MINIMUM INFORMATION DECONVOLUTION

Jon F. Claerbout

What is the *simplest* earth model consistent with a given geophysical data set? The conventional definition of *simple* is implied by the minimization of power in a filtered earth model. Our present approach will instead be to define a *simple* model as one containing *little information*. Information will then be defined in either of two ways: the first is by counting bits, much as a computer memory size is reckoned; the second is like Shannon's definition of entropy, the expected logarithm of inverse probability. We will see that both of these measures of information emerge directly from the *geometric inequality*.

Definitions of information

In simplest form the geometric inequality relates the arithmetic mean of two positive numbers u_1 and u_2 to their geometric mean, namely

$$\frac{1}{2} u_1 + \frac{1}{2} u_2 \quad \geq \quad u_1^{1/2} \, u_2^{1/2} \tag{1}$$

The inequality becomes an equality if and only if $u_1 = u_2$. Repeated subdivision of u_1 and u_2 and application of (1) to the subdivisions yields a more general inequality:

$$\frac{1}{N} \sum_{i=1}^{N} u_i \quad \geq \quad \prod_{i=1}^{N} u_i^{1/N} \qquad \text{where} \quad u_i \geq 0 \tag{2}$$

Likewise, in this more general inequality (2), we get equality if and only if all the u_i are identical. The closeness of the u_i to one another can be measured by the closeness of the inequality to an equality. To quantify this we could form the ratio of the product in (2) to the sum in (2). Clearly

such a ratio approaches unity as the u_i become identical to one another, and the ratio approaches zero as the u_i become more dissimilar.

A slightly more convenient measure is the logarithm of the ratio, namely S, where

$$S = \ln \prod_{i=1}^{N} u_i^{1/N} - \ln \frac{1}{N} \sum_{i=1}^{N} u_i$$

$$S = \sum \ln u_i^{1/N} - \ln \frac{1}{N} \sum u_i$$

$$S = \frac{1}{N} \sum_{i=1}^{N} \ln u_i - \ln \frac{1}{N} \sum_{i=1}^{N} u_i \tag{3}$$

By this definition S, being the logarithm of a number less than unity, will always be negative. Maximizing S will drive it toward zero and the u_i toward homogeneity. Minimizing S will drive it toward minus infinity and the u_i apart from one another.

The first application which we will consider will be termed *bit count deconvolution*. Let the positive numbers u_t, to be used in (3), be defined to be instantaneous power or the envelope of a deconvolved seismogram x_t. Instantaneous power is defined by simply squaring x_t, say

$$u_t = x_t^2 \tag{4a}$$

Envelope is defined by first constructing an imaginary part for x_t by the Hilbert transform of the real part. Envelope u_t is then defined by

$$u_t = \overline{x}_t \, x_t \tag{4b}$$

A deconvolved seismogram x_t is defined by application of a deconvolution filter a_t to an observed seismogram y_t. Basically, it will be the deconvolution filter a_t that will be adjusted to find a minimum for S.

In most applications there will be many seismograms with a single deconvolution filter to be used on each of them. The seismograms could be concatenated into a single y_t vector with a suitable number of zeros

separating them in such applications. In single channel applications the filter a_t will be constrained to have fewer adjustable parameters than the data channel has independent values.

An integer $K \leq 2^L$ can be encoded in $\log_2 K = L$ bits of computer memory. Likewise, K^2 is stored in $2L$ bits of computer memory. It is an inconsequential choice of scale that S in Equation (3) is written in terms of natural logarithms rather than binary logarithms. So the first logarithm term in (3) can be thought of as the scaled information content of the seismogram envelope. The second term in (3) is the scaled seismogram energy. We may interpret minimization of S to be minimization of the amount of *bit count information* in u_t, subject to keeping the total energy constant.

