure. The SConstruct file accomplishes this task via use of sfspray - see the rule for creating csq24big_ext. Note that an important part of this construction is the addition of nonstandard keywords dim, gdim, and idx, x = 0, 1, and 3. The functioning of these keywords is explained in (Symes, 2014). In brief, x = 0, 1 signify the two spatial dimensions, and 3 = dim + 1 tells IWAVE to treat the third axis in csq24big_ext as the same as the shot axis implied by the structure of the SU files in the simulation (see discussion above).

Use these extended files as input under the keywords for the background model fields (active, input fields - see the discussion of the fields struct in (Symes, 2014)). For constant density acoustic IWAVE, there is only one such field, with keyword csq. All related fields, eg. the model perturbation inputs for the first derivative (keyword csq_d1 in constant density acoustic IWAVE) or the adjoint first derivative (RTM) outputs (keyword csq_b1), must have the same structure including the same additional keywords. The SConstruct uses sfadd to copy the c-squared file onto the migration output file to initialize it, including all of its header info.

Models such as the shot-record extension are *extended* because the modeling operator or forward map is an extension of the ordinary one. That means that the data output by ordinary modeling is the same as the data output by a "physical" extended model, one in the range of the extension map. In the case of the shot record extension, the extension map simply duplicates (or "sprays", hence the use of sfspray) an ordinary model field as many times as required, and identifies the axis introduced in this way. The adjoint of this spray operation is the stack, which explains the relation between extended migration (adjoint linearized modeling) and ordinary migration: output of the latter is the stack of the output of the former.

An example appears as Figure 16. Panel (shot record migration) 5, or p=-0.02, is displayed in the large front section; this image is comparable to Figure 8. The right panel is an example of a *common image gather*: the horizontal axis is shot index, and the flatness (or lack of same) indicates the kinematic correctness of the background model - that is, the 11 trace. Since this example is an inverse crime, of course the image gather is as flat as possible. Note that it is not perfectly flat or uniform in amplitude, as it would be if the image volume were replaced by the physical extended model (result of spray) which would have generated the data - migration is only the adjoint of linearized modeling, not its inverse.

Replacing the correct c-squared model (Figure 5) by a convex combination with 20% homogeneous water c-squared produces a comparable image volume, shown in Figure 17. Comparison with Figure 16 reveals two obvious differences: the image (front,

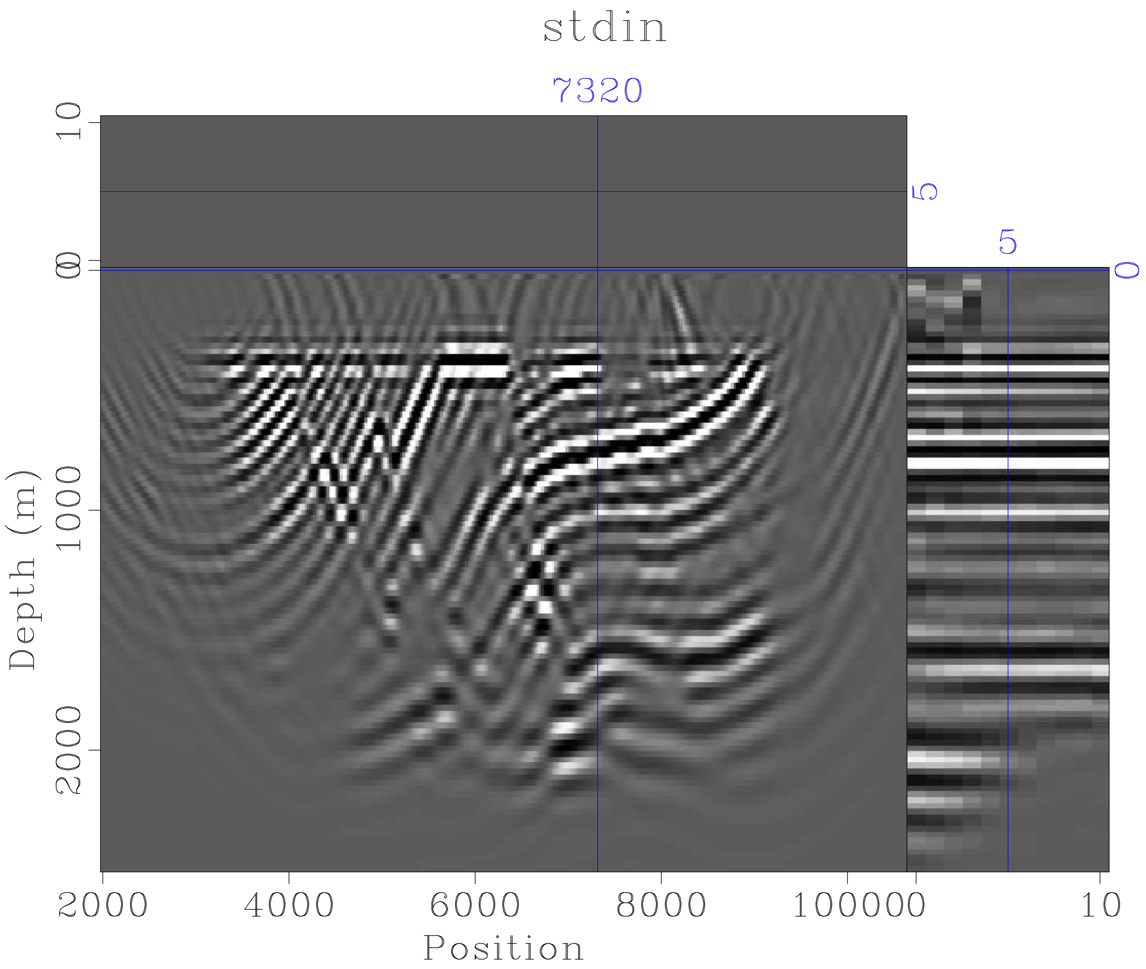


Figure 16: Shot-record extended migration of data in Figure 14 with correct background model, i.e. that used to generate the data. pw/project mig11pext

large rectangle) is misplaced (too shallow), and the image gather is not quite flat. Correct location of the image is not known *a priori*, but it is known that the images for the various values of p should all be the same, at least in the location of events. So that latter provides a means of judging the correctness of the background model. This flatness criterion, and related criteria for other extended models. are the basis of velocity analysis, or as it has come to be known recently, image domain inversion.

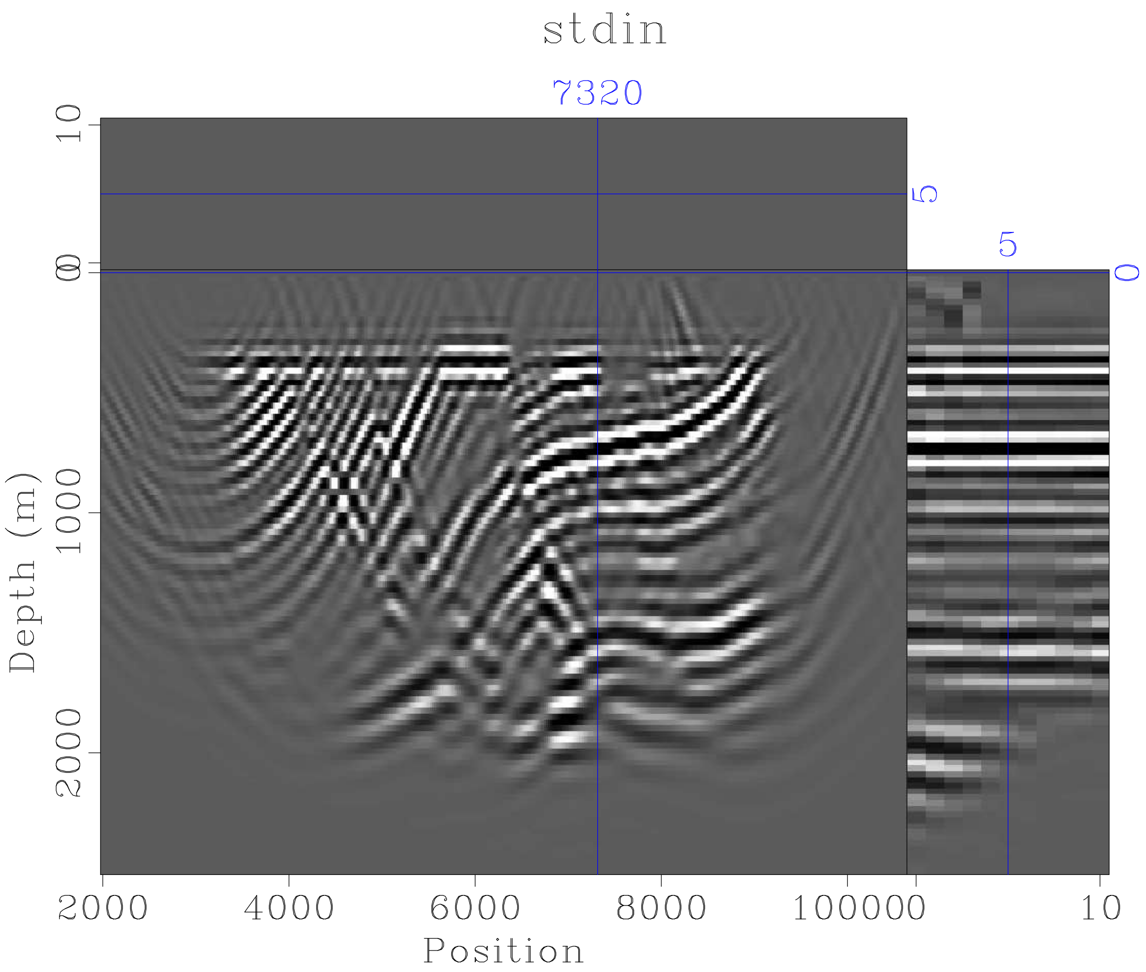


Figure 17: Shot-record extended migration of data in Figure 14 with 80% correct background model, 20% water c-squared. Note non-zero slope of bottom events in image gather (right-hand panel). pw/project mig11p80pctext

CONCLUSION

IWAVE is built to solve equation 1 and similar systems, so the particular choice of right-hand side implicit in plane wave modeling must be the result of an external (to an IWAVE modeling driver) utility. The `planewave` utility generates a matching pair of SEGY trace files: (i) a trace header file, serves as a prototype for the output data of a simulation or the input data of a migration, and (ii) ah source gather file, also SEGY traces, encoding the plane wave source (or sources, for a multiple plane wave simulation). Coupled with IWAVE driver code (such as `acd`), the `planewave` utility creates a basic tool for plane wave simulation and migration.

