Q and Adaptive Prediction Error Filters

Dave Hale

Abstract

We first specify an auto-regressive model for a seismic trace which includes the effects of attenuation and spherical divergence. From this model, we determine how an adaptive prediction error filter (APEF) should change with time as it "deconvolves" our model, non-stationary seismogram.

Most APEF algorithms developed to cope with non-stationarity are not based on a model of the physical processes which cause the non-stationarity. We applied two such algorithms to our model seismogram, attempting not only to deconvolve the data, but also to estimate Q from changes in the APEF with time. Both algorithms yielded reasonable deconvolved traces along with rough estimates of Q, although one algorithm performed significantly better than the other.

We also derive an APEF algorithm based on (in fact, constrained by) our model of attenuation. When applied to the model seismogram, this algorithm produced a better deconvolved trace and a more accurate estimate of Q than either of the two unconstrained algorithms; it is, however, limited by its inability to adapt to the time-varying signal-to-noise ratio present in our model (and in real seismic data as well).

Introduction

Adaptive, time-varying deconvolution filters provide a means of deconvolving non-stationary seismograms. A variety of adaptive algorithms are available, and the choice of a particular algorithm is typically based on our knowledge of the physical processes which produced the seismogram. For example, if parameters describing the source waveform, attenuation, spherical divergence, and noise are known, then Kalman filtering, a time-varying algorithm which can use these known parameters, may be appropriate. (See "Q and Kalman filtering" elsewhere in this report.) If, on the other hand, such parameters are unknown, then (armed with assumptions of an auto-regressive model and a white reflectivity series) we might choose adaptive prediction error filtering (APEF).

In this paper, we assume that APEF is our chosen method and that attenuation effects at least partially account for the non-stationarity which motivates our use of APEF. We then explore the following possibilities:

(1) We use an APEF which has no a priori knowledge of how it should adapt; information is then contained in how the filter does adapt. Specifically, we may estimate Q, the quality factor of the subsurface which we believe parameterizes attenuation, from changes in the PEF coefficients with time.

(2) We use an APEF which is constrained to adapt only to compensate for attenuation effects. We again obtain an estimate of Q.

With respect to the problems of producing a "good" deconvolved trace and estimating Q, each method has its advantages as well as its disadvantages; and these are discussed below.

An AR model with Q and divergence

We first require a model connecting a seismic trace with a reflectivity series. Our model, although time-variable, is linear and can be stated in the matrix form

$$\mathbf{z} = \mathbf{A}^{-1}\mathbf{Q}\mathbf{D}\mathbf{r} + \mathbf{n} \tag{1}$$

where

z is a column vector composed of the samples z_t of a seismic trace,

A is a lower-triangular, Toeplitz matrix with auto-regressive (AR) coefficients on the diagonals. The first column of **A** is $[1\ a_1^0\ a_2^0\ \cdots\ a_l^0\ 0\ 0\ \cdots\ 0]^T$.

Q is a lower-triangular matrix described by Hale (1981) which transforms the impulse response for a non-attenuating medium to that for an attenuating medium. The elements of this matrix depend only on Q, the quality factor of the medium (assumed constant).

D is a diagonal matrix which transforms the impulsive, vertically propagating plane-wave response for a layered medium to the point source response for such a medium. In other words, multiplying by **D** applies spherical divergence. For a medium of constant velocity, the diagonal, non-zero elements of **D**, D_{tt}, are proportional to $1/t$.

r is a column vector containing the sampled response r_t of a layered, non-attenuating medium to a vertically propagating, impulsive plane-wave. **r** should include all multiple reflections and transmission losses.

n is a column vector containing samples n_t of additive noise.

QDr is the impulse response of an attenuating, layered earth. If we knew this impulse response, we could find the response to any time-invariant source waveform through convolution. The source waveform is here approximated by an AR(l) form; i.e., the sampled waveform has the Z transform

$$A^{-1}(Z) \equiv \frac{1}{1 + a_1^0 Z + a_2^0 Z^2 + \cdots + a_l^0 Z^l}$$

If the source waveform has moving average (MA) as well as AR components, then we assume l can be chosen large enough for $A^{-1}(Z)$ to sufficiently approximate the Z-transform of the source waveform. Multiplication by the matrix $\mathbf{A}^{-1}$ then represents convolution with the waveform.

Figure 1 illustrates the creation of a synthetic trace based on equation (1). **c** is a column vector (here plotted on its side) 500 samples long which represents a reflection coefficient sequence; the coefficients were derived from a Gaussian distribution of variance 0.05 with a 10% probability of having a non-zero value at any given sample. **r** was computed from **c** via the algorithm given by Claerbout (1976, p.160). **Dr** is the result of applying spherical divergence (i.e., dividing by t); **QDr** is the result of applying attenuation for $Q = 100$. The filter a^0 can be thought of as the inverse of a synthetic source waveform; the "signal" **s** is then obtained by convolving this source waveform with the earth's impulse response or, equivalently, $\mathbf{s} = \mathbf{A}^{-1}\mathbf{QDr}$. **n** is a Gaussian random noise sequence of variance 2.0×10^{-6}. The synthetic seismic trace **z** is the sum of the signal **s** and the noise **n**. Correcting **z** for spherical divergence (i.e., multiplying by t) yields the trace **y**.

