Stanford Exploration Project, Report 67, January 13, 1998, pages 1–55

Extending a CMP gather by optimization in velocity space

Jon F. Claerbout[1]

keywords: *velocity, algorithm*

ABSTRACT

A test of linear inversion theory yields weak and unconvincing extrapolations of a CMP gather.

INTRODUCTION

Most conventional data processing treats missing data the same as it treats zero-valued data. Improving this is a problem of long standing. Potentially the most rewarding circumstances for such research is when data contains strong events that can be extrapolated significantly beyond the bounds of measurement. The extended events need not be signal—they can be coherent noise such as multiple reflections. By extrapolating such events we sharpen the velocity spectrum, thus better outlining the velocity of all reflections.

A variety of optimization strategies have been proposed in the literature with none particularly established in practice. Here I test the idea of minimizing the relative power measured in velocity space. Unfortunately, the final result casts doubt on the formulation that follows below.

FORMULATON

Let $\mathbf{x}$ denote a data plane in time and offset. Let $\mathbf{v}$ denote a model plane in velocity and vertical time. Let $\mathbf{B}$ denote a modeling program. In this study I chose $\mathbf{B}'$ to be the pixel-precise data processing operator (Claerbout [Pixel..., 1990]). Thus the modeling operator is *defined* to be the transpose conjugate of a well-known processing operator. To a wave theorist this may seem strange, but the goal is to learn how to best handle truncation and aliasing so any reasonable approximation to the kinematics should be fine.

[1]**email:** not available

Linear inverse theory suggests an algorithm based on

$$\min_{v} \quad [(\mathbf{x}-\mathbf{B}\mathbf{v})'\mathbf{W}_{xx}(\mathbf{x}-\mathbf{B}\mathbf{v}) + \lambda \mathbf{v}'\mathbf{W}_{vv}\mathbf{v}] \tag{1}$$

where $\mathbf{W}_{xx}$ and $\mathbf{W}_{vv}$ are the usual inverse covariance weighting functions and λ is a parameter that is presumably explained by the textbook literature (though SEP-65 page 214 shows some pitfalls there.) We'll consider weights later. For the unweighted problem the formal solution is a power series in $\mathbf{B}\mathbf{B}'$. In real life, the operator $\mathbf{B}$ is expensive, especially considering the huge volume of data requiring analysis. I chose to consider only the first two terms of the power series for careful examination to identify the most significant practical effect,

$$\mathbf{v} \quad = \quad (\beta \mathbf{B}' + \alpha \mathbf{B}'\mathbf{B}\mathbf{B}')\mathbf{x} \tag{2}$$

where α and β have yet to be chosen. In the formula (2) the data is taken to be separable into two parts, the known k and the missing m which is taken to be zero, say

$$\mathbf{x} \quad = \quad (\mathbf{x}_k, \mathbf{x}_m) \quad = \quad (\mathbf{x}_k, 0) \tag{3}$$

MOVIE OF NONSTATIONARITY

Now we face the question, what are the best values of α and β to use in equation (2)? There is a well known least-squares answer, but I have been frustrated frequently enough, by lack of consideration to weighting functions, non-stationarity, and such that I chose a more careful investigation. To gain insight in the matter I prepared a movie sweeping through various values of α/β. I'll be excruciatingly explicit about what I computed. Say β is unity although it was really determined by the plotting program. For the missing traces define $d\mathbf{x} = \mathbf{x}_m$ where in conventional processing, $d\mathbf{x} = 0$.

$$\mathbf{v}_k \quad = \quad \mathbf{B}'\mathbf{x}_k \tag{4}$$

$$d\mathbf{x} \quad = \quad \mathbf{M}\mathbf{B}\mathbf{v}_k \tag{5}$$

$$d\mathbf{v} \quad = \quad \mathbf{B}'d\mathbf{x} \tag{6}$$

where $\mathbf{M}$ is a diagonal matrix containing ones in locations of missing data and zeros in locations of known data. I studied side-by-side panels of

$$(\mathbf{x}_k + \alpha\, d\mathbf{x}) \quad \text{and} \quad (\mathbf{v}_k + \alpha\, d\mathbf{v})$$

as a movie for various values of α.