ACKNOWLEDGEMENTS

The IWAVE project owes a great deal to several open source seismic software packages (Seismic Un*x, SEPlib, Madagascar), debts which we gratefully acknowledge. This work was supported in part by the sponsors of The Rice Inversion Project.

REFERENCES

Achenbach, J., 1973, Wave propagation in elastic solids: North Holland.

Aki, K., and P. Richards, 1980, Quantitative Seismology: Theory and Methods: Freeman.

Carrion, P., J. Kuo, and P. Stoffa, 1984, Inversion method in the slant-stack domain using amplitudes of reflection arrivals: Geophysical Prospecting, **32**, 379–391.

Courant, R., and D. Hilbert, 1962, Methods of mathematical physics: Wiley-Interscience, **II**.

Diebold, J., and P. Stoffa, 1981, The traveltime equation, tau-p mapping, and inversion of common midpoint data: Geophysics, **46**, 238–254.

Soubaras, R., and B. Gratacos, 2007, Velocity model building by semblance maximization of modulated-shot gathers: Geophysics, **72**, U67.

Stoffa, P. L., P. Buhl, J. B. Diebold, and F. Wenzel, 1981, Direct mapping of seismic data to the domain of intercept time and ray parameter - a plane-wave decomposition: Geophysics, **46**, 255–267.

Sun, D., and W. Symes, 2009, A nonlinear differential semblance strategy for waveform inversion: Experiments in layered media: 80th Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, 2526–2530.

———, 2012, Waveform inversion via nonlinear differential semblance optimization: 82nd Annual International Meeting, Expanded Abstracts, Society of Exploration Geophysicists, SI3.3.

Symes, W., 2008, Migration velocity analysis and waveform inversion: Geophysical Prospecting, **56**, 765–790.

———, 2014, IWAVE structure and basic use cases, *in* TRIP 2014 Annual Report: The Rice Inversion Project. (available summer 2015).

Treitel, S., P. Gutowski, and D. Wagner, 1982, Plane-wave decomposition of seismograms: Geophysics, **47**, 1375–1401.

Acoustic Staggered Grid Modeling in IWAVE

William. W. Symes [*]

ABSTRACT

IWAVE is a framework for time-domain regular grid finite difference and finite element methods. The IWAVE package includes source code for infrastructure component, and implementations of several wave physics modeling categories. This paper presents two sets of examples using IWAVE acoustic staggered grid modeling. The first set illustrates the effectiveness of a simple version of Perfectly Matched Layer absorbing boundary conditions. The second set reproduce illustrations from a recent paper on error propagation for heterogeneous medium simulation using finite differences, and demostrate the interface error effect which renders all FD methods effectively first-order accurate. The source code for these examples is packaged with the paper source, and supports the user in duplicating the results presented here and using IWAVE in other settings.

INTRODUCTION

Domain-specific simulation such as seismic modeling begs for software re-use via modular design. All applications of this type have the same structure: static fields are initialized, dynamic fields updated, output extracted. A modular approach to code architecture is implicit in this structure, and further specialization leads to even more opportunity for code re-use via modular design.

IWAVE is open source software for finite difference or finite element time-domain simulation on regular rectangular grids, written exclusively in the C/C++. IWAVE is built around a core framework: that is, a collection of separate software packages which together provide essential services upon which applications may be built. These service components completely define the interfaces to which additional code must be written to formulate a complete application. The core framework defines

- parameter-driven job control;
- grid generation and memory allocation in 1D, 2D, and 3D space;
- serial, loop-parallel, and task-parallel execution models, scaling to thousands of threads;

- arbitrary source and receiver locations, and flexible source specification including simultaneous source modeling (random, plane-wave,...)
- standard input and output data formats (SEGY, RSF)
- predefined support for linearized (Born) modeling and adjoint linearized (RTM) modeling, both first and second order;
- uniform interface to optimization and linear algebra for creation of inversion applications via the Rice Vector Library ("RVL") (Padula et al., 2009; Symes et al., 2011).

Symes (2014) describes the desgn principles underlying the IWAVE core framework, and illustrates the construction of a complete acoustic modeling application using centered finite differences for the second order acoustic constant denstity wave equation.

The primary purpose of this short paper is to illustrate synthetic seismogram generation using another finite difference scheme implemented in IWAVE, the staggered grid approximation to variable-density velocity-pressure acoustodynamics (Virieux, 1984). Exactly the same framework supports this application as was described in Symes (2014); as explained there, only two data structures and six principal functions need be defined to implement this (or any) finite difference method in IWAVE.

The examples illustrate two aspects of finite difference modeling. The IWAVE staggered grid implementation includes a version of PML absorbing boundary conditions (Hu et al., 2007), permitting accurate finite grid approximation of wave propagation in a full- or half-space. The first set of examples demonstrates the effectiveness of these very simple PML conditions. The second set reproduce the examples presented in Symes and Vdovina (2009), and illustrate a fundamental limitation in the use of straightforward finite-difference methods for modeling waves in heterogeneous media.

IWAVE was used in a quality control role in the SEAM Phase I project - see Fehler and Keliher (2011) for an account, including discussion of the many difficulties of large scale numerical simulation of seismograms.

The internal details of IWAVE are not discussed here, except insofar as is necessary to explain the use of the main commands. As mentioned above, Symes (2014) overviews the design of IWAVE and the main features of its internal structure, and defines the elements necessary to compile a new IWAVE application. Symes et al. (2011) briefly describe the IWAVE/RVL mechanisms for coupling modeling with optimization packages to produce inversion applications.

The paper begins with a brief review of the system of partial differential equations solved (approximately) by IWAVE's acoustic application, and the choice of finite difference method. The next section evaluates the effectiveness of the PML absorbing boundary conditions included in the IWAVE staggered grid acoustic application. The following section presents the examples of Symes and Vdovina (2009), along with some additional examples based on the same distribution of mechanical parameters which shed light on the impact of finite difference order on solution accuracy. Instructions follow for recreating these examples, and for using them as starting points for further modeling exercises. The paper ends with a brief discussion of the prospects for improvements in performance and accuracy in FD technology, and the evolutionary advantages flowing from the modular, or object, orientation of IWAVE. Two appendices describe the job parameters used in the examples, and download and install instructions.

ACOUSTODYNAMICS

The pressure-velocity form of acoustodynamics consists of two coupled first-order partial differential equations:

$$\begin{aligned} \rho \frac{\partial \mathbf{v}}{\partial t} &= -\nabla p & (1) \\ \frac{1}{\kappa}\frac{\partial p}{\partial t} &= -\nabla \cdot \mathbf{v} + g & (2) \end{aligned}$$

In these equations, $p(\mathbf{x},t)$ is the pressure (excess, relative to an ambient equilibrium pressure), $\mathbf{v}(\mathbf{x},t)$ is the particle velocity, $\rho(\mathbf{x})$ and $\kappa(\mathbf{x})$ are the density and bulk modulus respectively. Bold-faced symbols denote vectors; the above formulation applies in 1, 2, or 3D.

The inhomogeneous term g represents externally supplied energy (a "source"), via a defect in the acoustic constitutive relation. A typical example is the *isotropic point source*

$$g(\mathbf{x},t) = w(t)\delta(\mathbf{x}-\mathbf{x}_s)$$

at source location $\mathbf{x}_s$.

Virieux (1984) introduced finite difference methods based on this formulation of acoustodynamics to the active source seismic community. Virieux (1986) extended the technique to elastodynamics, and Levander (1988) demonstrated the use of higher (than second) order difference formulas and the consequent improvement in dispersion error. Many further developments are described in the review paper Moczo et al. (2006). IWAVE's acoustic application uses the principles introduced by these authors to offer a suite of finite difference schemes, all second order in time and of various orders of accuracy in space.

The bulk modulus and buoyancy (reciprocal density) are the natural parameters whose grid samplings appear in the difference formulae. These are the parameters displayed in the figures below, rather than, say, velocity and density, which might seem more natural.

PML EFFECTIVENESS

The IWAVE acoustic staggered grid scheme implements the Perfectly Matched Layer (PML) approach to absorbing boundary conditions, in one of the simpler of its many guises (a split field approach - (Hu et al., 2007)). After some manipulation, the acoustic PML system for the physical velocity $\mathbf{v}$ and an artificial vector pressure $\mathbf{p}$ takes the form

$$\begin{aligned} \rho\left(\frac{\partial v_k}{\partial t} + \eta_k(x_k)v_k\right) &= -\frac{\partial p_k}{\partial x_k}, \\ \frac{1}{\kappa}\left(\frac{\partial p_k}{\partial t} + \eta_k(x_k)p_k\right) &= -\nabla\cdot\mathbf{v} + g \end{aligned} \tag{3}$$

in which the kth component of the attenuation profile vector η depends only on x_k, and can be stored as a collection of 1D objects. Ordinary acoustic wave propagation takes place where $\eta = \mathbf{0}$, and if the components of the vector pressure $\mathbf{p}$ are all the same in this zone, then they remain the same there, and any one of them may be regarded as the same as the physical pressure field. Outside of the physical domain, where waves are to be attenuated, η should ibe positive; at the boundary of the physical domain, it should vanish to positve order. We elected to make η cubic in distance to the boundary: for a PML layer of width $L_{k,r}$, beginning at $x_k = x_{k,r}$ along the kth coordinate axes,

$$\eta(x_k) = \eta_0\left(\frac{x_k - x_{k,r}}{L_{k,r}}\right)^3$$

Thus there are four PML boundary layer thicknesses in 2D, six in 3D, one for each side of the simulation cube. The IWAVE convention imposes pressure-free boundary conditions on the exterior boundary of the PML domain. Thus $L = 0$ signifies a free surface boundary face. Any face of the boundary may be assigned a zero-pressure condition ($L = 0$) or a PML zone of any width ($L > 0$).