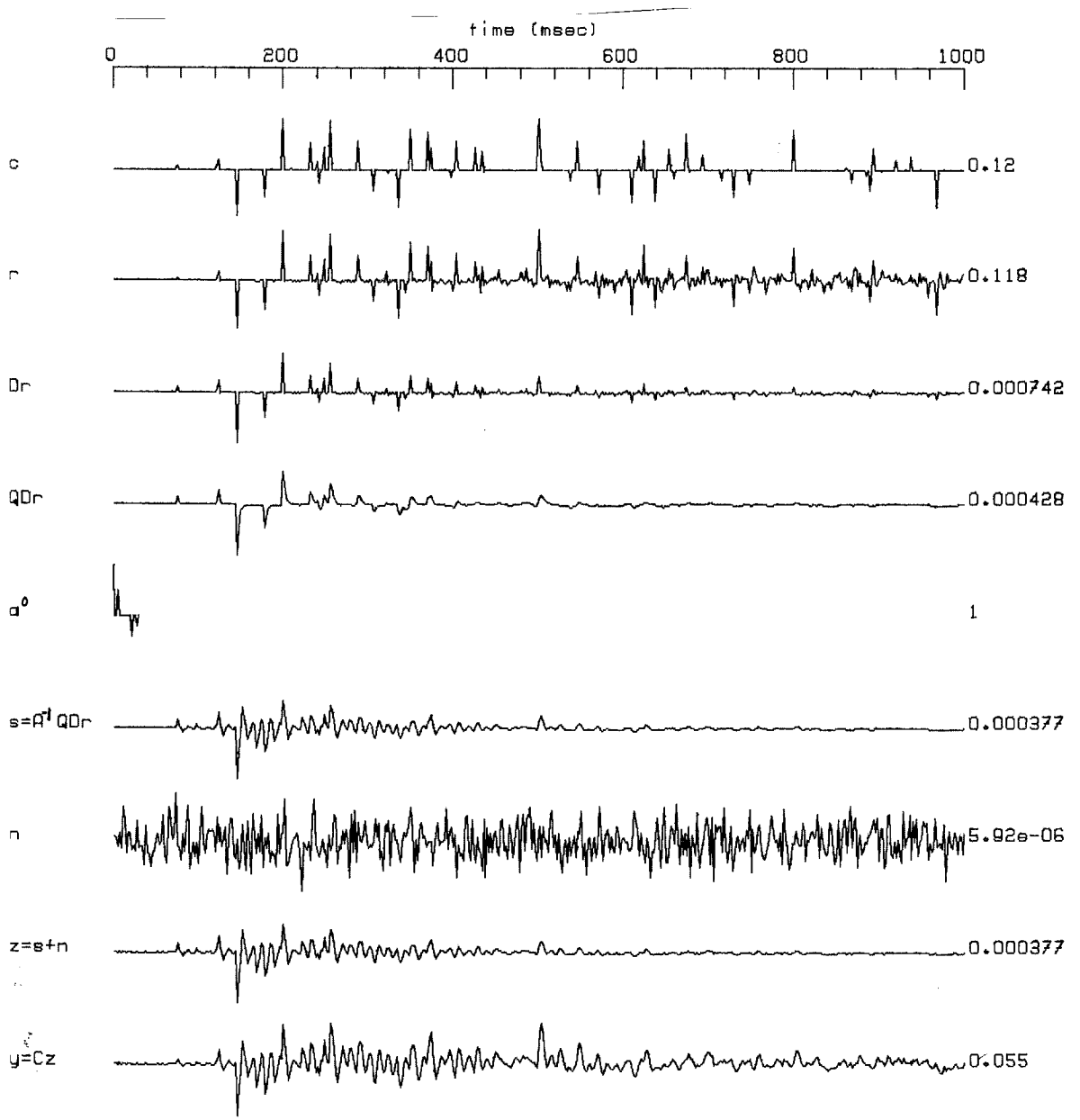

FIG. 1. Creation of a synthetic trace via the model of equation (1). Each trace is scaled independently for plotting; the maximum amplitude is given to the right of each trace. **c** is the reflection coefficient sequence; **r** includes all multiples and transmission losses; **Dr** includes spherical divergence; **QDr** includes attenuation for $Q = 100$; a^0 is the autoregressive coefficient sequence; **s** is the signal including all of the above; **n** is the noise; **z** is the synthetic trace; and **y** is **z** corrected for spherical divergence. The sampling interval is 2 msec.

Now assume that **n** = 0 and define two new matrices: $\mathbf{C} \equiv \mathbf{D}^{-1}$ and $\mathbf{P} \equiv \mathbf{Q}^{-1}$. Then

$$\mathbf{CPAz} = \mathbf{r} \tag{2}$$

In SEP-26 (Hale,1981) we noted that the inverse Q-filter **P** is a lower-triangular matrix with elements $P_{tj} = p^t_{t-j}$, where $p^t \approx p^{t-1} * (1+\varepsilon,-\varepsilon)$ and $p^0 = \delta$; i.e., p^t is approximately the tth convolutional power of the two-term filter $(1+\varepsilon,-\varepsilon)$. This two-term filter is the (least-squares) best, two-term inverse to an initially impulsive waveform that has been traveling in an attenuating medium (of quality factor Q) for a traveltime of one sampling interval. A numerical study has shown that $\varepsilon \approx 0.64/Q$ for a wide range (at least 25 to 400) of reasonable Q.

We assume (for simplicity) a constant velocity medium, for which the diagonal matrix **C** has elements C_{tt} proportional to t. Equation (2) can then be rewritten as

$$t \sum_{u=0}^{k} p^t_u \sum_{s=0}^{l} a^0_s z_{t-u-s} = \sum_{u=0}^{k} p^t_u \sum_{s=0}^{l} a^0_s t z_{t-u-s} = r_t \tag{3}$$

Because ε is typically a very small number, p^t effectively lengthens only gradually with increasing t; we choose k large enough so that $p^t_u \approx 0$ for $u \geq k$ and for all t of interest. Next define a trace y_t obtained by applying a spherical divergence correction to z_t, $y_t \equiv t z_t$; and rewrite equation (3) as

$$\sum_{u=0}^{k} p^t_u \sum_{s=0}^{l} a^0_s y_{t-u-s} \approx \sum_{u=0}^{k} p^t_u \sum_{s=0}^{l} a^0_s \frac{t}{t-u-s} y_{t-u-s} = r_t \tag{4}$$