For data I began with a CMP gather used by Dave Hale. Denoting missing by m and known by k I interleaved and extended the data offsets by

$$m, m, m, m, m, m, k, m, k, m, k, \cdots, k, m, m, m, m, m, m$$

The movie illustrated that a concept familiar in Fourier analysis also applies to the velocity transform. This is the idea that roughness in the *interior* of the data corresponds to energy "off ends" in velocity space whereas energy "off ends" in offset space, controls resolution in the *interior* in velocity space.

The movie also illustrated that a subjective choice of the best value of α was a function of location in the (x,t)-plane. The best choices of α for internal regions and for external regions were quite different. Internally the best choice was one that would make interleaved traces look most like their neighbors which simply meant that they would be the same polarity and amplitude. Within the data the best fitting α was about $\alpha = .4$ near the top and $\alpha = .2$ near the bottom. Off the sides of the data, a subjective analysis was not so simple, but it seemed clear to me that the best value of α needed to be much larger, say about $\alpha = 2.0$.

I was prepared to see a spectral difference between $\mathbf{x}$ and $d\mathbf{x}$, but none was apparent, indicating the rho filters in the definition of $\mathbf{B}$ had generally done their job.

The internal results can be explained by less than perfect quality of the operator $\mathbf{B}$. Clearly a more stationary value of α in the internal region could be achieved by a definition of the velocity transformation that put more effort to yielding a more unitary operator. The external behavior is more troublesome, and potentially far more interesting because it controls the resolution in velocity space.

From the results of this movie study I decided to separate the issues of trace interpolation from those of trace extrapolation though obviously there is some interaction near the ends of the cable. Of the two problems I chose the data extrapolation problem to have the most interest because it affects resolution of the geologically interesting part of our data. Since the $\mathbf{B}'$ operator chosen for this study works best at wide offsets, I chose to extend the far end of the cable.

TRACE EXTRAPOLATION

The velocity transform $\mathbf{B}$ comes close to an invertible operator because Radon transforms are theoretically invertible. The velocity transform is a Radon transform with hyperbolic distortion. I think the main invertibility problem stems from numerical approximations in my definition of $\mathbf{B}$. Unlike $\mathbf{B}$, operators more typically found in inversion formulations like (2) are both overdetermined and unconstrained. In consequence, I believe we can regard the missing data as the free variables instead of regarding the velocity space $\mathbf{v}$ as the free variables. I'd like to regard the transformation of the complete data space $\mathbf{x}$ to the velocity space $\mathbf{v}$ as nearly unitary and noiseless by comparison to that transformation with the truncated data space. Thus in the formulation (2), the first term could be identically zero. Whether this is valid, it is my presumption for the rest of this paper. I presume the problem to be solved is the minimization of the right-hand term in (2). (As we'll see the validity of this assumption is not supported by the field-data trials.)

Happiness is noticing that a problem is much simpler than you had expected. I suddenly realized that the heart of the problem is contained in a much smaller operator than I had thought. Notice that it is computationally easy to decompose integral operators into parts, in this case, into the part of known data and missing data. That is to say, velocity space is

$$\begin{aligned} \mathbf{v} &= \mathbf{B}'(\mathbf{x}_k + \mathbf{x}_m) \\ &= \mathbf{B}_k{}'\mathbf{x}_k + \mathbf{B}'{}_m\mathbf{x}_m \\ &= \bar{\mathbf{v}} + \mathbf{B}'{}_m\mathbf{x}_m \end{aligned}$$

The big operator is the one involving the known data $\mathbf{B}'{}_k$ and the little operator is the one involving the few missing traces we will add at the end of the gather $\mathbf{B}'{}_m$. Both transform to the same plane in velocity space. Thus the regression

$$\mathbf{0} \quad \approx \quad \mathbf{v} \quad = \quad \bar{\mathbf{v}} + \mathbf{B}'{}_m\mathbf{x}_m \tag{7}$$

involves the "small" operator and because it is small, it is reasonable to use many iterations if that should turn out to be helpful.

The idea of minimizing power in image space (velocity space) may be questionable, but makes more intuitive sense when you include a weighting function. Ideally the weighting function would be the inverse of the envelope of the expectation of the variance of the velocity space. In practice we don't know what that is, so I used the envelope of the first estimated velocity space, i.e. that determined from zero padded data. As is common practice I smoothed the envelope and added a constant and experimented with various sized constants.

In practice it happened that few iterations were needed. Convergence appeared to happen on the *first* iteration.

