Extending Toldi's velocity analysis algorithm to include geologic structure

Paul Fowler

INTRODUCTION

John Toldi in his dissertation (Toldi, 1985) developed velocity analysis algorithms that had some attractive features. The algorithms were automatic, requiring no human intervention to pick velocities or horizons, and the stacking velocities found were related to an underlying model of (laterally varying) interval velocities. In his implementation he made two strong assumptions: first, that all reflectors were nearly flat, and second, that he knew a background velocity model well enough to convert from depth to time at the start, and did not need to update this calculation as he proceeded. For the models and data he considered, these were reasonable assumptions. For example, in the Central Valley data set he used, the time sag in the middle of the section was almost certainly a velocity pulldown on a flat bed, and he could assume a priori that it was not structural. By smoothing his objective function (semblance) over a window greater than the amount of the time sag, he could ignore the apparent structure as he updated his interval velocity model.

In this paper I will consider ways to extend Toldi's algorithms to incorporate explicitly consideration of geologic structure. Toward the end of his dissertation, Toldi examined the case of simple dipping reflectors. Loinger (1983) suggested applying a similar algorithm to DMO corrected data. I have not found a workable and flexible algorithm based on either of these approaches. Instead, in the last SEP report (Fowler, 1985), I proposed an extension on Toldi's methods using prestack migration velocities instead of stacking velocities to attempt to incorporate structural effects as well. I argued there that prestack constant velocity migrations form a better reference velocity analysis space than do constant velocity stacks because they better unscramble the misleading effects complex structure can have on observed apparent velocities. The migration method looks at individual points as diffractors, and considers the apparent changes in their lateral and vertical positions as well as in the apparent migration velocity as an interval velocity model is updated.

I will here re-examine the construction of an algorithm for migration velocity analysis, discussing two different possible formulations based on the linear theory developed in my previous paper. I will also outline methods whose development is similar to the migration algorithms, but using stacking velocities rather than migration velocities. These latter methods attempt to distinguish between structural perturbations and velocity variations without the strict assumptions Toldi used, but are limited to gentle structure only. They may thus be considered as intermediate in approach between Toldi's algorithms and the more radical extensions using migration velocities that I have proposed.

MIGRATION VELOCITY ANALYSIS RE-EXAMINED

Toldi's algorithms, and my proposed extensions to them, attempt to maximize an objective function that measures in some sense the "quality" of the corresponding imaged (stacked or prestack migrated) section. Toldi used total semblance, but for our purposes here this may be considered equivalent to maximizing the cumulative energy. Call the energy function E and the objective function Q. Given a migration slowness function $\mathbf{w}=w(y,\tau)$, where y is midpoint and τ is migrated time, let $E(w,y,\tau)$ be the energy in the corresponding section. The objective function Q is then defined as

$$Q(\mathbf{w}) = \sum_i \sum_j E(w(y_i,\tau_j),y_i,\tau_j) \tag{1}$$

In implementation, the algorithm begins by imaging the data with a suite of constant slowness functions. Evaluation of the objective function may be done either by extracting the image corresponding to a specified slowness function, and then summing up E for that image, or by calculating E for each reference panel, and interpolating and summing the E values corresponding to the choice of slowness function.

The process of choosing a migration velocity function and inverting it for interval velocities may then be formulated as an optimization problem. The optimization algorithm uses a gradient to update iteratively a starting model. The gradient $\nabla_\mathbf{w} Q$ of the objective function Q with respect to the migration slownesses $\mathbf{w}$ can be evaluated easily from the constructed cube of energy as a function of w, y, and τ. What is really needed for a good algorithm, however, is the gradient $\nabla_\mathbf{m} Q$ with respect to changes in an underlying model $\mathbf{m}$ of interval slowness as a function of the physical location x and depth z. To derive this one needs to specify a relation between $\mathbf{w}$ and $\mathbf{m}$. The simplest approach is to treat each midpoint separately and calculate $\mathbf{w}$ as an rms average of $\mathbf{m}$. This approach, however, assumes that the velocity function is nearly laterally invariant. In the presence of complex structure this will often be a poor assumption. Allowing for laterally varying velocities requires a more complicated relation between the interval slownesses and the migration slownesses.