The approximation, most valid for large t, is further justified by the fact that both p^t and a^0 are minimum-phase; hence, the significant contributions to the sums come from small u and s. (The approximation is a necessary consequence of the fact that **C** does not commute with **PA**.) After defining $g^t_u \equiv p^t_u / p^t_0$, $e_t \equiv r_t / p^t_0$, and $m \equiv l+k$, we manipulate indices and interchange summations in equation (4) to obtain

$$\sum_{s=0}^{m} \left[\sum_{u=0}^{k} g^t_u a^0_{s-u} \right] y_{t-s} = e_t$$

If we define the e_t to be prediction errors, then the quantity in parentheses must be a time-varying prediction error filter of length $m+1$ which we define to be

$$a^t_s \equiv \sum_{u=0}^{k} g^t_u a^0_{s-u} \tag{5}$$

and

$$\sum_{s=0}^{m} a^t_s y_{t-s} = e_t \tag{6}$$

The point of the development thus far has been to put the model given by equation (1) into the form of equation (6); APEF algorithms are based on this form. Given a trace y_t, the various APEF algorithms compute the prediction errors e_t while allowing the PEF coefficients a_s^t to vary more or less slowly with time t. We now establish a closer connection between Q and the changes in a_s^t with time.

Recall that $p^t = p^{t-1} * (1+\varepsilon,-\varepsilon)$ where $p^0 = \delta$ and $\varepsilon \approx 0.64/Q$. Defining $\gamma \equiv -\varepsilon/(1+\varepsilon) \approx -0.64/Q$ and using the definition of g_u^t, we obtain $g^t = g^{t-1} * (1,\gamma)$ where $g^0 = \delta$; or, in other words, g^t is the tth convolutional power of $(1,\gamma)$ with coefficients

$$g_u^t = \frac{t!\gamma^u}{(t-u)!u!}$$

We now rewrite equation (5) as

$$a_s^t = \sum_{u=0}^{k} \frac{t!\gamma^u}{(t-u)!u!} a_{s-u}^0 \tag{7}$$

Examples are

$$a_0^t = 1$$

$$a_1^t = a_1^0 + t\gamma$$

$$a_2^t = a_2^0 + t\gamma a_1^0 + \frac{t(t-1)}{2}\gamma^2$$

and so on. Verify that for no attenuation, $\gamma \to 0$, and $a^t = a^0$. An alternative, useful way of expressing equation (7) is

$$a_s^t = a_s^{t-1} + \gamma a_{s-1}^{t-1} \tag{8}$$

Also, recalling that $a_0^t = 1$, we note that

$$a_1^t = a_1^{t-T} + T\gamma \approx a_1^{t-T} - \frac{0.64T}{Q}, \tag{9}$$

a relation we shall find useful in the following discussion.

Q-ignorant adaptive prediction error filtering

We are now equipped to investigate the first of the two possibilities stated earlier; we attempt to learn Q from the way in which an APEF, one which knows nothing about the physical process of attenuation, adapts. From among the many existing APEF algorithms, we chose two as generally representing the capabilities of APEF. The first (hereafter referred

to as APEF1), derived by Lee et al (1981), is a recursive least squares algorithm which, according to the authors, has "fast parameter tracking capability compared to gradient based algorithms." An example of a gradient based algorithm is that derived by Widrow (1970) and implemented in a seismic data context by Griffiths et al (1977). For our purpose, the similarities between recursive least squares and gradient based algorithms are more important than the differences. Both types of algorithms effectively work by weighting the input data, y_t. Suppose you wanted to compute the PEF a^τ from y_t. Since y_t is non-stationary, you would place more trust in (or weight on) the samples $y_{\tau-1}, y_\tau, y_{\tau+1}$ than in the samples $y_{\tau-100}$ or $y_{\tau+100}$. APEF1 effectively computes a^τ from exponentially weighted present and *past* samples of y_t; i.e., a^τ is derived from $y_\tau, \lambda y_{\tau-1}, \lambda^2 y_{\tau-2}, \lambda^3 y_{\tau-3}, \cdots$, where $0 < \lambda < 1$ is a user-chosen parameter governing the rate of adaptation or, equivalently, the rate at which past data is "forgotten". Griffiths et al show how the same exponential weighting is effectively present in gradient based APEFs.

The second APEF (APEF2) we chose was developed (to the best of the author's knowledge) in the SEP around 1972, primarily by Don Riley and J.P. Burg. The algorithm has never been published (except possibly in preprint form for the 1972 meeting of the SEG); the author learned of it from Francis Muir in 1981. A detailed derivation of the algorithm and a Fortran 77 subroutine to apply it are provided in the Appendix. APEF2, like APEF1, applies exponential weighting. The major differences between the two algorithms are that (1) APEF2 exponentially weights the prediction errors e_t rather than the input data y_t and that (2) APEF2 uses *future* as well as present and past prediction errors to derive a^τ. Simply stated, APEF2 computes a^τ to minimize the sum of squared $\cdots, \lambda^2 e_{\tau-2}, \lambda e_{\tau-1}, e_\tau, \lambda e_{\tau+1}, \lambda^2 e_{\tau+2}, \cdots$. [Actually, as in Burg's non-adaptive algorithm (Burg, 1975), the sum of both forward and backward squared prediction errors is minimized. See the Appendix.] Again, λ governs the adaptation rate of a^t.

To test both APEF1 and APEF2 under optimum conditions, we applied them to a *noiseless* version of the divergence-corrected trace y=Cz of Figure 1. The results for APEF1 with $\lambda = 0.97$ are shown in Figure 2a. The traces r_t and (noiseless) y_t are replotted for comparison with the estimate r_t'. The output of APEF1 is an estimate of $e_t = r_t / p_0^t \approx r_t / (1+\varepsilon)^t$. Exponential gain applied to the output of APEF1 yields r_t'.