DISCUSSION AND CONCLUSION

Results were disappointing. I thought of switching to synthetic data but instead switched to a high quality data set (Yilmaz and Cumro number 27). The quality of the results remained poor. The data extensions found in figure 1 do not look as expected. The stepouts are only roughly appropriate. Further, they are numerically too small to be easily visible, so the amplitudes in the display were replaced by the signed square roots.

Is there a program bug or a conceptual bug? I don't know. (This is a progress report.) The theory is given above and the program listing in an appendix. Various pitfalls and nagging doubts are summarized below.

- The zero-offset truncations look more severe than the wide offset and maybe they should have been handled first.

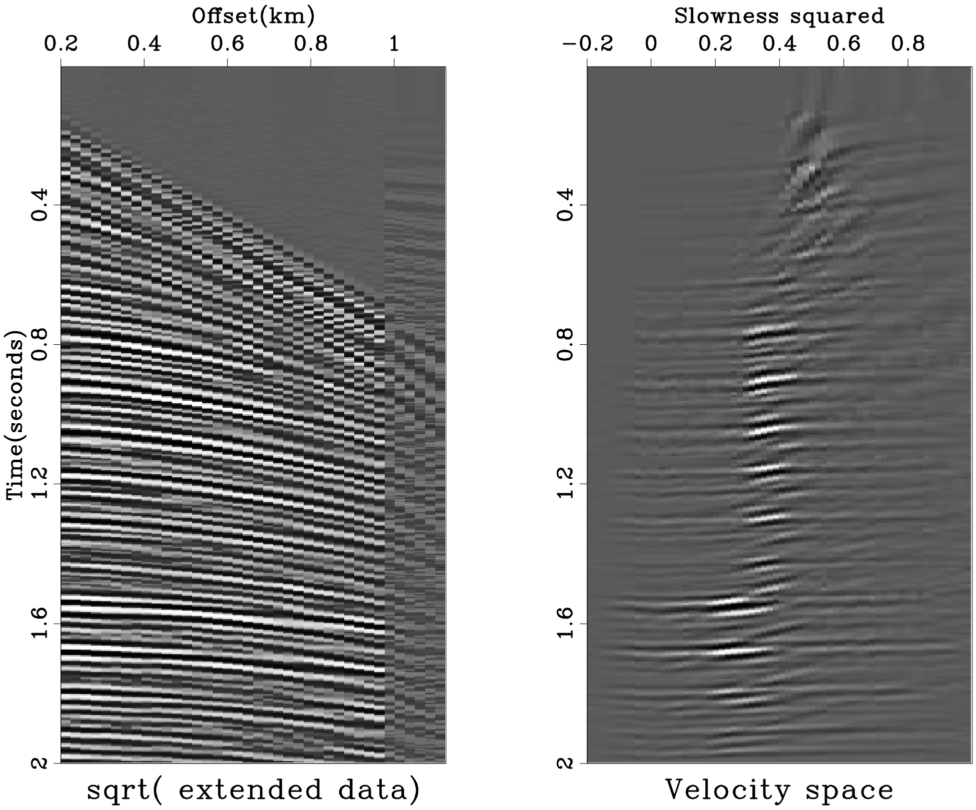

Figure 1: Attempt to extrapolate a data set. jon1-myplot [NR]

- The simplifications arising from taking recorded data to be noiseless could be reexamined.
- The presumption that we could ignore the lead term in (2) should be reexamined.
- Notice that the trace closest to the known data is smaller than the others. This can be understood as follows: If a trace can extinguish its neighbor by being equal to it but of opposite polarity, then in a power minimization this will happen. Theoretically, however it should not happen because two nearby traces of opposite polarity imply signal in velocity space at slownesses beyond practical interest which unfortunately are likely to be omitted from the computation. (This is an argument for parameterizing with slowness instead of slowness squared.)
- The weighting function should be different.
- The theoretical approach could follow that of Bill Harlan.
- The theoretical approach could follow some entropy theory. For example extremalizing some quadratic form like the ratio of the terms in 2 instead of the sum would not have solutions biased towards zero because the ratio is scale invariant.
- Possible program bug.

REFERENCES

Claerbout, J., 1990, Hyperbola tricks: SEP–**65**, 241-246.

Claerbout, J., 1990, Model fitting by least squares: SEP–**65**, 221-228.