Suppose an interval velocity model $\mathbf{m}(x,z)$ is specified. Pick a point $\mathbf{r}=(x_r,z_r)$. For the purposes of analyzing migration velocities, one can visualize this point as causing a pyramidal diffraction surface in the prestack data with apex at $\mathbf{d}=(y_r,\tau_r)$ and characterized by a migration slowness w_r. The observed data then arises from the constructive and destructive interference of all such diffractions. Suppose one now perturbs the velocity model at some anomaly point $\mathbf{a}=(x_a,z_a)$. One wants to find the resulting change in the observed migration slowness, i.e., $\partial w_r / \partial m_a$. In my previous paper I showed how to find such a linear relation between a perturbation in the model $\mathbf{m}$ and the resulting perturbations in $\mathbf{d}$ and w_r. I will summarize that derivation now and then return to the problem of incorporating this information into an iterative optimization algorithm.

Calculating derivatives

In this section I will repeat in summary form from my paper (Fowler, 1985) in the last SEP report some derivations of linear relations between interval slowness perturbations and the resulting perturbations in scatterer location and apparent migration slowness. Consider first a single point diffractor at $\mathbf{r}$ in a medium of constant slowness w. If one were to run a seismic survey passing over this point, the kinematics of the prestack point diffractor would be given by the pyramid equation (Claerbout, 1985)

$$t = w\sqrt{{z_r}^2+(y-h-x_r)^2} + w\sqrt{{z_r}^2+(y+h-x_r)^2} \tag{2}$$

where y is midpoint and h is offset. Suppose now that the slowness model is perturbed. The travel-time data for the point diffractor would now be a set $\{t_{ik}, y_i, h_k\}$ which would no longer satisfy equation (2) exactly. Here the subscripts i and k index midpoints and offsets, respectively. If the perturbations are not too large, it should be possible to define a slowness W, a zero-offset time T, and a location Y for which an equation of the form

$$t = \sqrt{T^2+W^2(y-h-Y)^2} + \sqrt{T^2+W^2(y+h-Y)^2} \tag{3}$$

best fits the data points in a least-squares sense. Note that it is necessary to consider the changes in T and Y as well as W because, for laterally varying perturbations in the slowness model, it would not in general be true that $T=Wz_r$ and $Y=x_r$ as they would be for the constant slowness background of the starting model.

Now linearize around an initial value of $(\hat{W}, \hat{T}, \hat{Y})$, so that

$$t_{ik} \approx t(\hat{W}, \hat{T}, \hat{Y}) + \frac{\partial t}{\partial W} \Delta W + \frac{\partial t}{\partial T} \Delta T + \frac{\partial t}{\partial Y} \Delta Y \tag{4}$$

where all the partial derivatives are evaluated at $(\hat{W}, \hat{T}, \hat{Y}, y_i, h_k)$. The solution for this least-squares system is of the form

$$\begin{bmatrix} \Delta W \\ \Delta T \\ \Delta Y \end{bmatrix} = \frac{1}{D} \begin{bmatrix} \sum_{i,k} A_{ik} \ \Delta t_{ik} \\ \sum_{i,k} B_{ik} \ \Delta t_{ik} \\ \sum_{i,k} C_{ik} \ \Delta t_{ik} \end{bmatrix} \tag{5}$$

The coefficients A_{ik}, B_{ik}, C_{ik}, and D are complicated functions of W, T, and Y, as well as depending on the geometry of the seismic experiment. I derived analytic solutions for them in my previous paper and will not repeat them here.

These equations (5) describe how the $W, T,$ and Y arising from a single point diffractor at **r** change when the model, and hence also the travel-times, are perturbed. I intend, as the notation suggests, to identify (W, T, Y) with (w, τ, y). The definition of migration slowness by maximizing the energy in the image is not tractable for easy analytical expression, since it is fundamentally dependent on the wave nature of the original data. Hence I substitute a definition based on ray tracing and travel-times. The two are not exactly interchangeable, but they should agree well if the velocity model does not vary too radically.