Also plotted as a function of time in Figure 2a are the first three coefficients, a_1^t, a_2^t, and a_3^t, of the computed PEF. (The entire PEF was 20 coefficients long.) As expected from equation (7), the coefficient a_1^t exhibits a dominant linear trend with time t; the slope of this trend is an estimate of γ which, in turn, provides an estimate of Q. The curvature with t of a_2^t reflects the quadratic trend predicted by equation (7), while a_3^t exhibits the predicted cubic trend.

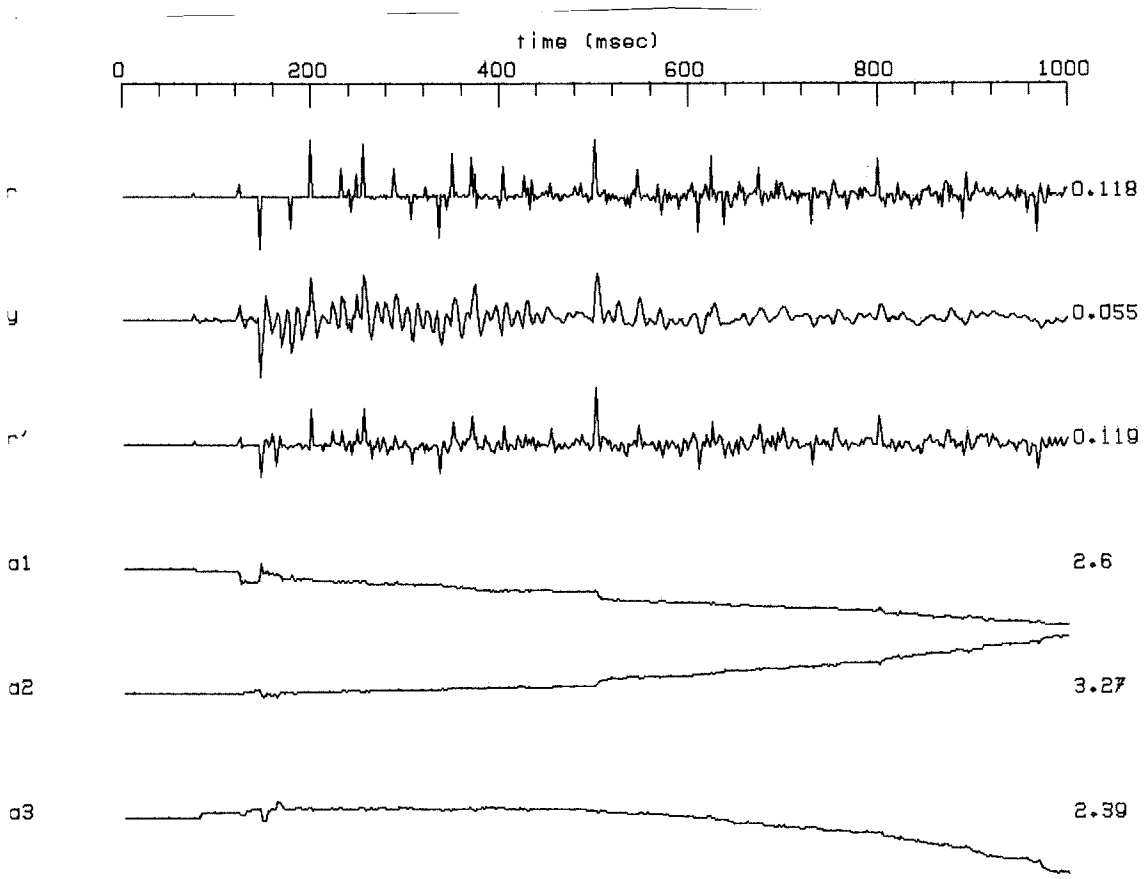

FIG. 2a. Adaptive prediction filtering with APEF1 ($\lambda = 0.97$). r is replotted from Figure 1 for comparison with the estimate r′. y is the noiseless, divergence-corrected trace input to APEF1. a1, a2, and a3 are the first three coefficients of the time-varying prediction error filter.

An obvious way to estimate Q is to fit curves given by equation (7) to the m coefficients of a^t. Such a procedure would necessarily include the assumption that Q is constant over the fitting interval. Equation (8), on the other hand, suggests that an estimate of Q is theoretically available at *every* time t. Although equation (8) was derived assuming a constant Q, time (actually, depth) variable Q can be estimated by replacing γ in that equation with γ_t. This substitution is strictly valid only in the absence of multiple reflections but, even in the presence of multiples, may provide a useful estimate. An example of data for which this substitution is dubious is that containing strong seafloor multiple reflections. Due to the relatively high Q of water, these multiple events would be only slightly attenuated relative to the primary events on which they are superimposed; attempts to estimate time-variable Q via equation (8) would then yield erroneous (indeed, even negative) Q.

For our synthetic trace y_t, Q = 100 is constant. Equation (8) applied to the PEF coefficients of Figure 2a should then yield reasonably constant estimates; if it does not, then our ability to estimate time-variable Q is questionable. We estimated $1/Q$ using equation (9), equivalent to equation (8) for s = 1, for T = 1, 50, and 100. The estimates are plotted in Figure 2b.