Many implementations of PML, especially for elasticity, confine the extra PML fields (in this case, the extra pressure variables) to explicitly constructed zones around the boundary, and use the standard physical system in the domain interior. We judged that for acoustics little would be lost in either memory or efficiency, and much code bloat avoided, if we were to solve the system (3) in the entire domain.

Considerable experience and some theory (Hu et al., 2007; Moczo et al., 2006) suggest that the system 3 will effectively absorb waves that impinge on the boundary, emulating free space in the exterior of the domain, if the PML zones outside the physical domain in which η are roughly a half-wavelength wide, and $\eta_0 = 0$.

A simple 2D example illustrates the performance of this type of PML. The physical domain is a 1.8 x 7.6 km; the same domain is used in the experiments reported in the next section. A point source is placed at z=40 m, x = 3.3 km, with a Gaussian derivative time dependence with peak amplitude at about 5 Hz, and signifcant energy at 3 Hz but little below. The acoustic velocity is 1.5 km/s throughout the domain, so the effective maximum wavelength is roughly 500 m. The density is also constant, at 1 g/cm^3. A snapshot of the wavefield at 1.2 s after source onsiet (Figure 1), before the wave has reached the boundary of the domain, shows the expected circular wavefront. At 4.0 s, a simulation with zero-pressure boundary conditions on all sides of the physical domain produces the expected reflections, Figure 2. With PML zones of 250 m on the bottom and sides of the domain, so that only the top is a zero-pressure surface, and $\eta_0 = 1$, the wave and its free-surfacec ghost both appear to leave the domain (Figure 3, plotted on the same grey scale). The maximum amplitude visible in Figure 2 is roughly 7.1×10^{-2}, whereas the maximum amplitude in Figure 3 is 7.0×10^{-5}. The actual reflection coefficient is likely less than 10^{-3}, as the 2D free space field does not have a lacuna behind the wavefront, but decays smoothly, so the low end of the wavelet spectrum remains.

It is not possible to decrease the PML layer thickness much beyond the nominal longest half-wavelength and enjoy such small reflections. Figure 4 shows the field at 4.0 s with PML zones of width 100 m on bottom and sides, and an apparently optimal choice of η_0. The maximum amplitude is 2.3×10^{-4}, and a reflected wave is clearly visible at the same grey scale.

ALL FD SCHEMES ARE FIRST ORDER IN HETEROGENEOUS MEDIA

The bulk modulus and buoyancy models depicted in Figures 5 and 6 embed an anticline or dome in an otherwise undisturbed package of layers. These figures display sampled versions of the models with $\Delta x = \Delta z = 5$ m; the model fields are actually given analytically, and can be sampled at any spatial rate. The IWAVE utility *sfstandardmodel* (in the Madagascar `bin` directory) builds this example and a number of others that can be sampled arbitrarily for grid refinement studies. See its self-doc for usage instructions.

Symes and Vdovina (2009) use the model depicted in Figures 5 and 6 to illustrate the *interface error* phenomenon: the tendency, first reported by Brown (1984), of all finite difference schemes for wave propagation to exhibit first order error, regardless of formal order, for models with material parameter discontinuities. Figure 7 exhibits a shot gather, computed with a (2,4) (= 2nd order in time, 4th order in space) staggered grid scheme, $\Delta x = \Delta z = 5$ m (more than 20 gridpoints per wavelength at the wavelength corresponding to the highest frequency, 12 Hz, with significant energy, and the smallest $v_p = 1.5$ km/s) and an appropriate near-optimal time step, acquisition geometry as described in caption. The same gather computed at different spatial sample rates seem identical, at first glance, however in fact the sample rate has a considerable effect. Figures 8 and 9 compare traces computed from models sampled at four different spatial rates (20 m to 2.5 m), with proportional time steps.

The scheme used is formally 2nd order convergent like the original 2nd order scheme suggested by Virieux (1984), but has better dispersion suppression due to the use of 4th order spatial derivative approximation. Nonetheless, the figures clearly show the first order error, in the form of a grid-dependent time shift, predicted by Brown (1984).

Generally, even higher order approximation of spatial derivatives yields less dispersive propagation error, which dominates the finite difference error for smoothly varying material models. For discontinuous models, the dispersive component of error is still improved by use of a higher order spatial derivative approximation, but the first order interface error eventually dominates as the grids are refined. Figure 10 shows the same shot gather as displayed earlier, with the same spatial and temporal sampling and acquisition geometry, but computed via the (2,8) (8th order in space) scheme. The two gather figures are difficult to disinguish. The trace details (Figures 11, 12) show clearly that while the coarse grid simulation is more accurate than the (2,4) result, but the convergence rate stalls out to 1st order as the grid is refined, and for fine grids the (2,4) and (2,8) schemes produce very similar results: dispersion error has been suppressed, and what remains is due to the presence of model discontinuities.

See Symes and Vdovina (2009) for more examples, analysis, and discussion, also Fehler and Keliher (2011) for an account of consequences for quality control in large-scale simulation.

Note that the finest (2.5 m) grid consists of roughly 10 million gridpoints. Consequently the modeling runs collectively take a considerable time, from a minutes to a substantial fraction of an hour depending on platform, on a single thread. This example is computationally large enough that parallelism via domain decomposition is worthwhile. IWAVE is designed from the ground up to support parallel computation; a companion report will demonstrate parallel use of IWAVE.

CREATING THE EXAMPLES - RUNNING IWAVE APPLICATIONS

IWAVE builds with SConstruct (`http://www.scons.org`), either as an independent package or as part of Madagascar (Fomel, 2009). See the Madagascar web site

`http://www.ahay.org/wiki/Main_Page`

for download and install instructions. Source for IWAVE and other TRIP software reside in the `trip` subdirectory of the top-level Madagascar source directory. A `README` file describes how to install TRIP software independently of the rest of Madagascar, which is useful to configuring TRIP differently from other parts of the package (for example, with MPI support).

The IWAVE acoustic staggered grid modeling command is `sfasg` for the Madagascar build, stored in the Madagascar `$RSFROOT/bin` directory, or

```
$RSFSRC/trip/iwave/asg/main/asg.x
```

for the standalone build. All IWAVE commands self-document: entering the command path prints a usage statement to the terminal, including descriptions of all parameters.

The paper you are currently reading follows the reproducible research pattern described on the Madagascar web site, using Madagascar reproducible research tools. You can find the LaTeX source in the subdirectory `book/trip/asg` of the Madagascar source directory, and the script for building the data in

```
$RSFSRC/book/trip/asg/project/SConstruct
```

This script, together with the self-doc for the acoustic staggered grid command and the remarks in the remainder of this section, should enable you to build your own examples after the pattern used in this project.

IWAVE applications currently expect model data files in the RSF format of Madagascar (Fomel, 2009). Data from other sources will need conversion to this format. An RSF data set consists of two files, an ascii header (grid metadata) file and a flat binary data file. The data set is referenced by the header file name; one of the parameters listed in the header file is the pathname of the binary data file, with key `in`. The header file is small and easily created by hand with an editor, if necessary. Madagascar commands add processing history information to header files, and modify their parameters. By convention, the last value of a parameter (`key=value` pair) appearing in the file is the current value. Many archival data formats make the grid sample values available as a flat binary file - this is true for instance of the gridded models output by GOCAD (`http://www.gocad.org`), for which the `vo` files contain virtually the same information as (so may easily be translated to) RSF header files in ascii form, and the `vodat` files are flat binary files which may be used unaltered as RSF binary files.

IWAVE uses two extensions of the Madagascar RSF standard. The first is the optional inclusion of the `dim` and `gdim` keywords. These permit IWAVE applications to treat an RSF file image as defining a `gdim` dimensional data hypercube divided into `dim` dimensional slices. The second is the axis identification keyword set, `id1`, `id2`, etc.: these supply information on the physical meaning of various axes. For an IWAVE `dim` space-dimensional modeling problem, axes labeled `id1`,...,`id[dim-1]` are the spatial grid axes. If `gdim ¿ dim`, then `id[dim]` labels the time axis, and `id[n]`, `n > dim`, axes other than those of space-time. The IWAVE structure paper (Symes, 2014) explains the use of the additional keywords in more detail.

An example of this construction appears in the script that builds the PML examples above, which are actually frames of movies. The output of the 2D simulations are 3D RSF files (`gdim=3, dim=2`) with `id3=2`, that is, the third axis is treated as time. Madagascar applications ignore these keywords: in particular, you can view the 3D RSF simulation output as a movie using `sfgrey` and `xtpen` as usual. The presence of the additional keywords is necessary in order for IWAVE to correctly interpret the data geometry.

This example illustrates another important feature of IWAVE applications: any output data files must exist prior to execution - their data samples are overwritten. The `SConstruct` for this project uses `sfmakevel` to create the movie output files and `sfput` to add the IWAVE-specific keywords to the headers, before invoking the IWAVE command.

By IWAVE convention, the dimension of the problem is that of the primary model grid. In the acoustic staggered grid application, the primary model grid is that associated with the bulk modulus data. This grid is also the primary grid of the simulation: that is, the space steps used in the finite difference method are precisely those of the bulk modulus data. Thus the choice of simulation grid is made externally to IWAVE.

The IWAVE acoustic application uses specific internal scales - m/ms for velocity, g/cm^3 for density, and corresponding units for other parameters. To ensure that data in other (metric) units are properly scaled during i/o, the RSF header file may specify a value for the `scale` key, equal to the power of 10 by which the data should be multiplied on being read into the application, to convert to the internal scale. For example, if velocities are given in m/s, the header file should include the line `scale = -3`. In forthcoming releases, this device will be deprecated in favor of explicit unit specifications.