To complete the linearization, I need a relation between Δt_{ik} and Δm_a. For a given ray S_{ik} one has

$$t_{ik} = \int_{S_{ik}} dS_{ik} \, m(x_a, z_a) \tag{6}$$

Perturb the model and calculate the changes in travel times integrating the slowness perturbations along the *unperturbed* ray S_{ik}:

$$\Delta t_{ik} = \int_{S_{ik}} dS_{ik} \, \Delta m(x_a, z_a) \tag{7}$$

This last calculation is valid for a general model, but to apply it directly requires tracing many rays at every iteration. I choose instead to evaluate analytically these derivatives against a simple constant slowness background model for which the ray paths are straight, and use these approximate values in place of the more accurate values that would be calculated by ray tracing using an iteratively updated model.

Figure 1 shows the geometry of the ray path for a particular diffractor point, midpoint, and offset in a simple, constant slowness reference medium. Represent the traveltime data as (t_{ik}, y_i, h_k). Using the notation of figure 1, equation (7) becomes

$$\Delta t_{ik} = \int_{x_a} dx_a \int_{z_a} dz_a \, \Delta m(x_a, z_a) \left(\frac{\delta_1}{\cos\theta_{ik}} + \frac{\delta_2}{\cos\phi_{ik}} \right) \tag{8}$$

where

$$\delta_1 \equiv \delta[x_a - y' + \mu_{ik}(z_a)] \tag{9}$$

and

$$\delta_2 \equiv \delta[x_a - y' - \mu_{ik}(z_a)] \tag{10}$$

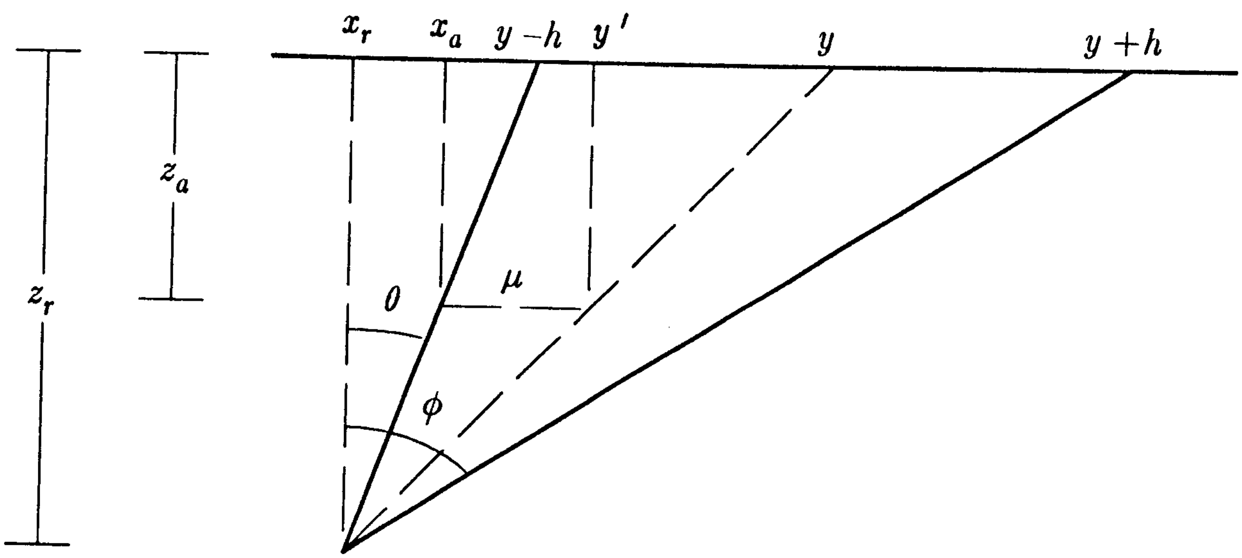

FIG. 1. Geometry of rays for a single diffractor point and a constant background slowness. The rays for a single trace with midpoint y and offset h are shown. The diffractor is at (x_r, z_r). The point in the model at which the slowness is perturbed is (x_a, z_a). The quantities θ, ϕ, μ, and y' are used in calculating the effect on the traveltime of perturbing the slowness.