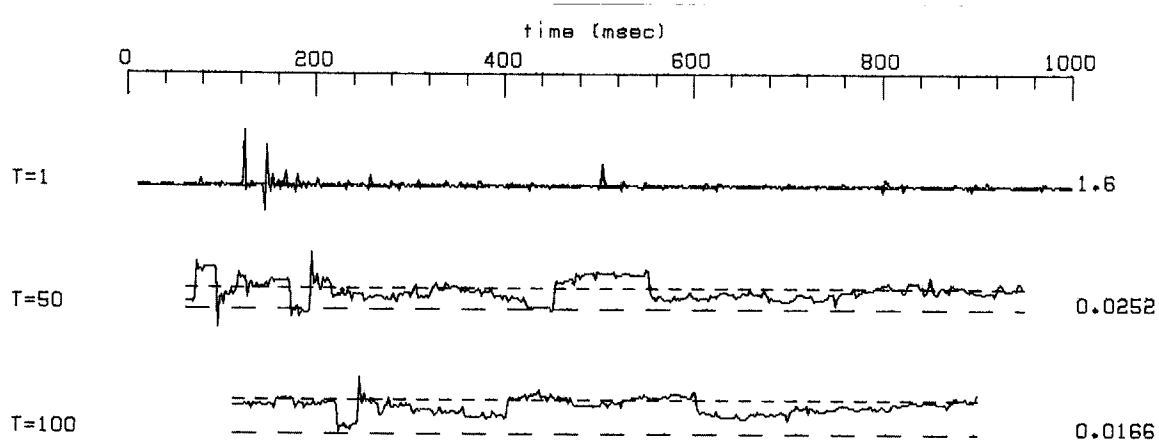

FIG. 2b. Estimates of $1/Q$ based on the coefficient a1 plotted in Figure 2a. The estimates were computed using equation (9) for three different values of T. The short dashes represent the correct answer, $1/Q$ = 0.01; and the long dashes represent $1/Q$ = 0 (plotted for reference).

The attempt to estimate $1/Q$ at each time sample (T=1) has clearly failed to produce accurate, constant estimates. The largest errors in the estimates occur at times for which a_1^t changes drastically which, in turn, correspond to the temporal locations of large reflection coefficients. The statistical knowledge accumulated by APEF1 and hence, each PEF coefficient, changes as each new reflection coefficient is discovered. (Remember that APEF1 designs its PEF from exponentially weighted past input.) The rapid fluctuations in PEF coefficients degrade our estimate of Q.

A better approach might be to admit that we cannot realistically resolve changes in Q to within a sampling interval. The traces labeled T=50 and T=100 in Figure 2b are effectively the result of estimating $1/Q$ from smoothed changes in a_1^t. While smoothing has improved the estimates significantly (they are at least predominantly positive), the considerable fluctuation remaining implies that a resolution of 200 msec (T=100) would still be too

optimistic. Smoothing over the entire length of a_1^t (T=500) yields the reasonable estimate $Q = 120$.

In spite of the attention we have given to the matter, estimation of Q should perhaps be viewed as a fringe benefit of APEF; "deconvolution" of the seismic trace is likely the primary purpose. r_t' in Figure 2a, aside from some rather large errors at early times when APEF1 is still "learning", is a reasonable estimate of r_t. A comparison of this estimate with those obtained with other APEFs, as well as with more conventional time-invariant PEF, is provided below.

The traces in Figures 3a and 3b were computed in the same way as those of Figures 2a and 2b, except that APEF2, again with $\lambda = 0.97$, was used instead of APEF1. Recall that APEF2 is an APEF with foresight as well as hindsight; it uses future as well as past prediction errors to design a^t. This feature of APEF2 explains the relatively smooth variations with time of both the PEF coefficients in Figure 3a and the estimates of $1/Q$ in Figure 3b. With 200 msec smoothing (T=100), the $1/Q$ estimates are fairly constant when compared with the corresponding estimates in Figure 2b. If estimation of Q is even a secondary goal of APEF, then APEF2 should be used rather than hindsight-only methods such as APEF1.

The errors in the deconvolved trace r_t' of Figure 3a are comparable to those of Figure 2a, except at early time where APEF2 yields the better r_t'. APEF2, in a sense, knows what lies ahead and, therefore, requires no learning time before yielding good estimates of r_t.

In the application of both APEF1 and APEF2, the adaptation parameter λ was chosen to be $\lambda = 0.97$. How critical is our choice of this parameter in estimating both $1/Q$ and r_t? Figures 4a and 4b, like Figures 3a and 3b, were obtained using APEF2 except that $\lambda = 0.94$ was used instead of $\lambda = 0.97$. Using a smaller value of λ is roughly equivalent to using a smaller window of y_t in computing the PEF a^t. In using $\lambda = 0.94$ rather than $\lambda = 0.97$, we have effectively halved the amount of information available to APEF2 for computing each a^t.

The dangers of adapting too rapidly (i.e., choosing λ too small) are seen by comparison of Figures 4a and 4b with Figures 3a and 3b. The $1/Q$ estimates of Figure 4b are less consistent than those of Figure 3a, although the general trend is the same in both figures. Perhaps more significant is the effect of our choice of λ on the deconvolved trace. The quality of r_t' in Figure 4a is generally inferior to that in Figure 3a. The faster adaptation rate has allowed APEF2 to undesirably attack reflection coefficients along with the attenuated source waveform. Compare, for example, the estimates at about 620 msec.

The sensitivity of both $1/Q$ and r_t estimates to λ suggests that this parameter should be chosen carefully. In most cases, our choice will be conservative; we will choose λ as close as possible to unity while still coping with the non-stationarity of the trace y_t. We