The current release is configured to use Seismic Unix ("SU") (SEGY without reel header) format for trace data i/o. Units of length and time are m and ms respectively, consistent with other internal unit choices. Two peculiarities of which the user should be aware: (i) *receiver* coordinates (`gx`, `gy`, and `gelev` keywords) *always* specify trace location , that is, the location at which values are sampled in space-time, and (ii) on input, traces are regarded as point sources, so that each trace multiplies a discrete spatial delta (hence values are scaled by the reciprocal grid cell volume). Both of these design choices stem from the migration (adjoint modeling) and inversion uses of IWAVE, discussed for example in (Symes et al., 2011; Symes, 2014).

Source traces must be modified to conform to this rubric. The `sftowed_array` application relieves the user of the necessity to manually adjust the headers of an SU file containing source traces. It accepts three arguments: (i) an input source source file containing `gx`, `gy`, and `gelev` values representing source trace location *relative* to a source center location - the source coordinates of source traces are ignored; (ii) a data file whose sx, sy, and selev values are the source center locations to be used - its receiver coordinates are ignored, and (iii) an output file (name), to which output source traces will be written, each with *source* coordinates equal to those of a data trace, and *receiver* coordinates equal to the sums of the source trace *receiver* coordinates and the data trace *source* coordinates. The result is a collection of source coordinate gathers with the same source coordinates as the data file, but within each gather the same receiver coordinates *relative* to the source coordinates as the source file. Thus the source array is translated to each of the source centers specified in the data file. Because the source file may contain arbitrarily many traces with arbitrary relative locations, any source radiation pattern may be approximated (Santosa and Symes, 2000).

The example scripts in the project subdirectory use Madagascar commands to create these prototype trace files.

One of IWAVE's design criteria is that acquisition geometry parameters should have no *a priori* relation to the computational grid geometry: source and receiver locations may be specified anywhere in Euclidean space.

DISCUSSION AND CONCLUSION

The rather large and only slowly disappearing error revealed by the examples from Symes and Vdovina (2009) suggests strong limits for the accuracy of regular grid finite difference methods. Finite element methods suffer from the same limitations: accurate solution of acoustodynamic or elastodynamic problems appears to demand interface-fitted meshed (Cohen, 2002), with the attendant increase in code and computational complexity.

The situation may not be so bleak, however. For one special case, namely constant density acoustics, Terentyev and Symes (2009) show that a regular grid finite difference method, derived from a regular grid Galerkin finite element method, has accuracy properties one would expect in homogeneous media (second order convergence, reduction of grid dispersion through higher order space differencing) even for discontinuous models: the interface error effect is attenuated. This type of result actually goes quite far back in computational geophysics (see for example Muir et al. (1992)), though theoretical support has been slower in coming.

Pure regular grid methods cannot take advantage of changes in average velocity across the model, and concommitant changes in wavelength. Coupling of local regular grids is possible, however, and can yield substantial computational efficiency through grid coarsening in higher velocity zones - see Moczo et al. (2006). IWAVE already accommodates multiple grids (in domain decomposition parallelism), and extension to incommensurable multiple grids would be a significant change, but in principle straightforward. The use of logically rectangular but geometrically irregular ("stretched") grids is completely straightforward, on the other hand.

These and other extensions, both past and future, are eased by the reusability designed into the IWAVE core framework. This design has produced reasonably well-performing and easy-to-use applications, and has proven extensible to new models and schemes. Moreover, as explained by Symes et al. (2011), the object-oriented design of IWAVE dovetails with similarly designed optimization software to support the construction of waveform inversion software. The inversion applications resulting from this marriage inherit the features of IWAVE - parallel execution, high-order stencils, efficient boundary conditions, simple job control - without requiring that these aspects be reworked in the code extensions.

ACKNOWLEDGEMENTS

IWAVE has been a team effort: the original design of the core framework is due to Igor Terentyev, and Tanya Vdovina, Dong Sun, Marco Enriquez, Xin Wang, Yin Huang, Mario Bencomo, and Muhong Zhou have each made major contributions. Development of IWAVE was supported by the SEG Advanced Modeling (SEAM) project, by the National Science Foundation under awards 0620821 and 0714193, and by the sponsors of The Rice Inversion Project. The IWAVE project owes a great deal to several open source seismic software packages (Seismic Un*x, SEPlib, Madagascar), debts which we gratefully acknowledge. The author wishes to record his special gratitude to Sergey Fomel, the architect of Madagascar, for his inspiring ideas and his generous and crucial help in the integration of IWAVE into Madagascar.

REFERENCES

Brown, D., 1984, A note on the numerical solution of the wave equation with piecewise smooth coefficients: Mathematics of Computation, **42**, 369–391.

Cohen, G. C., 2002, Higher order numerical methods for transient wave equations: Springer.

Fehler, M., and J. Keliher, 2011, SEAM Phase 1: Challenges of subsalt imaging in Tertiary basins, with emphasis on deepwater Gulf of Mexico: Society of Exploration Geophysicists.

Fomel, S., 2009, Madagascar web portal: http://www.reproducibility.org, accessed 5 April 2009.

Hu, W., A. Abubakar, and T. Habashy, 2007, Application of the nearly perfectly matched layer in acoustic wave modeling: Geophysics, **72**, SM169–SM176.

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

Moczo, P., J. O. A. Robertsson, and L. Eisner, 2006, The finite-difference time-domain method for modeling of seismic wave propagation: Advances in Geophysics, **48**, 421–516.

Muir, F., J. Dellinger, J. Etgen, and D. Nichols, 1992, Modeling elastic fields across irregular boundaries: Geophysics, **57**, 1189–1196.

Padula, A. D., W. Symes, and S. D. Scott, 2009, A software framework for the abstract expression of coordinate-free linear algebra and optimization algorithms: ACM Transactions on Mathematical Software, **36**, 8:1–8:36.

Santosa, F., and W. Symes, 2000, Multipole representation of small acoustic sources: Chinese Journal of Mechanics, **16**, 15–21.

Symes, W., 2014, IWAVE structure and basic use cases, *in* TRIP 2014 Annual Report: The Rice Inversion Project. (available summer 2015).

Symes, W., and T. Vdovina, 2009, Interface error analysis for numerical wave propagation: Computational Geosciences, **13**, 363–370.

Symes, W. W., D. Sun, and M. Enriquez, 2011, From modelling to inversion: designing a well-adapted simulator: Geophysical Prospecting, **59**, 814–833. (DOI:10.1111/j.1365-2478.2011.00977.x).

Terentyev, I., and W. Symes, 2009, Subgrid modeling via mass lumping in constant density acoustics: Technical Report 09-06, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA.

Virieux, J., 1984, SH-wave propagation in heterogeneous media: Velocity stress finite-difference method: Geophysics, **49**, 1933–1957.

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

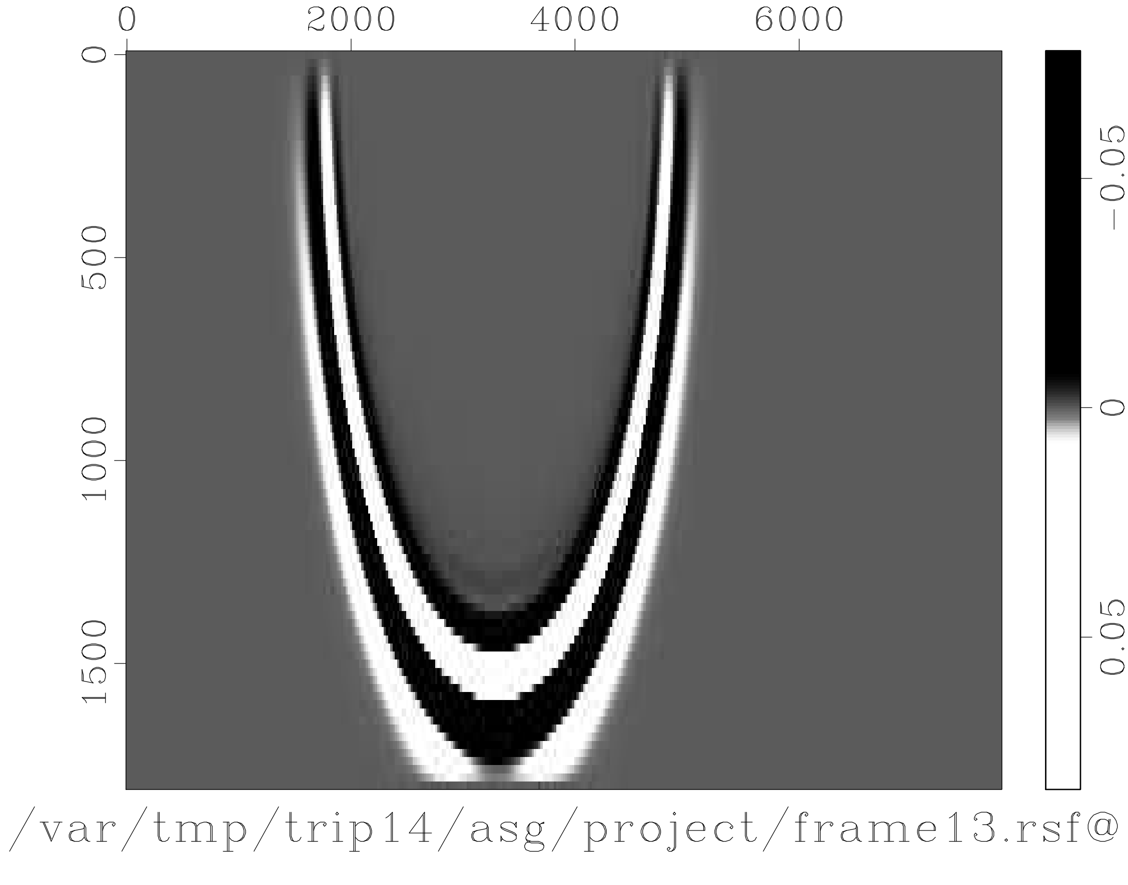


Figure 1: Point source field, homogeneous medium with $v_p = 1.5$ km/s, at 1.2 s asg/project frame13