Rewrite equation (5) as

$$\Delta W(Y,T) = \frac{1}{D} \sum_i \sum_k A_{ik}\, \Delta t_{ik} \tag{11}$$

and substitute from equation (8) for Δt_{ik} to yield

$$\Delta W(Y,T) = \frac{1}{D} \sum_i \sum_k A_{ik} \int_{x_a} dx_a \int_{z_a} dz_a\, \Delta m\,(x_a\,,z_a) \left(\frac{\delta_1}{\cos\theta_{ik}} + \frac{\delta_2}{\cos\phi_{ik}} \right) \tag{12}$$

Pull the integrals outside the sums and make the identification of $W(Y,T)$ with $w\,(y_r\,,\tau_r\,)$ to get

$$\Delta w\,(\mathbf{d}) = \int_{\mathbf{a}} d\,\mathbf{a} G_W(\mathbf{d},\mathbf{a}) \Delta m\,(\mathbf{a}) \tag{13}$$

where

$$G_W(\mathbf{d},\mathbf{a}) = \sum_{i=1}^{N_y} \sum_{k=1}^{N_h} \frac{A_{ik}}{D} \left(\frac{\delta_1}{\cos\theta_{ik}} + \frac{\delta_2}{\cos\phi_{ik}} \right) \tag{14}$$

This Green function $G_W(\mathbf{d},\mathbf{a})$ can be identified with $\partial w\,/\partial m_a$, since they both represent the change in $w\,(y\,,\tau)$ caused by a perturbation in $m\,(x\,,z\,)$. Evaluation of the Green function G_W in a form suitable for implementation involves substituting for the trigonometric terms in equation (14) and using the delta functions to eliminate one sum. The result for G_W is

$$G_W(\mathbf{d},\mathbf{a}) = \frac{\gamma}{\tau_r\,D} \left[\tau_r^2 + w^2(\gamma x_a - y_r\,)^2 \right]^{1/2} \times \tag{15}$$

$$\sum_{k=1}^{N_h} \left[A\,(y = \gamma x_a + h_k\,,h_k\,) + A\,(y = \gamma x_a - h_k\,,h_k\,) \right]$$

where

$$\gamma = \frac{w\,\tau_r}{w\,\tau_r - z_a} \tag{16}$$

One can also write similar Green function representations for $G_T(\mathbf{d},\mathbf{a}) = \partial\tau/\partial m_a$ and for $G_Y(\mathbf{d},\mathbf{a}) = \partial y\,/\partial m_a$ simply by substituting B or C in place of A in equations (14) and (15).

Eulerian and Lagrangian approaches

Having calculated in the preceding section how w_r changes when the underlying slowness model $\mathbf{m}$ is perturbed, I now need to show how to incorporate this information in constructing the gradient $\nabla_{\mathbf{m}} Q$. One major difficulty is apparent: y_r and τ_r also change when m_a is perturbed, so the point $\mathbf{d}$ at which w_r is to be evaluated moves. The key question to ask is, "At what points are all the partial derivatives evaluated?" There are two possibilities here: one may evaluate $\partial Q / \partial m$ for fixed points $\mathbf{d}$ in the data space of migration slownesses, or one may track the image of fixed diffracting points $\mathbf{r}$ as they move in (y, τ) space during successive iterations of the algorithm. The choice is somewhat analogous to the choice between Lagrangian or Eulerian viewpoints in fluid dynamics. A similar problem arises there, requiring a choice depending whether spatial gradients are to be calculated at a fixed location (Eulerian) or for a particular particle (Lagrangian). [1] Here the choice is between evaluating gradients for a fixed location $\mathbf{d}$ in the (y, τ) data space or for a fixed "particle" $\mathbf{r}$ in (x, z) space. I choose (a bit arbitrarily) to call the first choice "Eulerian" and the second one "Lagrangian." The names are not intended to be taken too seriously, but I do find the analogy useful in wading through what can otherwise become a forbidding morass of notation. ("It looks like you are doing general relativity theory," Jon Claerbout once commented after a seminar of mine became bogged down in notational complexities.)