A second definition of information, which is related to probability theory, will emerge from the geometric inequality if we make a slightly different definition of u_i. Let the seismogram x_t be sorted. That is, by reordering of the time points we may define $x_{(t)}$ from x_t such that

$$x_{(t)} \leq x_{(t+1)} \qquad \text{all } (t) \tag{5}$$

Now we will define u_t by

$$u_t = x_{(t+1)} - x_{(t)} \tag{6}$$

The inequality (5) assures that the u_t defined by (6) will be positive as required by the geometric inequality. [The probability that any $u_t = 0$ could be reduced by defining $u_t = x_{(t+M)} - x_{(t-M)}$ for some smoothing parameter M.] Where the $x_{(t)}$ are close together the probability of the value x is high, and if the $x_{(t)}$ are wide apart the opposite is true. More precisely, if $p(x)\,dx$ is the probability that the value x_t lies between x and $x + dx$, then $1/(Nu_t)$ is an estimate $\hat{p}$ of the probability density p in the interval between $x_{(t+1)}$ and $x_{(t)}$. Note that $\Sigma\, \hat{p}\, \Delta x = 1$.

Thus we may think of (3) as

$$S = \sum_{\text{data}} \ln\left(\frac{1}{\hat{p}}\right) - \ln\left[x_{(\max)} - x_{(\min)}\right] \tag{7}$$

But a sum over data estimates an expectation, which in turn is an integral over probability. So (7) is an estimate of

$$S_{\text{theoretical}} = \int p(x) \ln \left[\frac{1}{p(x)}\right] dx - \ln \left[x_{(\max)} - x_{(\min)}\right] \qquad (8)$$

which, but for the log of the range, is just Shannon's well-known definition of entropy or expected information.

As before, maximizing S will be smoothing the probability function, and minimizing S will cause the u_t to become dissimilar. Dissimilar u_t means that the probability function has become peaked and/or multimodal. It is attractive to speculate that one of the basic aspects of well logs, the "boxy" character, associated with alternation between sand, shale and carbonates, say trimodality, could be imposed upon deconvolved seismograms by minimization of this Shannon-like form (7) of the geometric inequality. I have not yet tested this possibility. Up to the present I have been trying to reconstruct from seismograms the reflectivity function rather than (the logarithm of) the impedance function. In this application the Shannon information definition has not yet performed any better than the bit count definition. The remainder of this paper will confine itself to further interpretation and details of implementation of *bit count deconvolution.*

In either case, S(u), defined by Equation (3), deserves a name. We have seen that the first term can be interpreted as *information* either under the definition of u in Equation (4) or the definition in Equation (6). In either case, the second term in (3) comes from the normalization. I propose to call S the *information density.* It is information per unit power in the case of (4) and information per unit probability in the case of (6).

Descent algorithm

To test these optimization concepts a descent algorithm was devised. Besides providing a mechanism for testing basic concepts, the descent algorithm also provides further insight. In particular, it turns out that the force exerted on the solution by each data point is independent of the smoothed envelope of the seismogram. This is a practical convenience because it means the deconvolution may be done before or after a slow gain readjustment. This

remarkable gain insensitivity is not shared by conventional processing such as least squares filter design. (Gain insensitivity could then be achieved by *post facto* introduction of weighting functions.) The natural gain insensitivity of minimum information deconvolution shows that our mathematical definition of information conforms to our common sense ideas of information.

Let us form some gradients. Take the derivative of the information density (3) with respect to u_j:

$$\frac{\partial S}{\partial u_j} = \frac{1}{N u_j} - \frac{1}{\sum_{i=1}^{N} u_i} = \frac{1}{N u_j} - \text{const} \tag{9}$$