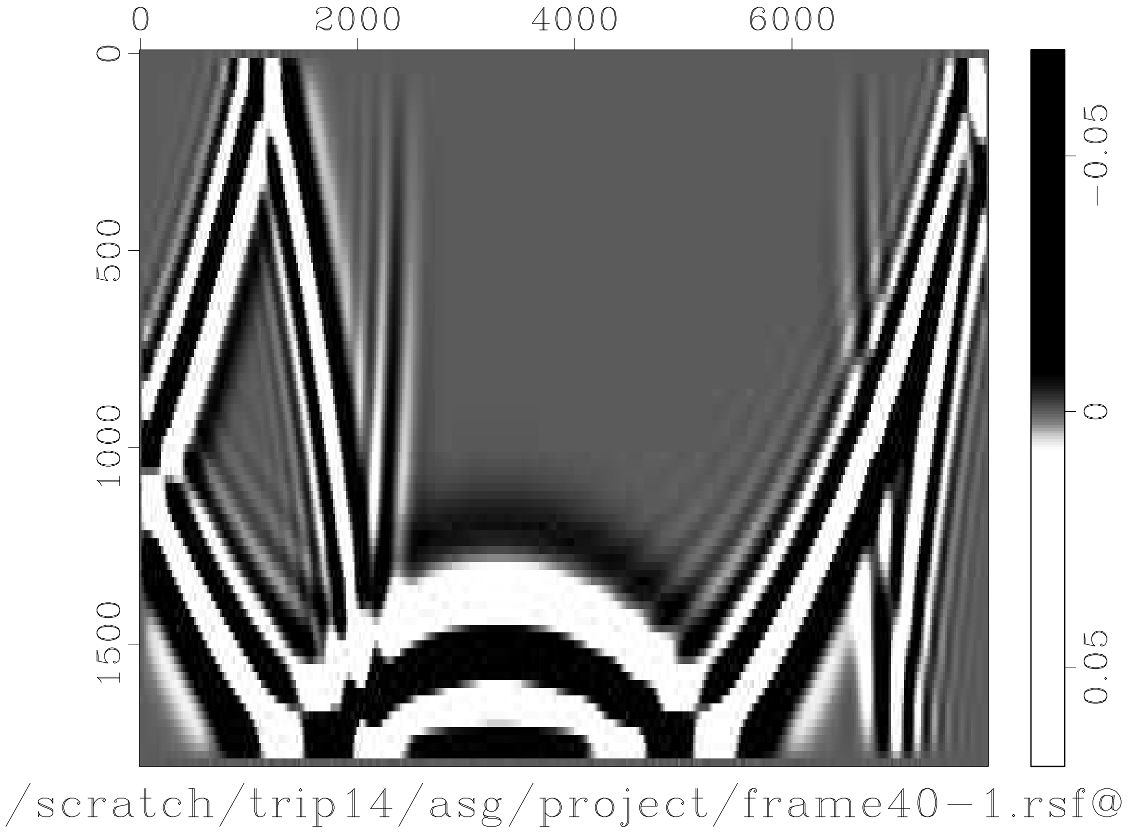


Figure 2: Point source field at 4.0 s, after interaction with reflecting (zero-pressure) boundaries asg/project frame40-1

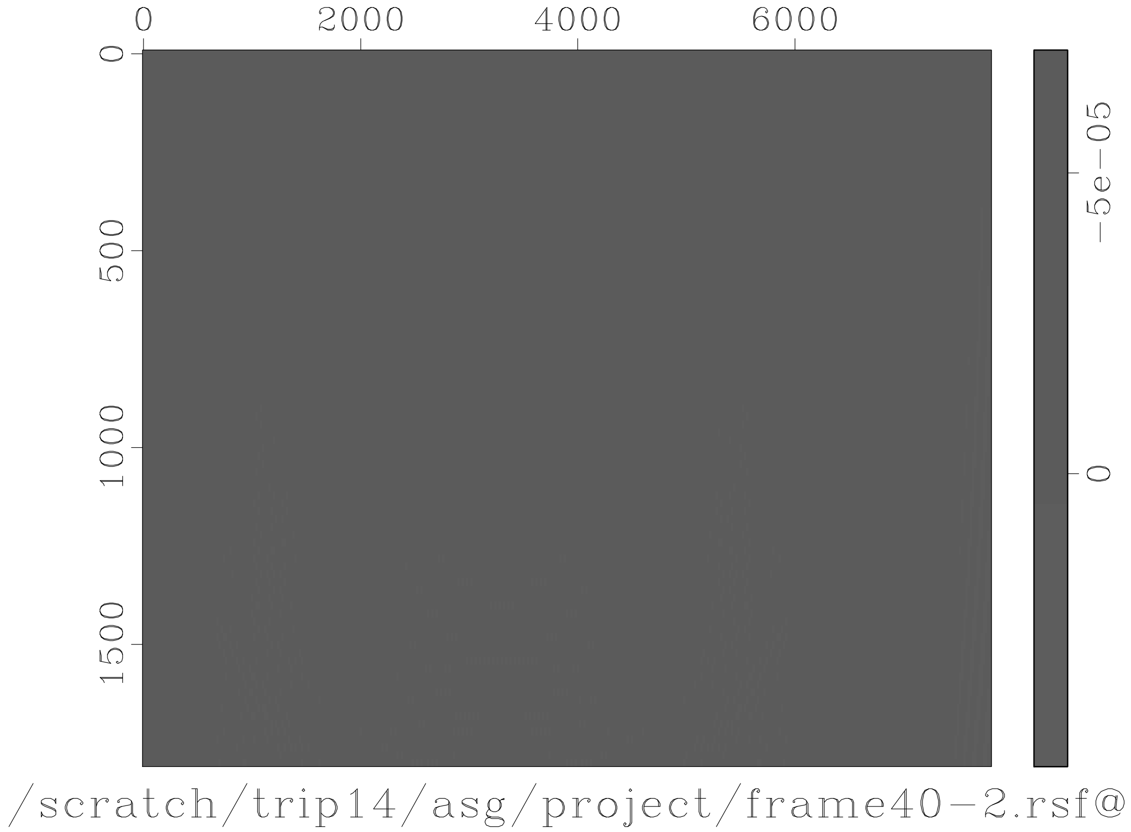


Figure 3: Point source field at 4.0 s, after interaction with 250 m PML boundary zones on bottom and sides ($\eta_0 = 1.0$) - same grey scale as Figure 2. Longest wavelength carrying significant energy is roughly 500 m. asg/project frame40-2

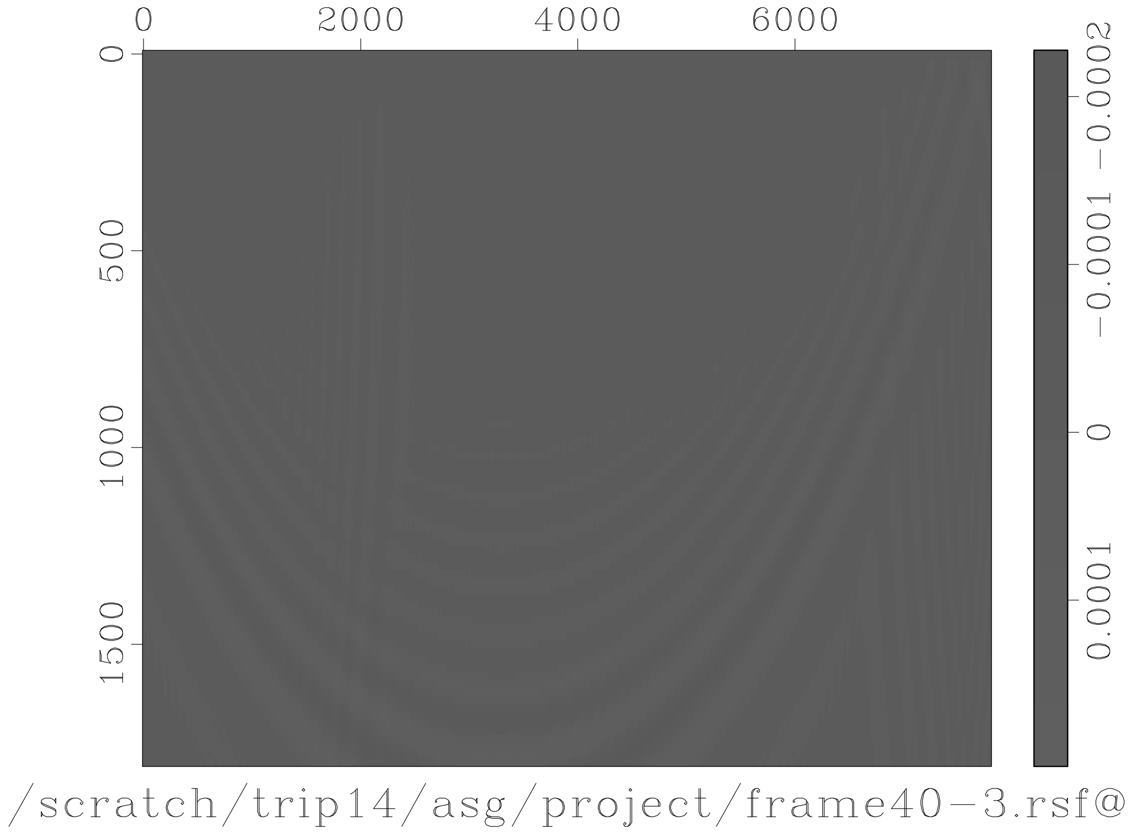


Figure 4: Point source field at 4.0 s, after interaction with 100 m PML boundary zones on bottom and sides ($\eta_0 = 1.0$) - same grey scale as Figure 2. asg/project frame40-3