In my paper on migration velocity analysis in SEP-44 I implicitly adopted the Eulerian viewpoint. In that paper I showed that the desired expression for the gradient of the objective function is given by:

$$(\nabla_{\mathbf{m}} \mathbf{Q})_a = \frac{\partial Q}{\partial m_a} = \sum_{y, \tau} \left[\frac{\partial E}{\partial w} \left(\frac{\partial w}{\partial m_a} - \frac{\partial w}{\partial \tau} \frac{\partial \tau}{\partial m_a} - \frac{\partial w}{\partial y} \frac{\partial y}{\partial m_a} \right) \right] \tag{17}$$

In applying this equation, $\partial E / \partial w$ is to be calculated by finite difference approximation at each point (y, τ) in the data space. Then $\partial w / \partial m_a = G_W(\mathbf{d}, \mathbf{a})$ is calculated from equation (15). The remaining terms in equation (17) compensate for the change in the particular diffracting point $\mathbf{r}$ which is being mapped into the data space coordinates (y, τ) at each iteration. The derivatives $\partial \tau / \partial m_a$ and $\partial y / \partial m_a$ are calculated just as $\partial w / \partial m_a$ is. The remaining derivatives $\partial w / \partial \tau$ and $\partial w / \partial y$ are calculated by finite differences, estimating changes now not in the whole (w, y, τ) cube, but specifically along the surface of the submanifold defined by the current best estimate of $w(y, \tau)$.

1) For introductory discussion of the Eulerian and Lagrangian approaches in mechanics problems, see Batchelor, 1967, or Aki and Richards, 1980. The latter reference notes that both methods were actually developed first by Leonhard Euler!

It is now possible to define

$$G(\mathbf{d},\mathbf{a}) = G_W(\mathbf{d},\mathbf{a}) - \frac{\partial w}{\partial \tau} G_T(\mathbf{d},\mathbf{a}) - \frac{\partial w}{\partial y} G_Y(\mathbf{d},\mathbf{a}) \tag{18}$$

This function G provides a jacobian matrix that can be incorporated readily in a steepest descent optimization algorithm similar to those outlined by Toldi. A generalized steepest ascent optimization algorithm of the type used here looks like:

Set $\mathbf{m}$ to starting model: $\mathbf{m} = \hat{\mathbf{m}}$
Set $\mathbf{w}$ to starting value: $\mathbf{w} = \hat{\mathbf{w}}$
Calculate $\mathbf{G}$ by taking derivatives of $\mathbf{w}$ at $\hat{\mathbf{m}}$.

Begin loop on iterations

1. Form $\nabla_{\mathbf{m}} Q$ at current model point $\mathbf{m}$:
   $$(\nabla_{\mathbf{w}} Q)_{\mathbf{d}} = \frac{\partial Q}{\partial w_{\mathbf{d}}} \approx \frac{E(w(\mathbf{d})+\Delta w, \mathbf{d}) - E(w(\mathbf{d}),\mathbf{d})}{\Delta w}$$
   $$\nabla_{\mathbf{m}} Q = \mathbf{G}^T \nabla_{\mathbf{w}} Q$$
2. Line search for α that maximizes $Q(\mathbf{m}+\alpha\nabla_{\mathbf{m}} Q)$
   $$Q(\mathbf{m}+\alpha\nabla_{\mathbf{m}} Q) = Q[\mathbf{w}+\alpha\mathbf{G}\nabla_{\mathbf{m}} Q]$$
3. Update model
   $$\mathbf{m} = \mathbf{m}+\alpha\nabla_{\mathbf{m}} Q$$
   $$\mathbf{w} = \mathbf{w}+\alpha\mathbf{G}\nabla_{\mathbf{m}} Q$$
   recalculate $\mathbf{G}$ by taking derivatives at new $\mathbf{m}$.

End loop on iterations.

This type of algorithm is discussed further in my earlier paper and in Toldi's dissertation (from which it is copied with only slight changes). The important point to note is that G here relates perturbations on a fixed grid of model points $\mathbf{m}(\mathbf{a})$ to perturbations again on a fixed grid in data space $\mathbf{w}(\mathbf{d})$. The migration slownesses $\mathbf{w}$ are explicitly treated as a function of the coordinates y and τ. Thus the objective function Q is always evaluated by a sum over the same grid points $\mathbf{d}=(y_d, \tau_d)$. Likewise, the finite difference approximations to the derivatives are always calculated at the same (y_d, τ_d) values.