Next define the *unconstrained gradient* g_t as the derivative of the information density with respect to the conjugate waveform $\overline{x}_t$:

$$g_t = \frac{\partial S}{\partial \overline{x}_t} = \sum_j \frac{\partial S}{\partial u_j} \frac{\partial u_j}{\partial \overline{x}_t} = \frac{\partial S}{\partial u_t} x_t$$

$$g_t = \left(\frac{1}{N u_t} - \text{const} \right) x_t \tag{10}$$

Recall now that the observed seismogram is y_t, the deconvolution filter is a_t, and the deconvolved seismogram is x_t, which in this case is the earth model. Letting $*$ denote convolution,

$$y * a = x \tag{11a}$$

or

$$y * (a + da) = x + dx \tag{11b}$$

or

$$y * da = dx \tag{11c}$$

The basic procedure is iterative. From some starting filter a, try to choose da so that dx is in the opposite direction of the unconstrained

gradient g_t given by (10). For convenience da may be chosen to solve the overdetermined system

$$dx = y * da \approx -\lambda g \tag{12}$$

where λ is some scalar which has yet to be chosen. The solution to (12) is given in the usual ways (such as Levinson recursion or smoothing followed by frequency domain division). It may be roughly indicated by

$$da = -\lambda \frac{\langle \bar{y}\, g \rangle}{\langle \bar{y}\, y \rangle} \tag{13}$$

It is on the basis of the numerator of (13) that assertions about gain insensitivity have been made. When equilibrium is attained it will be because of the vanishing of the crosscorrelation of the observed seismogram y_t with the unconstrained gradient g_t. Looking back to Equation (10) we see that g is composed of two parts. Use of (12) with only the right-hand term in the gradient (10), namely

$$dx_t \approx \lambda \text{ const } x_t$$

shows that the effect of this term is merely to provide a rescaling of the deconvolved seismogram x_t. The interesting part of (10) is the left-hand term $1/u_t$, which is the inverse of the envelope of the deconvolved seismogram x_t. To the extent that the filter a_t has a short memory function, it can be said that the envelope of the output x_t is (but for a scale) the same as the envelope of the input y_t. Combining these ideas we see that in the calculation of $\langle \bar{y}\, g \rangle$ the envelope of the data seismogram y_t is compensated for by the inverse envelope in the unconstrained gradient g_t. Thus we see the qualitative result that minimum entropy deconvolution is insensitive to slow gain adjustment on the original data.

Achieving convergence

In a non-linear optimization problem like this, it is not easy to prescribe solution methods which are reasonably economical and still guaranteed to work for all sensible data sets. All that I have really done so far is to demonstrate convergence with some single channel synthetic data, where the starting filter is very far from the final filter. Presumably, with field

data the descent will be stabilized by the presence of more channels, and a closer starting filter will be known. On the other hand, the environment will be sufficiently different that the program presented here may be far from optimum and may even require some changes (besides packing many data channels into one vector).

The first parameter choice is for λ, the amount of dx to add to x. A sensible choice is to relate λ to the *relative* perturbations in the output x_t. That is, λ may be chosen to prescribe some ratio

$$\frac{||dx||}{||x||} = \frac{\max_t (dx_t)}{\max_t (x_t)} \quad \text{or} \quad \left[\frac{\sum dx_t^2}{\sum x_t^2}\right]^{1/2} \tag{14}$$

Typically I begin by choosing the ratio at 30 percent and upon successive iterations decreasing it exponentially until the resolving power of our plotting machine is reached. Naturally, if descent is made in a too small number of steps the lowest minimum is not obtained. A good suggestion by Will Gray has not yet been tested. It is to develop S in a Taylor Series in λ up to the quadratic term. Then the minimum value of S as a function of λ is given by the choice $\lambda_{min} = -S'/S''$.

Another parameter choice in the algorithm is associated with the desirability of doing some smoothing of the envelope. To show this, Figure 1 is a plot of a component of the unconstrained gradient g_t as a function of the same component of the deconvolved seismogram.