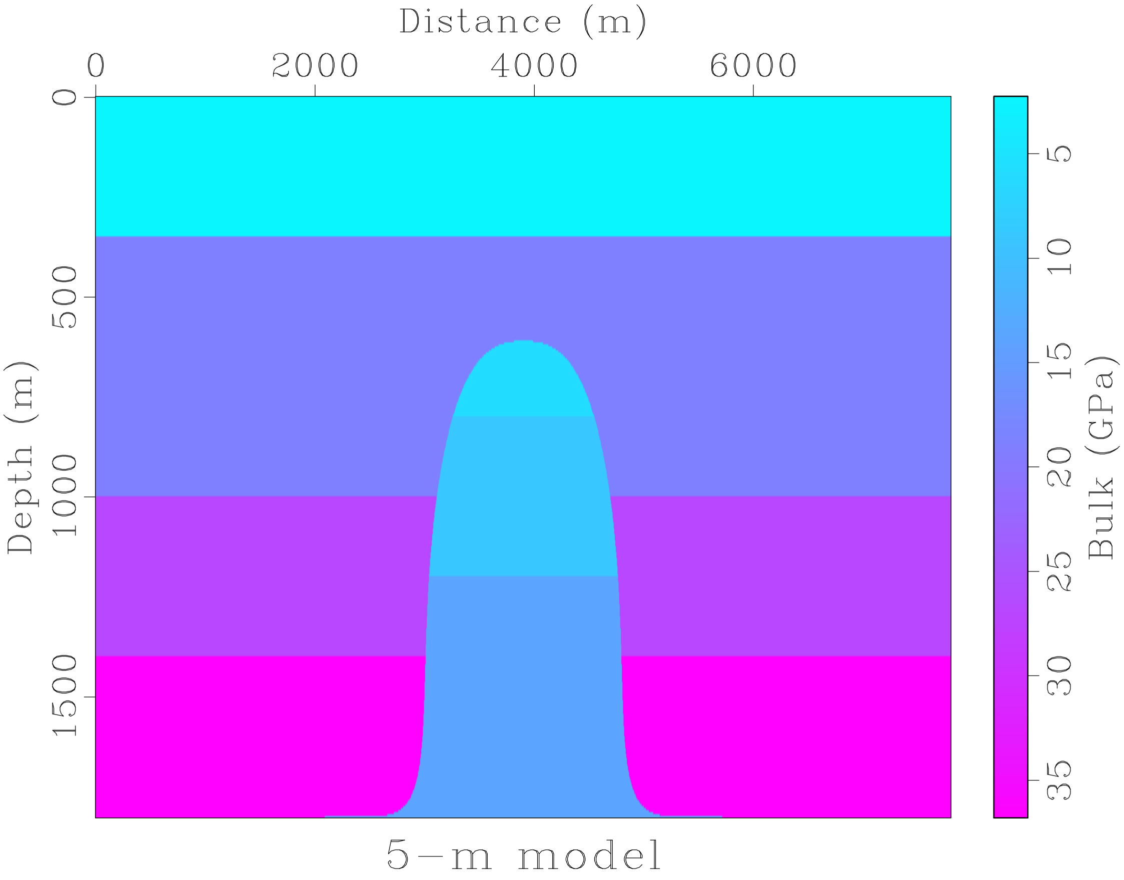


Figure 5: Dome bulk modulus asg/project bm1

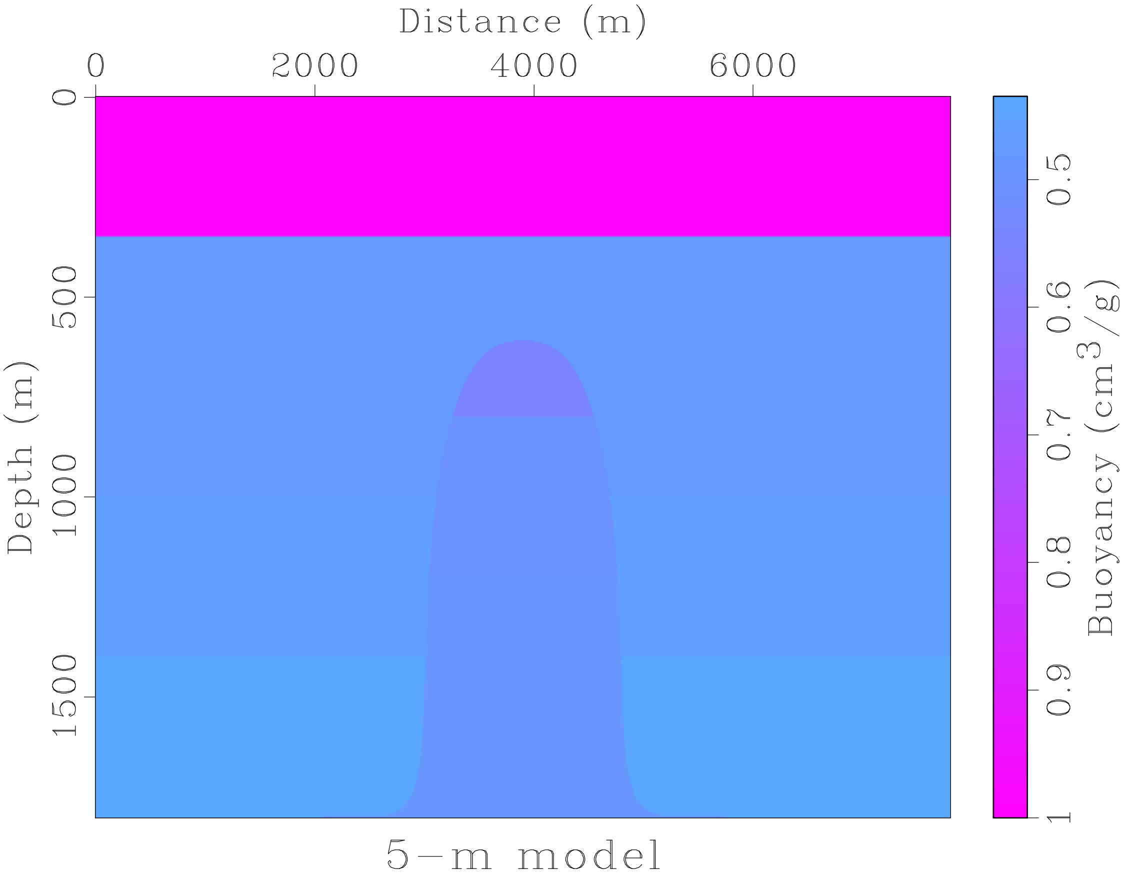


Figure 6: Dome buoyancy asg/project by1

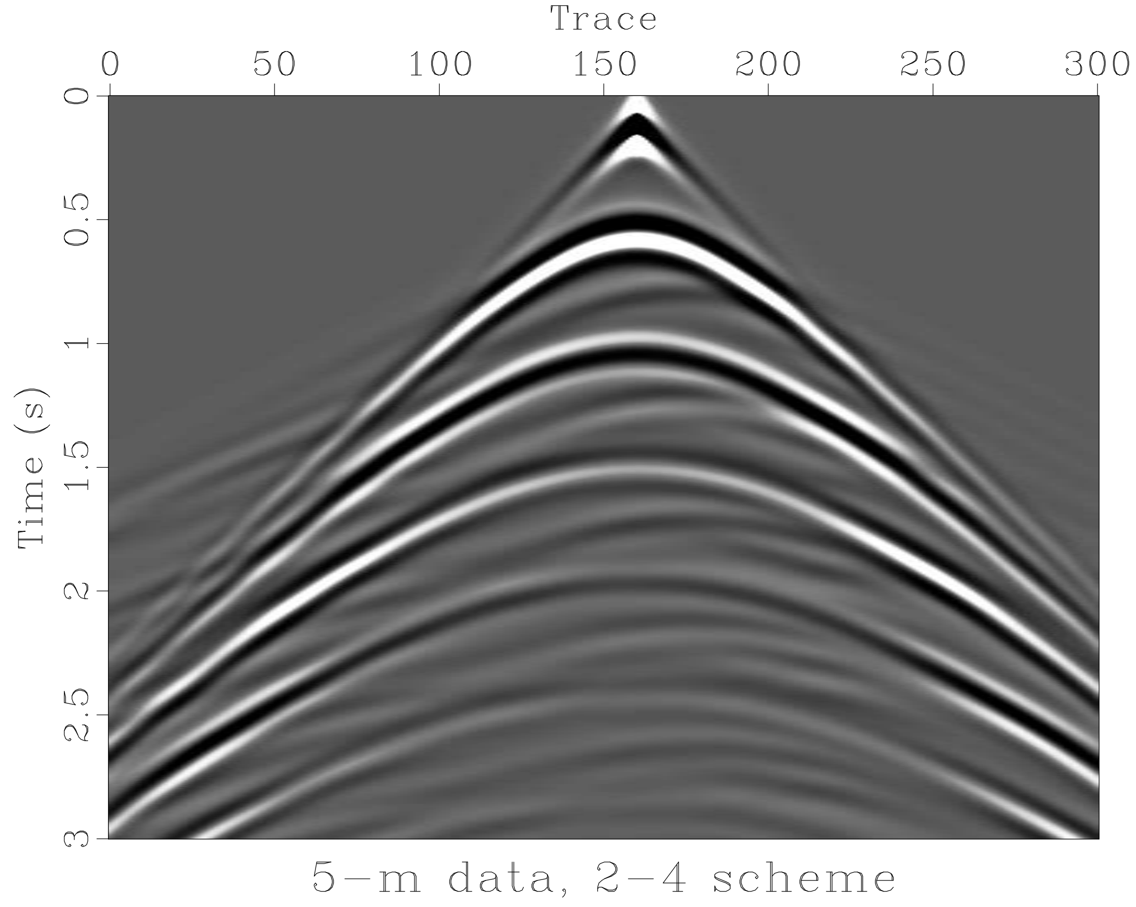


Figure 7: 2D shot record, (2,4) staggered grid scheme, $\Delta x = \Delta z = 5$ m, appropriate Δt, 301 traces: shot x = 3300 m, shot z = 40 m, receiver x = 100 - 6100 m, receiver z = 20 m, number of time samples = 1501, time sample interval = 2 ms. Source pulse = zero phase trapezoidal [0.0, 2.4, 15.0, 20.0] Hz bandpass filter. $\boxed{\text{asg/project data1}}$