In the "Lagrangian," or fixed diffractor point, approach, the model space consists both of slowness anomaly points $\mathbf{a}$ and diffracting points $\mathbf{r}$; the same grid in (x, z) may or may not be used for both. A map must be explicitly maintained taking each diffracting point $\mathbf{r}$ in the model space to an image point $\mathbf{c} = (w(\mathbf{d}),\mathbf{d})$ in (w, y, τ) space. Clearly, the map will depend on $\mathbf{m}$, so $\mathbf{c}$ may be considered as a function both of $\mathbf{r}$ and of the slowness model $\mathbf{m}$. A straight forward application of the chain rule then yields

$$(\nabla_{\mathbf{m}}\mathbf{Q})_a = \frac{\partial Q}{\partial m_a} = \sum_{\mathbf{r}} \left[\frac{\partial E}{\partial w} \frac{\partial w}{\partial m_a} + \frac{\partial E}{\partial \tau} \frac{\partial \tau}{\partial m_a} + \frac{\partial E}{\partial y} \frac{\partial y}{\partial m_a} \right] \tag{19}$$

The identification of $\partial w / \partial m_a$ with $G_W(\mathbf{d},\mathbf{a})$, etc., proceeds as in the Eulerian case. The various partial derivatives of E can again be computed by finite difference approximations, but note that the points at which to evaluate them can only be known from knowing to which $\mathbf{c}(\mathbf{r})$ each $\mathbf{r}$ is mapped. The objective function $Q(\mathbf{m})$ is thus evaluated as $Q(\mathbf{c}(\mathbf{m}))$, and is evaluated as a sum over $\mathbf{r}$ instead of $\mathbf{d}$. Using notation similar to that in the Eulerian case, equation (19) can be recast as

$$\nabla_{\mathbf{m}} Q = \mathbf{G}_W^T \nabla_{\mathbf{w}} Q + \mathbf{G}_Y^T \nabla_{\mathbf{y}} Q + \mathbf{G}_T^T \nabla_{\tau} Q \tag{20}$$

I will abuse the notation somewhat at this point and simplify this equation even further by writing

$$\nabla_{\mathbf{m}} Q = \mathbf{G}^T \ \nabla_{\mathbf{c}} Q \tag{21}$$

where $\mathbf{G}^T = (\mathbf{G}_W^T, \mathbf{G}_Y^T, \mathbf{G}_T^T)$ is a "matrix" whose elements are 3-vectors, $\nabla_{\mathbf{c}} Q$ is a "vector" whose elements likewise are 3-vectors, and the product of them is understood to incorporate implicitly the summation in equation (20). Given this gradient $\nabla_{\mathbf{m}} Q$ Given this gradient one can then line search as in the algorithm outlined for the Eulerian case to find an optimal α for updating the model:

$$\mathbf{m} = \mathbf{m} + \alpha \ \nabla_{\mathbf{m}} Q \tag{22}$$

Continuing the manhandling of notation, I will also write the corresponding equation for updating $\mathbf{c}$ as

$$\mathbf{c}(\mathbf{m}+\alpha\nabla_{\mathbf{m}} Q\,) = \mathbf{c}(\mathbf{m}) + \alpha \mathbf{G} \nabla_{\mathbf{m}} Q \tag{23}$$

where $\mathbf{G} = (\mathbf{G}_W, \mathbf{G}_Y, \mathbf{G}_T)$ is another 3-vector "matrix," and the product this time is understood as acting on three copies of $\nabla_{\mathbf{m}} Q$, converting a normal vector into a vector of 3-vectors. This last equation, with the notation decrypted, simply tells how the vector $\mathbf{c}$ changes when the slowness model $\mathbf{m}$ is changed:

$$\mathbf{c}(\mathbf{r};\mathbf{m}) = \mathbf{c}(\mathbf{r};\hat{\mathbf{m}}) + \sum_{\mathbf{a}} \frac{\partial \mathbf{c}}{\partial m_a} \ \Delta m_a \tag{24}$$

The point of the above notational peculiarities is to allow casting of the Lagrangian algorithm in a form nearly identical to that given for the Eulerian case. The previously outlined steepest ascent routine may now be applied intact, with the only major change the substitution of $\mathbf{c}$ for $\mathbf{w}$ throughout.