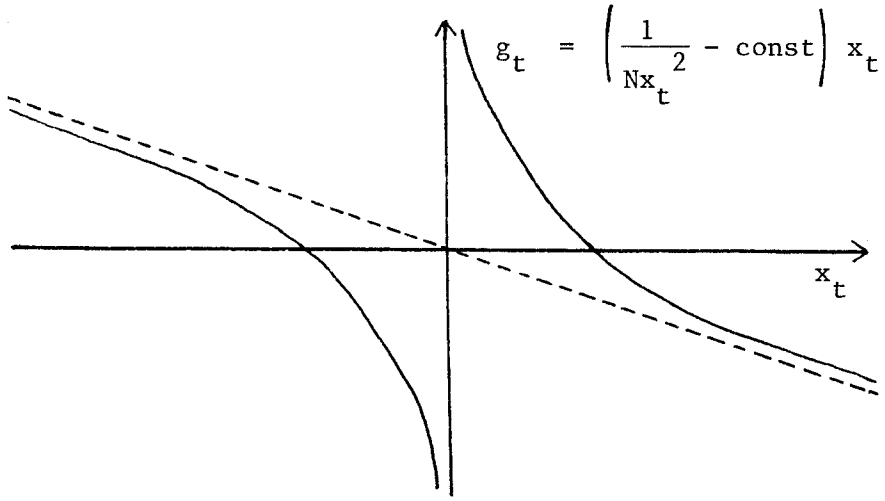

FIGURE 1.--Plot of equation (10) with (4a)

A nice thing about this gradient is that for large data values it becomes linear. This means that a pulse will exert the same force on the solution whether it is exactly on a mesh point, say δ_t, or if instead it is divided between two mesh points, say $(\delta_t + \delta_{t+1})/2$.

It is clear that the gradient becomes very strong when the data becomes very small. In fact, simple zero crossings of the seismogram would seem to present a problem. The problem is considerably reduced if the envelope definition $u = \overline{x}\, x$ is used instead of $u = x^2$. This does not totally eliminate the problem. Consider, for example, packing many seismograms into one vector with zeros inbetween the seismograms. Clearly the whole method considers them to be about as important as any other data points. This indicates something of a contradiction in our problem formulation. On the one hand we believe information content to be independent of data magnitude, on the other hand it is convenient to suppose that data less than some threshold magnitude $\overline{v}$ is noise and may be ignored. It is easy to incorporate $\overline{v}$ into the analysis by the definition $u = \overline{x}\, x + \overline{v}$ or preserving the scale invariance of S with x by taking

$$u_t = \overline{x}_t\, x_t + \frac{\overline{v}}{N} \sum_1^N \overline{x}_t\, x_t \tag{15}$$

But this unfortunately introduces another parameter $\overline{v}$ which must then be chosen and can be mischosen. Additionally, I have found it advantageous to choose $\overline{v}$ large at the beginning and decrease it a bit with each iteration.

Experience has led me to one more stabilizing parameter. The numerator crosscorrelation in (13), namely $\langle \overline{y}\, g \rangle$ occasionally misbehaves when large step sizes λ are chosen at the beginning of the descent. In practice this should be stabilized by the existence of more channels and the better starting solution. What I did to stabilize the synthetic was to taper the crosscorrelation at early stages of the descent and then reduce the amount of taper as the solution was approached.

The computer program

The computer program which follows is structurally like the one in SEP-13, page 24. It uses the same subroutines which are found there. They are not

repeated here. On line 30 in function GRAD this program has arbitrarily defined u as the average of Equations (4a) and (4b). The numerical values of S computed by this program contain little information about the success of the descent since the parameter $\bar{v}$ is being changed as the iteration proceeds.