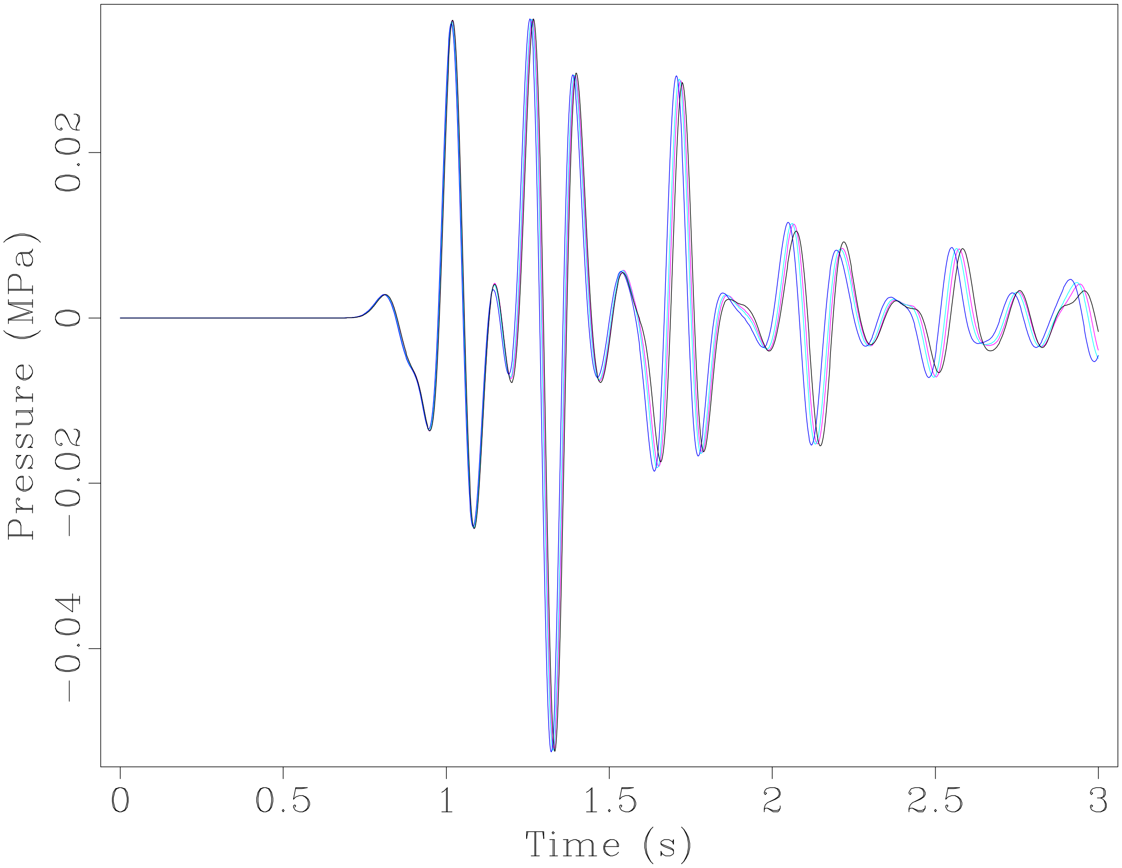


Figure 8: Trace 100 (receiver x = 2100 m) for $\Delta x = \Delta z = 20$ m (black), 10 m (blue), 5 m (green), and 2.5 m (red). Note arrival time discrepancy after 1 s: this is the interface error discussed in (Symes and Vdovina, 2009). Except for the 20 m result, grid dispersion error is minimal. asg/project trace

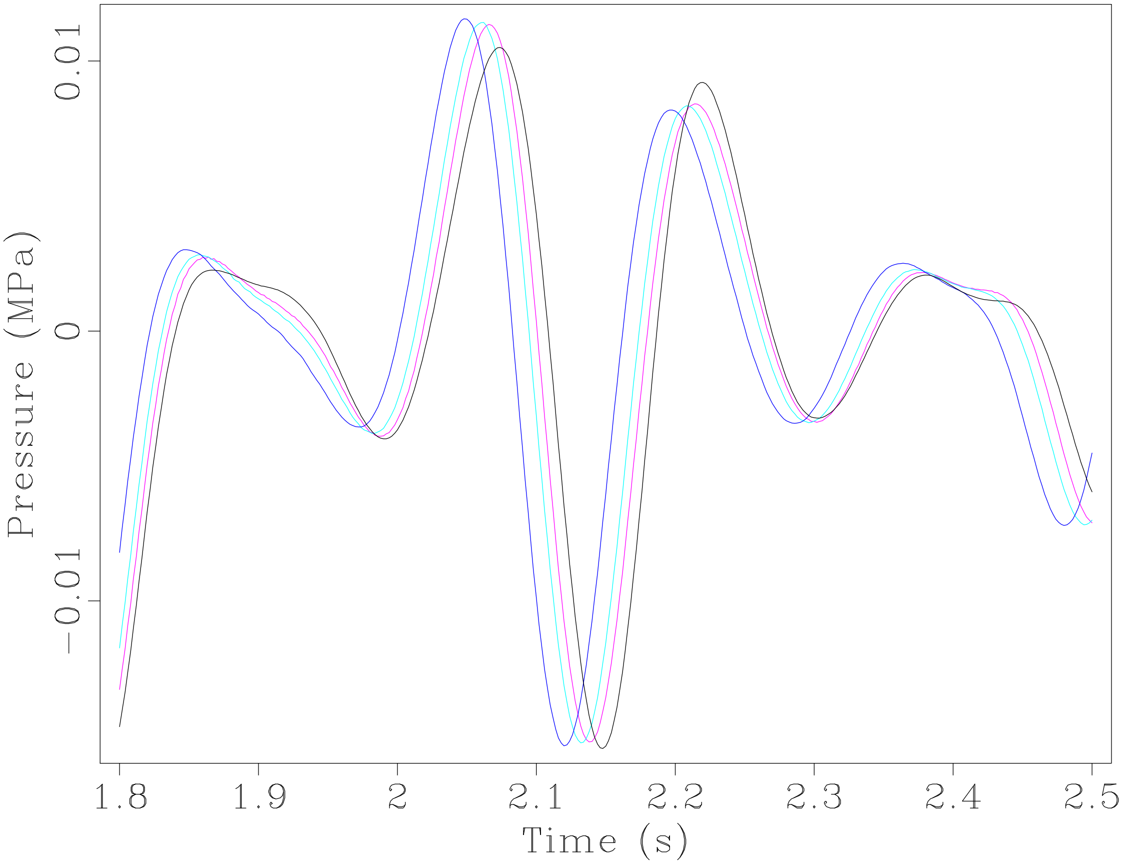


Figure 9: Trace 100 detail, 1.8-2.5 s, showing more clearly the first-order interface error: the time shift between computed events and the truth (the 2.5 m result, more or less) is proportional to Δt, or equivalently to Δz. asg/project wtrace

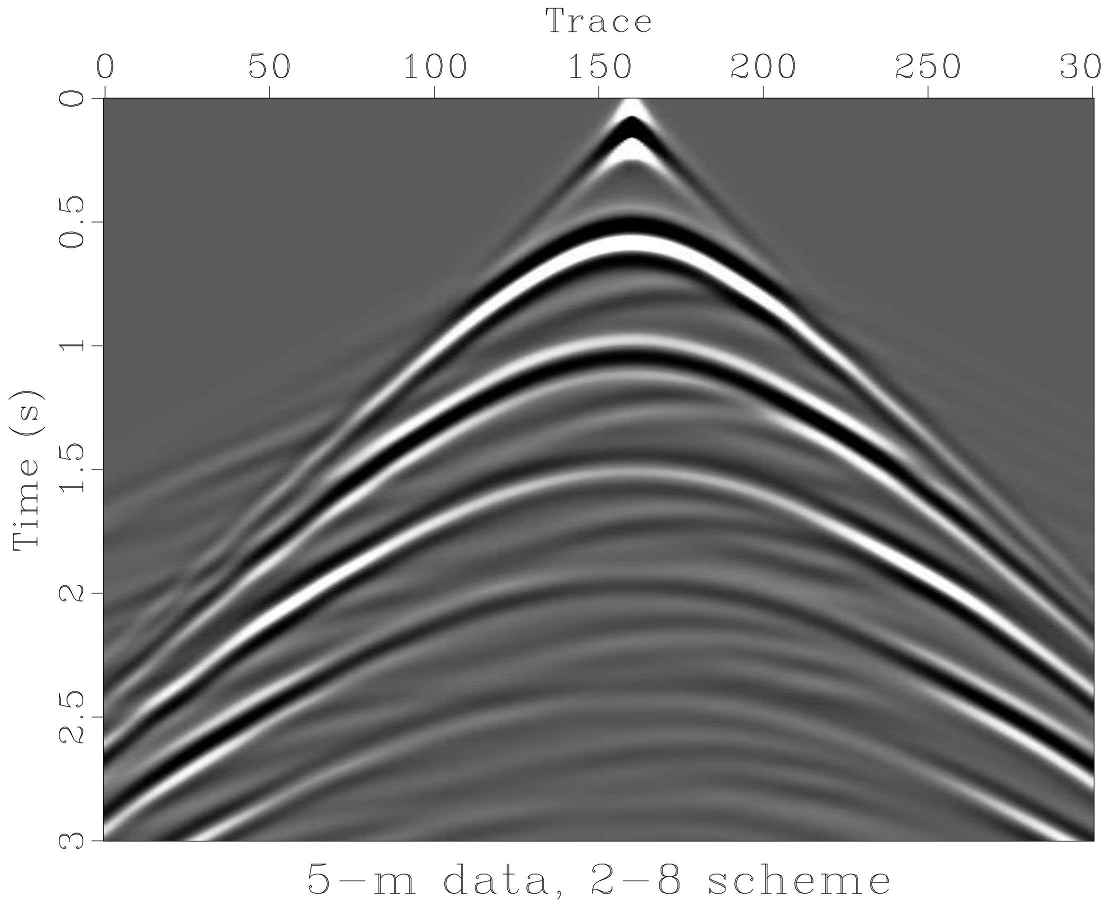


Figure 10: 2D shot record, (2,8) scheme, other parameters as in Figure 7. asg/project data8k1