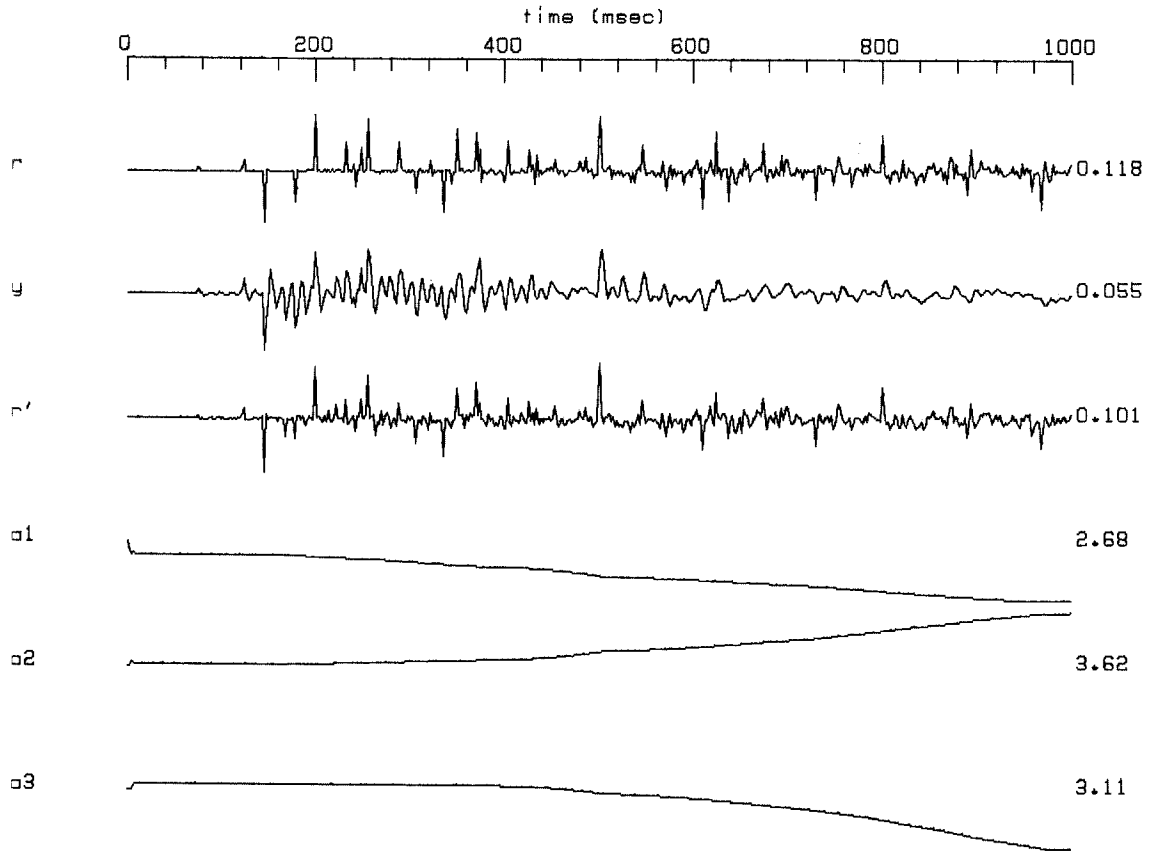

FIG. 3a. Adaptive prediction filtering with APEF2 ($\lambda = 0.97$). r is replotted from Figure 1 for comparison with the estimate r'. y is the noiseless, divergence-corrected trace input to APEF2. a1, a2, and a3 are the first three coefficients of the time-varying prediction error filter.

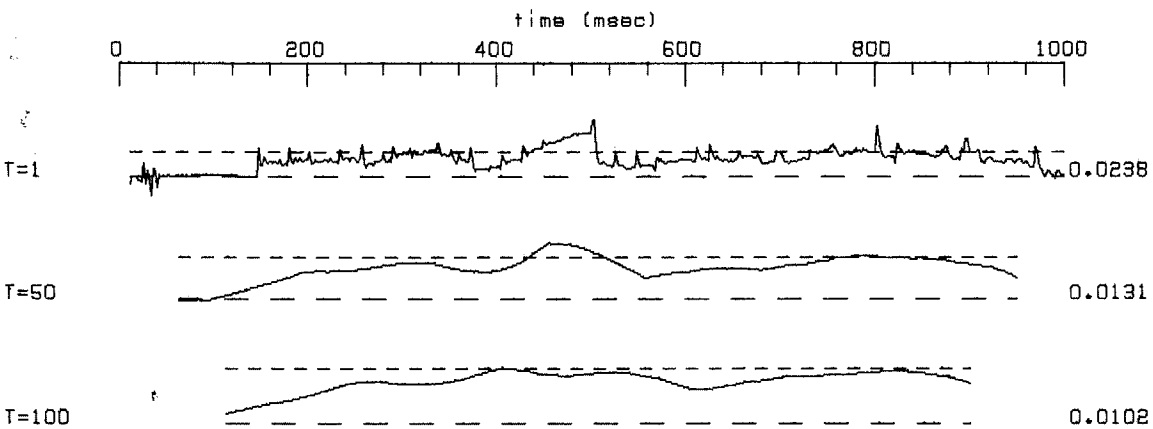

FIG. 3b. Estimates of $1/Q$ based on the coefficient a1 plotted in Figure 3a. The estimates were computed using equation (9) for three different values of T. The short dashes represent the correct answer, $1/Q = 0.01$; and the long dashes represent $1/Q = 0$ (plotted for reference).