bitcount.f

```
      real ryy(128),ryg(128)
      real x(256),g(256),dx(256),y(128),b(25),a(128),da(128)
      call setmod(11,128)
      nb=23
      ny=128
      na=101
      nam=(2*na)/3
      nx=ny+na-1
      amb=1./200.
      call job(x,ny,y,nb,b,na,nam,a,amb,rmssn)
      niter=16
      do 400 iter=1,niter
      alpha=.3*.10**((iter-1.)/niter)
      call conv(ny,y,na,a,x)
      elog=grad(alpha,nx,x,g)
      call scale(-1.,nx,g)
      do 80 i=1,na
      ryy(i)=dot(ny-i+1,y,y(i))
80    ryg(i)=dot(ny,y,g(i))
      do 90 i=1,na
90    ryg(i)=ryg(i)/(1.+(abs(i-nam+.1)*niter/(nam*iter+.1))**2)
      call fit(na,ryy,ryg,da)
      nskip=1
      if(mod(iter-1,nskip).ne.0)go to 150
      idy=2
      ink=1+(iter-1)/nskip
      call plot(ink,idy,100,nx,x)
      call plot(ink,idy,120+idy*nx,na,a)
      call plot(ink,idy,140+idy*(nx+na),na,da)
      call plot(ink,idy,160+idy*(nx+na+na),nx,g)
150   continue
      ix=(iter-1)*70/nskip+70
      iy=0
      call number(elog,'(f10.6)',ix,iy,2,1,9)
      call conv(ny,y,na,da,dx)
      sc=dot(nx,x,x)/dot(nx,dx,dx)
      sc=sqrt(sc)
      do 175 i=1,na
175   a(i)=a(i)+da(i)*alpha*sc
      call plot(ink,idy,180+idy*2*(nx+na),nx,dx)
400   continue
      stop
      end
      subroutine job(x,ny,y,nb,b,na,nam,a,amb,rmssn)
      dimension x(ny),b(nb),y(ny),a(na)
      read(17,17)(b(i),i=1,nb)
17    format(f5.1)
      nx=ny-nb+1
      do 10 i=1,nx
10    x(i)=0.
      x(1)=1.01
      x(nx-1)=.5
```

bitcount.f

```
      x(nx)=-.5
      rock=0.
      write(11,77) rock
77    format('preconceived best answer =',f20.8)
      call conv(nx,x,nb,b,y)
      call scale(1./bigest(ny,y),ny,y)
      rmssn=sqrt(dot(ny,y,y)/ny)/amb
      do 38 i=1,ny
38    y(i)=y(i)+2.*(ran(i1,i2)-.5)*amb
      write(11,78)amb,rmssn
78    format('ambient=',f9.6,'      rms s/n=',f7.1)
      do 40 i=1,na
40    a(i)=0.
      a(nam)=1.
      return
      end
      subroutine plot(iter,idy,ishift,n,p)
      dimension p(n)
      logical*1 m,d,e
      data m,d,e/'m','d','e'/
      call setmod(9,512)
      b=0.
      do 10 i=1,n
10    if(abs(p(i)).gt.b)b=abs(p(i))
      do 20 i=1,n
      iy=ishift+idy*i
      ix=iter*70+p(i)*45./b+.5
      if(i.eq.1)write(9)m,ix,iy
      write(9)d,ix,iy
      iy=iy+idy
      write(9)d,ix,iy
20    continue
      return
      end
      function grad(alpha,n,x,g)
      dimension x(n),g(n)
      complex cx(256),conjg
      n256=256
      do 5 i=1,n256
5     cx(i)=0.
      do 10 i=1,n
10    cx(i)=x(i)
      call fft(n256,cx,+1.,1.)
      nh=n256/2
      do 20 i=2,nh
      cx(i)=2*cx(i)
20    cx(i+nh)=0.
      call fft(n256,cx,-1.,1./n256)
      do 30 i=1,n
30    g(i)=cx(i)*conjg(cx(i))+x(i)*x(i)
      thresh=alpha*bigest(n,g)
      do 40 i=1,n
```