I do not know which approach is the better to use for practical implementation, although I now lean toward the Lagrangian view. The Eulerian approach is perhaps easier to implement, in so much as it works with the migration slowness data on the grid on which it is provided and never really concerns itself with the model grid. It could have considerable problems, however, should the mapping from $\mathbf{r}$ to $\mathbf{d}$ fail to be one-to-one, or if the model space is chosen badly, leaving parts of the data wholly outside the image of the model space under the mapping from $\mathbf{r}$ to $\mathbf{d}$. In either of these cases it would be impossible to figure out which $\mathbf{r}$ to associate with a given $\mathbf{d}$, and the algorithm could be expected to fail in one fashion or another. The Lagrangian approach would have no such problem, as it works in principle only with the map from $\mathbf{r}$ to $\mathbf{d}$ and not the inverse; it would simply count a contribution from $\mathbf{r}$ more than once in calculating the objective function if it maps onto multiple values of $\mathbf{d}$, and would never look at those parts of the data which are not images of some part of the model space. The two approaches thus interact with the data and model fundamentally differently. The objective function calculation in the Eulerian approach weights all parts of the data equally, without concern for whether they are actually well determined by the given model. The Lagrangian method, on the other hand, ignores those parts of the data that are not the images of any point in the data, and may allow some parts of the data to contribute much more than others. In either case, judicious application of damping is almost certainly needed.

INCORPORATING STRUCTURAL PERTURBATIONS IN STACKING VELOCITY ANALYSIS

Toldi's implementations considered only the changes in the stacking slownesses and suppressed consideration of the changes in y and τ. Because of this, he could not distinguish between apparent structure caused by velocity changes and that caused by real geologic structure. In this section I would like to show how the migration velocity analysis methods I have discussed above can be specialized to yield a version of Toldi's stacking velocity algorithm that allows for a degree of structural variation, and does not attempt to assign all apparent structure in the data to lateral velocity variations. I will assume that the reflecting horizons are continuous and roughly horizontal, so that the assumptions behind using the NMO and stack operator for imaging are reasonably well satisfied. I will thus consider the movement of reflectors in τ, but not the changes in the lateral coordinate y for the reflecting point. Consideration of such weak structure might be important where lateral slowness variation could reverse the apparent dip direction seen on a time section, thus causing an "updip" well to be drilled on the wrong side of a producing well. Also, in permafrost, the near surface slowness can vary erratically, and it might be hard to estimate a reasonable background slowness model for an initial depth to time conversion or to know whether small structural features seen were believable.

Either the Eulerian or the Lagrangian viewpoints discussed above may be used in deriving an extended stacking velocity algorithm. The principle changes in either case are the same: G_Y is ignored, and the computation of G_W and G_T becomes easier. The original data space used will, of course, now be generated by stacking over a range of velocities rather than by prestack migrating. Once the Green functions are found, the algorithms outlined above for migration velocity analysis may be applied here with only a few obvious modifications.

Consider first a single reflector at depth z_r in a medium of constant slowness w. For a given midpoint y_r, the kinematics of the reflection would be given by the NMO equation

$$t = 2w \sqrt{{z_r}^2 + {h_r}^2} \tag{25}$$

The least squares perturbation analysis now tries to find a slowness W and a zero-offset time T for which an equation of the form

$$t = \sqrt{T^2 + 4W^2h^2} \tag{26}$$

best fits the travel time data points in a least-squares sense, rather than the double square root pyramid equation (5).

One now has two obvious ways to solve the least squares problem of finding W and T which fit a set of travel times. Toldi used a reparametrization in terms of squared variables, which converts the problem to one of fitting a straight line, the solution to which is found in any elementary textbook. He then perturbed the resulting equation for W to find out how ΔW depended on perturbations Δt_i. The same method can be applied to ΔT. In the migration case, I had to deal with the double square root pyramid equation which does not lend itself to a t^2–x^2 linearizing parametrization. Instead, I solved the problem by linearizing around an initial value; the set of equations equivalent to (6) is now

$$t_k \approx t(\hat{W}, \hat{T}) + \frac{\partial t}{\partial W} \Delta W + \frac{\partial t}{\partial T} \Delta T \tag{27}$$

Solving either least-squares system yields a solution of the form

$$\begin{bmatrix} \Delta W \\ \Delta T \end{bmatrix} = \begin{bmatrix} \frac{1}{D_1} \sum_k A_k \ \Delta t_k \\ \frac{1}{D_2} \sum_k B_k \ \Delta t_k \end{bmatrix} \tag{28}$$

Values for the coefficients depend on which least squares approach one uses; I will not derive them here. I do not believe that it is of any fundamental importance which choice of parametrization one uses. The expressions are simpler if the t^2–x^2 linearizing parametrization is used, and later equations then allow simplifying approximations such as Loinger (1983) and Toldi (1985) used.