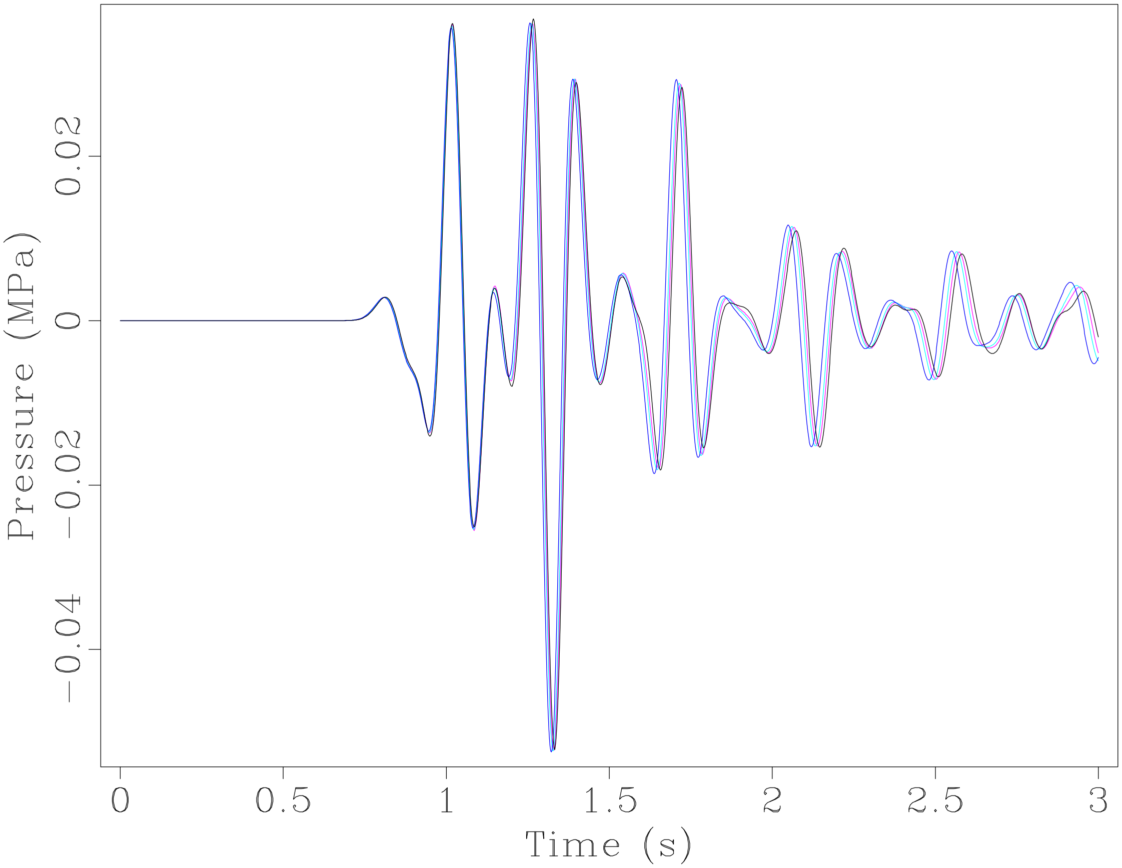


Figure 11: Trace 100 computed with the (2,8) scheme, other parameters as described in the captions of Figures 7 and 8. asg/project trace8k

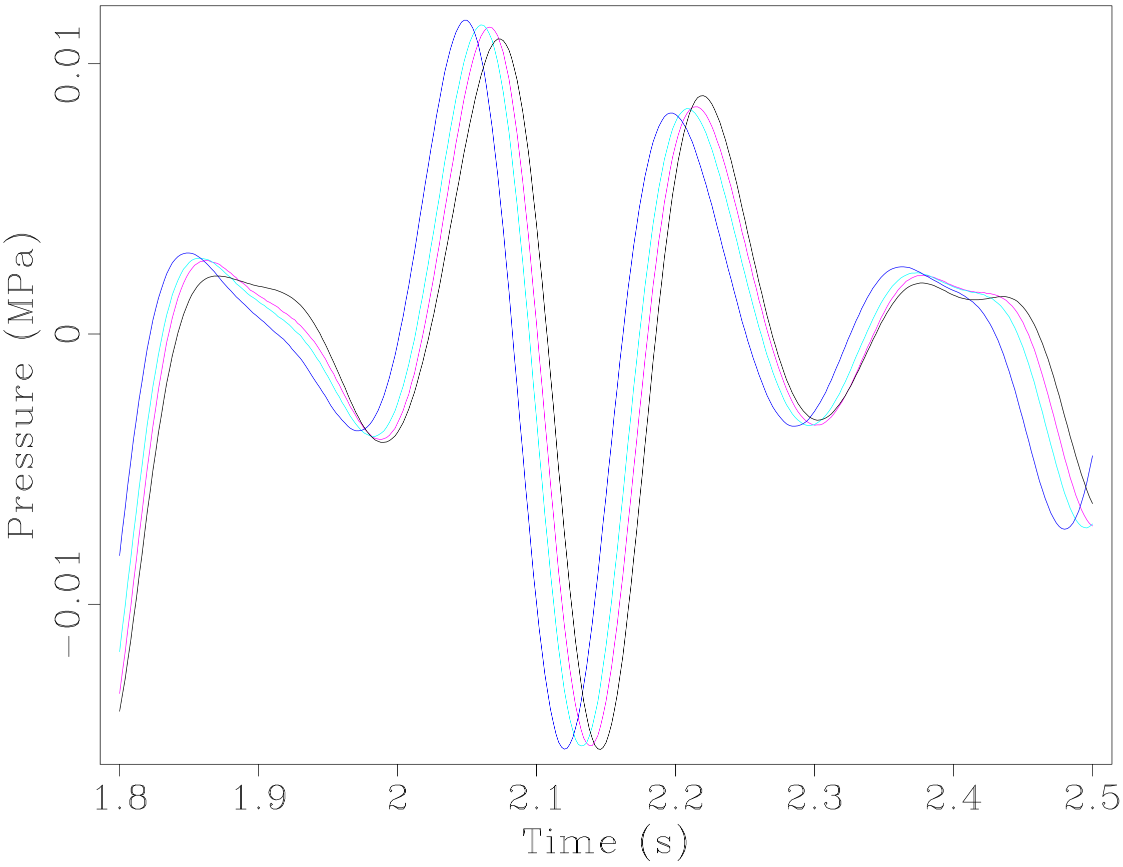


Figure 12: Trace 100 detail, 1.8-2.5 s, (2,8) scheme.. Comparing to Figure 9, notice that the dispersion error for the 20 m grid is considerably smaller, but the results for finer grids are nearly identical to those produced by the (2,4) grids - almost all of the remaining error is due to the presence of discontinuities in the model. asg/project wtrace8k

Wave equation based stencil optimizations on a multi-core CPU

Muhong Zhou

ABSTRACT

Wave propagation stencil kernels are engines of seismic imaging algorithms. These kernels are both compute- and memory-intensive. This work targets improving the performance of wave equation based stencil code parallelized by OpenMP on a multi-core CPU. To achieve this goal, we explored two techniques: improving vectorization by using hardware SIMD technology, and reducing memory traffic to mitigate the bottleneck caused by limited memory bandwidth. We show that with loop interchange, memory alignment, and compiler hints, both icc and gcc compilers can provide fully-vectorized stencil code of any order with performance comparable to that of SIMD intrinsic code. To reduce cache misses, we present three methods in the context of OpenMP parallelization: rearranging loop structure, blocking thread accesses, and temporal loop blocking. Our results demonstrate that fully-vectorized high-order stencil code will be about 2X faster if implemented with either of the first two methods, and fully-vectorized low-order stencil code will be about 1.2X faster if implemented with the combination of the last two methods. Our final best-performing code achieves 20%∼30% of peak GFLOPs/sec, depending on stencil order and compiler.

Note: this thesis is part of the 2014 TRIP annual report:

`http://www.trip.caam.rice.edu/reports/2014/trip2014_report.html`

Discontinuous Galerkin and Finite Difference Methods for the Acoustic Wave Equation with Smooth Coefficients

Mario J. Bencomo

ABSTRACT

This thesis analyzes the computational efficiency of two types of numerical methods: finite difference (FD) and discontinuous Galerkin (DG) methods, in the context of 2 D acoustic equations in pressure-velocity form with smooth coefficients. The acousti c equations model propagation of sound waves in elastic fluids, and are of particula r interest to the field of seismic imaging. The ubiquity of smooth trends in real data, and thus in the acoustic coefficients, validates the importance of this novel study. Previous work, from the discontinuous coefficient case of a two-layered media, demonstrates the efficiency of DG over FD methods but does not provide insight for the smooth coefficient case. Floating point operation (FLOPs) counts are compared, relative to a prescribed accuracy, for standard 2-2 and 2-4 staggered grid FD methods, and a myriad of standard DG implementations. This comparison is done in a serial framework, where FD code is implemented in C while DG code is written in Matlab. Results show FD methods considerably outperform DG methods in FLOP count. More interestingly, implementations of quadrature based DG with mesh refinement (for lower velocity zones) yield the best results in the case of highly variable media, relative to other DG methods.

Note: This thesis is part of the 2014 TRIP annual report:

`http://www.trip.caam.rice.edu/reports/2014/trip2014_report.html`