$u = \bar{X}X + X^2$

bitcount.f

```
40      g(i)=x(i)/(g(i)+thresh)
        proj=dot(n,g,x)/dot(n,x,x)
        do 50 i=1,n
50      g(i)=g(i)-proj*x(i)
        sum=0.
        prod=0.
        do 60 i=1,n
        u=cx(i)*conjg(cx(i))
        sum=sum+u
60      prod=prod+alog(n*u)
        grad=exp(prod/n)/sum
        return
        end
```

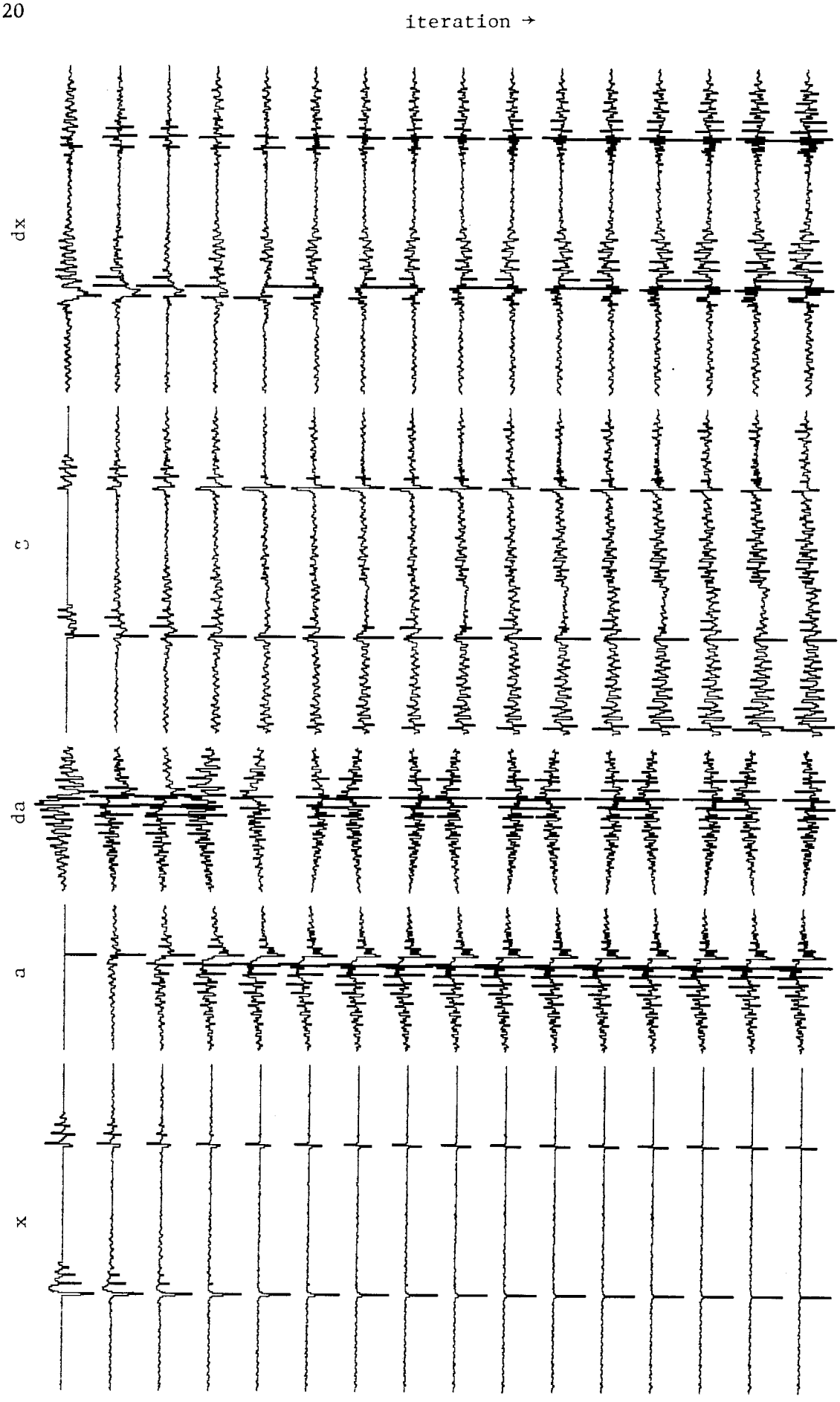

FIGURE 2.--Results of computer test program. The convolution a*x is the same at each iteration.