Completing the derivation of the jacobian matrix follows the same argument as in the migration case, but considering now only those ray paths symmetric around the reflecting point. Referring to figure 1, rewrite equation (8) as

$$\Delta t_k = \int_{y_a} dy_a \int_{z_a} dz_a \, \Delta m\,(y_a, z_a) \left(\frac{\delta_1 + \delta_2}{\cos\theta_x} \right) \tag{29}$$

Following the same line of reasoning as in the migration case one gets

$$G_W(y_r, \tau_r, y_a, z_a) = \sum_{k=1}^{N_k} \frac{A_k}{D_1} \left(\frac{\delta_1 + \delta_2}{\cos\theta_x} \right) \tag{30}$$

and

$$G_T(y_d, \tau_d, y_a, z_a) = \sum_{k=1}^{N_k} \frac{B_k}{D_2} \left(\frac{\delta_1 + \delta_2}{\cos\theta_x} \right) \tag{31}$$

Final evaluation of the Green functions G_W and G_T in a form suitable for implementation involves substituting for the trigonometric terms in equations (30) and (31) and using the delta functions to eliminate the sum; I will not work through the details here.

The results of this section are not really new; similar derivations can be found in Loinger (1983) and in Toldi (1985). My purposes here were to show that stacking velocity techniques arise easily as special cases of the migration velocity analysis, and that apparent movement of the reflectors in τ can be included in the algorithm, thus perhaps making it possible to distinguish some structural variations from distortions of the reflectors caused by lateral velocity variation. Incorporation of the $\partial\tau/\partial m_a$ derivatives adds only one further degree of freedom to the problem, rather than two as in migration velocity analysis, thus simplifying the calculation of the jacobian matrix elements. However, it still raises the question of choosing either a Lagrangian or an Eulerian approach to decide where and how to evaluate the gradients of the objective function.

ACKNOWLEDGMENTS

I thank John Toldi for several enlightening discussions. My understanding of the outlook that I have here termed "Lagrangian" I owe to his insights and explanations. I also thank John Sherwood and Chuck Sword for the many hours of patient discussion they have shared with me about problems of velocity analysis and optimization.

REFERENCES

Aki, K., and Richards, P.G., 1980, Quantitative seismology: Freeman, San Francisco

Batchelor, G.K., 1967, An introduction to fluid dynamics: Cambridge Press, Cambridge

Claerbout, J.F., 1985, Imaging the Earth's interior: Blackwell, London

Fowler, P.J., 1985, Migration velocity analysis: linear theory: SEP-44

Loinger, E., 1983, A linear model for velocity anomalies: Geophysical Prospecting, 31, 98-118

Toldi, J.L., 1985, Velocity analysis without picking: Ph.D. dissertation, Stanford University

Date: Thu, 28 Nov 85 10:26:41 pst
From: Francis Muir <francis>
To: joe, jon, paul
Subject: BUGSY
Cc: francis

Yet let me flap this bug with guilded wings,
This painted child of dirt, that stinks and stings;
Whose buzz the witty and the fair annoys,
Yet wit ne'er tastes, and beauty ne'er enjoys.

O.K., guys, so you never made a mistake?

DEFINE

    z is thickness
    s is vertical slowness
    w is horizontal alacrity (vel.vel) of paraxial ellipse
    Q is anelliptic factor is 4.q.q - 4.q - 1 (q is my old q)

THEN

    z   is Sum(z)
    s.z   is Sum(s.z)
    w.s.z   is Sum(w.s.z)
    Q.w.w.s.z   is Sum(Q.w.w.s.z)

    which is QUITE different...

happy holidays!

francis

    postscript:
        what is Dave Hale REALLY talking about?