Claerbout, J., 1987, The velocity transform and its preconditioning: SEP–**56**, 137.

Claerbout, J., 1980, Restoration of missing data by least squares optimization: SEP–**25**, 1-15.

Claerbout, J., 1980, Data restoration with parsimonious models: SEP–**25**, 23-26.

Thorson, J.R., and Claerbout, J.F., 1985, Velocity and slant stack stochastic inversion: Geophysics **v** 50, 2727-2741.

Thorson, J.R., 1984, Velocity stack and slant stack inversion methods: SEP–**39**

APPENDIX. PROGRAM LISTING

The program listing is

```
# Extend a few traces off the end of a CDP gather using the vspray operator.
#
#       <in.H Test np=6 eps=1. vspace=vspace > out.H
#       np is the number of padded traces off end, nx+1,...,nx+np
#       vspace is a space of slowness squared.
#
# keyword missing velocity
# JFC
#%
integer nt, n2, n3, esize, nx, nv, np, nxp, niter
from history:   integer n1:nt, n2, n3, esize
                        nx=n2
from par:       integer nv=2*nx,  np=6, niter=2
nxp = nx + np
allocate:       real xx(nt, nxp), vv(nt, nv), pp(nt, np)

subroutine dumb( niter, nt, nx, np, nxp, nv, xx, vv, pp)
integer it, nt, ix, nx, iv, nv, ip, np, nxp, iter, niter
integer rect1, rect2
real    xx(nt, nxp), vv(nt, nv), pp(nt, np)
real    t0, dt, x0, dx, s0, ds, p0, dp, sum, eps, smax
temporary real            dpp(nt,np), spp(nt,np)
temporary real  rr(nt,nv), dr (nt,nv), sr (nt,nv)
temporary real  ww(nt,nv)
temporary real  wr(nt,nv)
temporary real  rrr(nt,nv)
from history:   real o1:t0, d1:dt
from history:   real o2:x0, d2:dx
from par:       real eps=1.
call sreed('in', xx, 4 * nt * nx)
call hclose()
smax= 1./ 1.0 **2
s0 = -smax / 5.
ds = (smax-s0)  / (nv-1.)
dp = dx
p0 = x0 + nx * dx
do it=1,nt  { do ip=1,np  { xx(it,nx+ip) = 0. }}
call vspray( 0,  nt,dt,t0,  nx,dx,x0, xx,  nv,ds,s0, vv)

do it=1,nt  { do iv=1,nv  { ww(it,iv) = abs( vv(it,iv)) }}
rect1 = nt/10
rect2 = 3
call triangle2( rect1, rect2, nt,nv, ww, ww)
sum = 0.
do it=1,nt  { do iv=1,nv  { sum = sum + ww(it,iv) }}
do it=1,nt  { do iv=1,nv  { ww(it,iv) = 1. / (ww(it,iv)+eps*sum/(nt*nv)) }}

do it= 1, nt {
        do iv= 1, nv
```

```
                rr(it,iv) = - vv(it,iv)
        do ip= 1, np
                pp(it,ip) = 0.                  # padded data initialize
        }

do it=1,nt  { do iv=1,nv  { wr(it,iv) = rr(it,iv) * ww(it,iv) }}

do iter= 0, niter {
        do it=1,nt  { do iv=1,nv  { rrr(it,iv) = wr(it,iv) / ww(it,iv) }}
        do it=1,nt  { do iv=1,nv  { dr(it,iv) = wr(it,iv) * ww(it,iv) }}
        call vspray( 1,  nt,dt,t0,  np,dp,p0, dpp,  nv,ds,s0, dr)
        call vspray( 0,  nt,dt,t0,  np,dp,p0, dpp,  nv,ds,s0, dr)
        do it=1,nt  { do iv=1,nv  { dr(it,iv) = dr(it,iv) * ww(it,iv) }}
        call cgstep( iter, nt*np, pp, dpp, spp,
                           nt*nv, wr, dr,  sr  )
        do it= 1, nt
                do ip= 1, np
                        xx(it,nx+ip) = pp(it,ip)
        }
call srite('out',    xx, 4 * nt * nxp)
call srite('vspace',rrr, 4 * nt *nv)
return; end
```

The subroutines appear in previous sep reports and perhaps in this one. The interactive report (if any) should have pushbuttons to bring up the relevant library subroutines here.