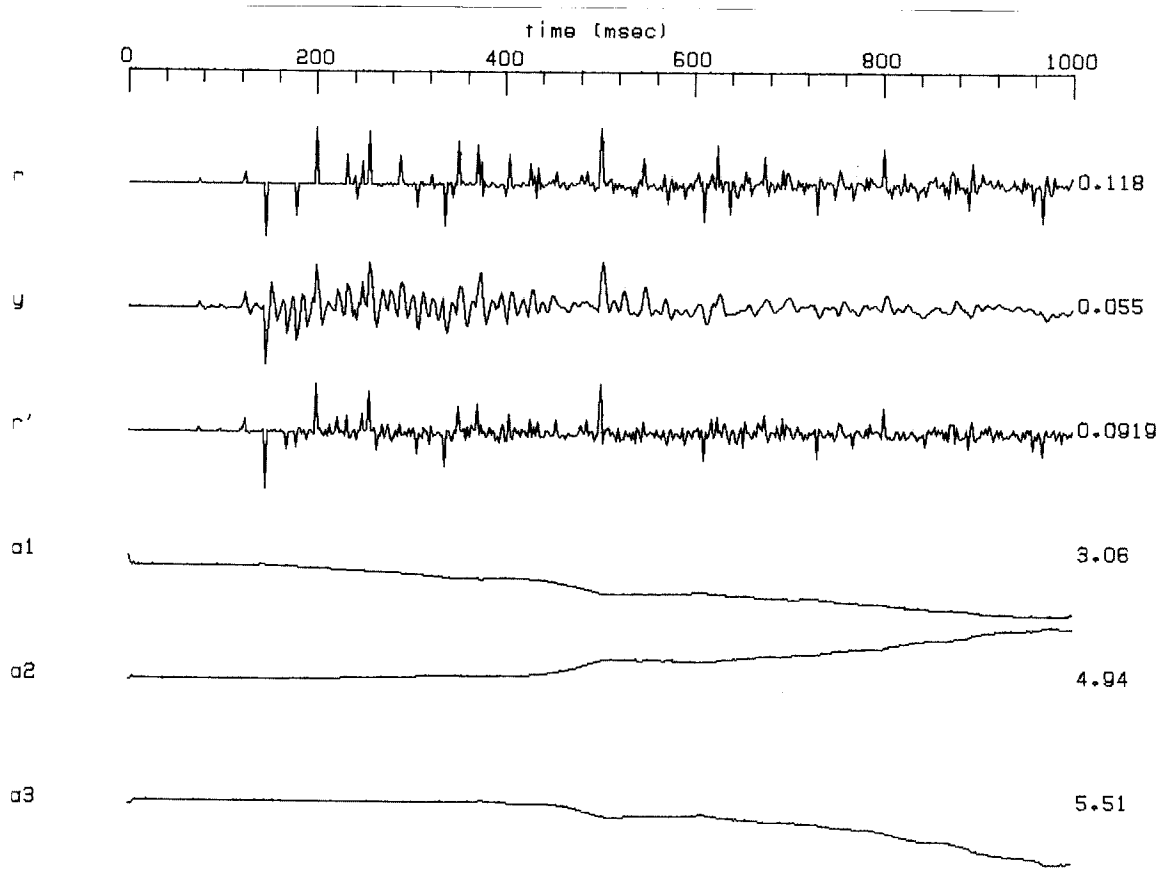

FIG. 4a. Adaptive prediction filtering with APEF2 (λ = 0.94). r is replotted from Figure 1 for comparison with the estimate r′. y is the noiseless, divergence-corrected trace input to APEF2. a1, a2, and a3 are the first three coefficients of the time-varying prediction error filter.

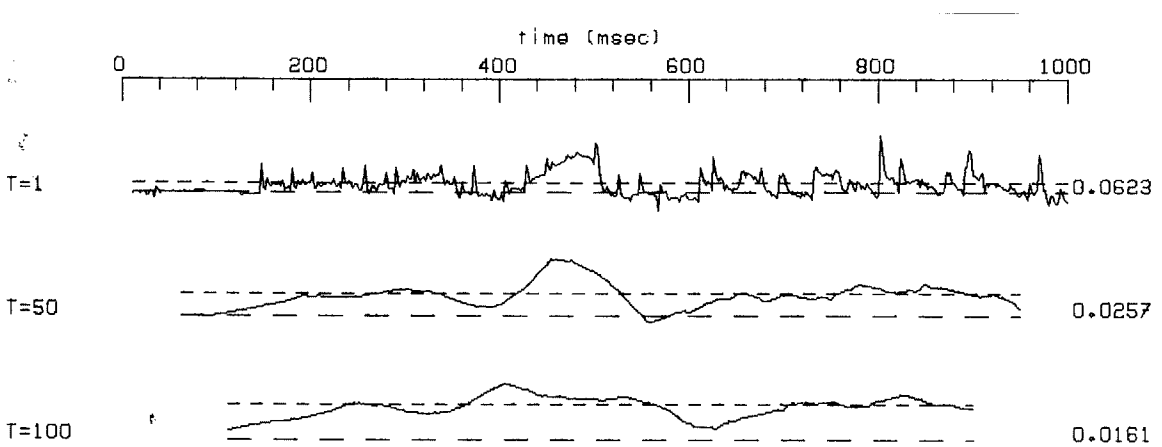

FIG. 4b. Estimates of $1/Q$ based on the coefficient a1 plotted in Figure 4a. The estimates were computed using equation (9) for three different values of T. The short dashes represent the correct answer, $1/Q$ = 0.01; and the long dashes represent $1/Q$ = 0 (plotted for reference).

can imagine, however, data in which non-stationarity is severe enough (e.g., Q is low enough) that no suitable λ can be chosen. In the next section of this paper, we present an APEF method for which no adaptation parameter λ is required.

Q-wise adaptive prediction error filtering

The two APEF methods discussed above have an important common feature: they adapt by *weighting* data. APEF1 weights input data and APEF2 weights output prediction errors, and weighting is equivalent to throwing away useful information. For example, if the model given by equation (1) is valid, then information pertaining to the *time-invariant* AR parameters $a_1^0, a_2^0, \cdots, a_l^0$ is contained in every reflection found in **z**. The basic source waveform does not change with time; the earth's impulse response **QDr** is the time-varying function.

We can avoid weighting by constraining our APEF to compensate only for attenuation effects; this is the second of the two possibilities mentioned earlier. We simply use an APEF which "understands" attenuation. The problem is then to estimate both the AR coefficients a^0 and Q. Once these parameters are estimated, we can estimate r_t using equation (4).

Our starting point for deriving a Q-wise APEF is equation (6) which, when combined with equation (7), becomes

$$e_t = \sum_{s=0}^{m} a_s^t y_{t-s} = \sum_{s=0}^{m} y_{t-s} \sum_{u=0}^{k} \frac{t!\gamma^u}{(t-u)!u!} a_{s-u}^0 \tag{10}$$

The standard approach in prediction error filtering is to minimize a sum of squared e_t with respect to the unknown parameters, in this case γ and the l coefficients a_s^0. This least-squares approach works best when e_t is a linear function of the parameters, for then the "best" parameters may be found by solving a linear system of equations. In equation (10), however, e_t is a non-linear function of γ; an alternative method of solving for the parameters is required.