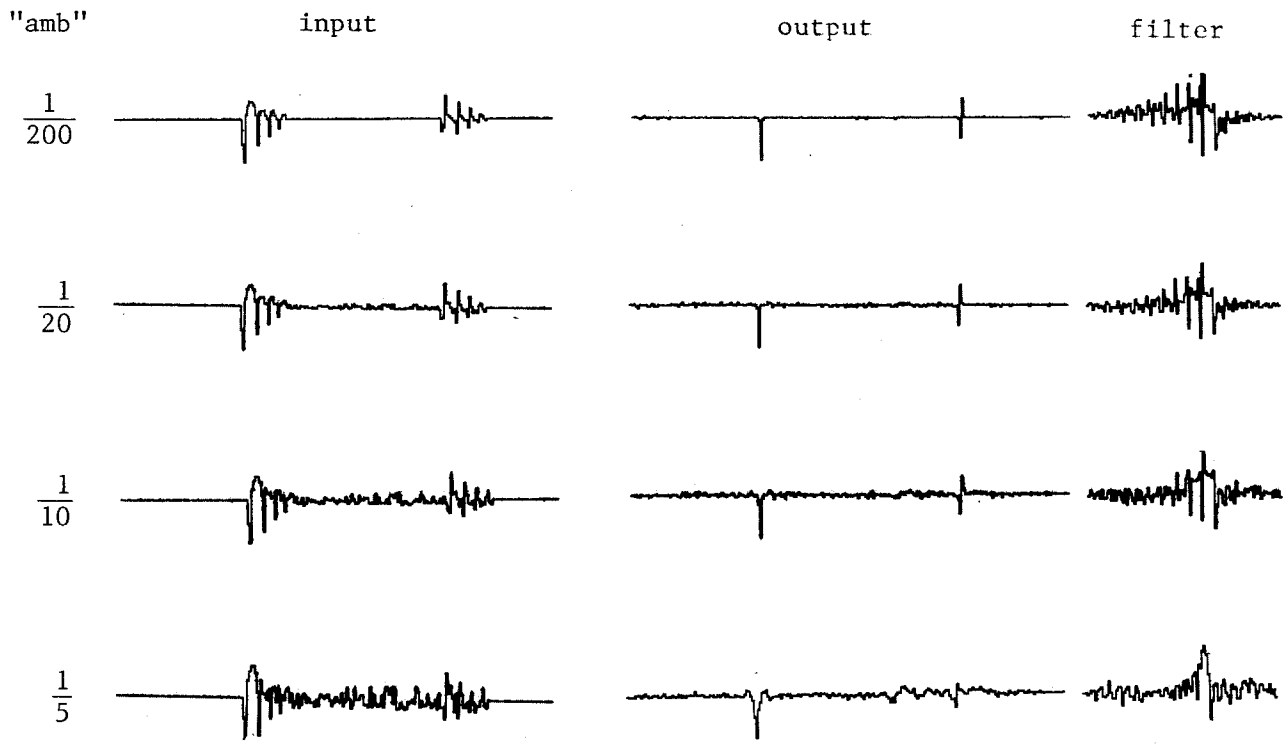

FIGURE 3.--Increasing the additive noise causes reasonable decrease in quality of outputs. The same pseudo-random noise was used in each case, but it was scaled by the parameter "amb."

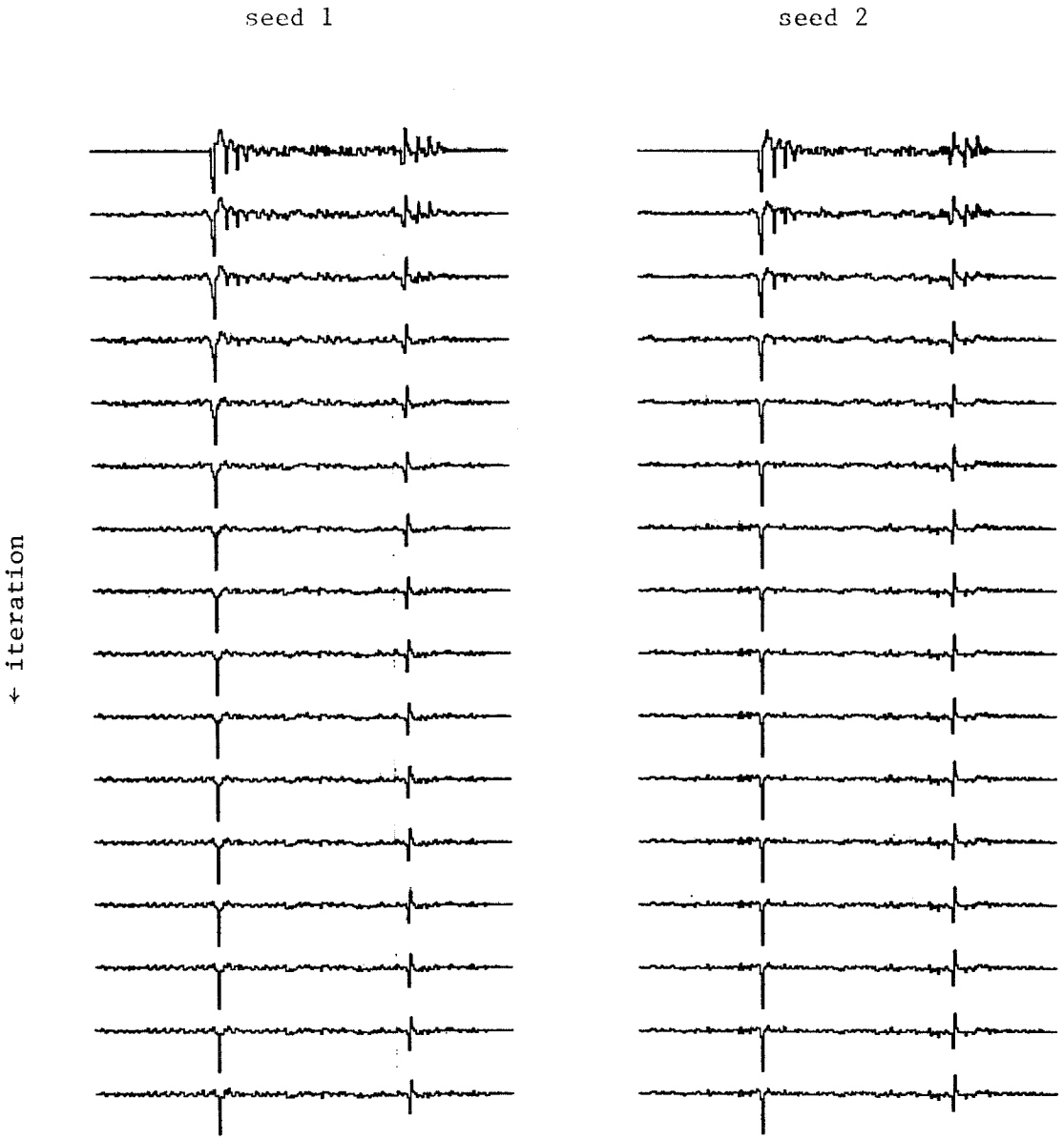

FIGURE 4.--Deconvolved trace as a function of iteration for two different random number generator seeds.