Assume for now that the a_s^0 are known and define $\gamma \equiv \bar{\gamma} + d\gamma$ where $\bar{\gamma}$ is a guess of the γ which minimizes the sum of squared $e_t(\gamma)$. Then define

$$E(d\gamma) \equiv \sum_t e_t^2(\bar{\gamma}+d\gamma) \approx \sum_t \left[e_t(\bar{\gamma}) + \frac{\partial e_t}{\partial \gamma} d\gamma \right]^2$$

where the partial derivative is to be evaluated at $\bar{\gamma}$. Setting $dE/d(d\gamma) = 0$ and solving for $d\gamma$ yields

$$d\gamma = -\frac{\sum_t e_t(\bar{\gamma}) \dfrac{\partial e_t}{\partial \gamma}}{\sum_t \left(\dfrac{\partial e_t}{\partial \gamma} \right)^2}$$

Partial differentiation of equation (10) yields (after some manipulation of summation indices) $\partial e_t / \partial \gamma = t e_{t-1}$ so that

$$d\gamma = -\frac{\sum_t t e_t(\bar{\gamma}) e_{t-1}(\bar{\gamma})}{\sum_t t^2 e_{t-1}^2(\bar{\gamma})} \tag{11}$$

Equation (11) makes good sense. Suppose our guess $\bar{\gamma}$ is a very good guess so that $d\gamma = 0$; equation (11) then requires that the first lag of the "autocorrelation" of weighted e_t equal zero. Except for the weighting by t, this requirement is familiar from our experience with time-invariant PEF, a process which is based on the assumption that the prediction errors e_t are mutually uncorrelated random variables; i.e., $E(e_t e_{t+\tau}) = 0$ for $\tau \neq 0$ where E denotes ensemble averaging. In practice, we approximate ensemble averaging with time averaging which may include some time-dependent weighting. Because a change in γ has its greatest effect at late traveltimes (for this is where attenuation has had its greatest effect), the weighting by t is quite reasonable.

Why is only the first lag of the error autocorrelation important in determining the perturbation $d\gamma$? The answer lies in the fact that trends with frequency in a power spectrum, such as the decaying exponential trend produced by attenuation, are most affected by the value of the first lag of the corresponding autocorrelation. (Recall that the power spectrum of a time series is the Fourier transform of its autocorrelation.) The values of greater lags affect more rapid variations in the power spectrum (e.g., notches in the spectra of marine data due to multiples or bubble pulses).

Given then that equation (11) is both mathematically and intuitively reasonable, we propose the following iterative algorithm for estimating the unknown parameters γ and the l coeffiecients a_s^0 and, ultimately, r_t.

Initially $\gamma = \bar{\gamma}$; $a_s^0 = \delta_s$; y_t = the divergence-corrected seismic trace

(1) $x_t = \sum_{u=0}^{k} p_u^t y_{t-u}$

(2) $a^0 = PEF(x_t)$

(3) $r_t = \sum_{s=0}^{l} a_s^0 x_{t-s}$

(4) $d\gamma = -\frac{\sum_t t r_t r_{t-1}}{\sum_t t^2 r_{t-1}^2}$

if $|d\gamma| < small$ go to Finally

(5) $\gamma = \gamma + d\gamma$
go to (1)

Finally $Q = -0.64(1+\gamma)/\gamma$

Step (1) is the application of the inverse Q-filter p^t (derived from γ) to y_t. *PEF* in step (2) of the algorithm is a subroutine which computes a time-invariant, minimum-phase, unit-lag prediction error filter from x_t computed in step (1). This subroutine is that used in conventional spiking deconvolution. Step (3) is the application of the filter derived in step (2). $d\gamma$ is computed in step (4) using equation (11) with r_t replacing e_t. Recall that r_t is just an exponentially gained version of e_t; the approximation in using r_t is made merely to save the cost of computing e_t. If $d\gamma$ is insignificant, then we stop iterating; otherwise we update γ in step (5) and go back to step (1). Finally (if desired), we compute from γ the estimate of Q.

Figure 5 illustrates the application of this iterative algorithm to the noiseless synthetic y_t. The initial guess for γ used to start the algorithm was $\bar{\gamma} = 0$. The trace labeled r1′ is the first estimate of r_t computed in step (3). Because we initially assumed no attenuation ($\bar{\gamma} = 0$), r1′ is also the result we would obtain with conventional spiking deconvolution. Notice that the high frequencies have not been restored at late times and that they have been overamplified at early times. The latter effect would be even more pronounced had we compensated for the amplitude decay with time of y_t prior to computing a^0 in step (2).

Successive estimates r2′, r3′, ... demonstrate progressive improvement with each iteration. The algorithm has effectively converged after about four or five iterations. The trace labeled r′ is the final estimate of r_t obtained after ten iterations; the final estimate of Q was $Q = 99.6$, essentially the correct value of 100. To decrease the computational burden, the crudest approximation to the inverse Q-filter p^t was used for all but the last iteration. As discussed by Hale (1981), $p^t \approx p^{t-1} * (1+\varepsilon,-\varepsilon)$ is only an approximate inverse Q-filter which can be improved by increasing the length of $p^1 \approx (1+\varepsilon,-\varepsilon)$ from two to, say, ten coefficients. A ten coefficient approximation to p^1 was used in computing r_t'.

Early experiments with Q-wise APEF showed that the algorithm as presented above is rather cautious in converging. The results presented in Figure 5 were obtained by updating γ with twice the perturbation computed in step (4). The algorithm still never over-corrected γ (i.e., $d\gamma$ was always negative), suggesting that even larger perturbations may be used to speed convergence.

Not surprisingly, the estimate r_t' obtained with Q-wise APEF (Figure 5) is generally superior to that obtained with Q-ignorant APEF (Figures 2a and 3a), because Q-wise APEF, unlike Q-ignorant APEF, uses all of the input data to estimate the model parameters Q and

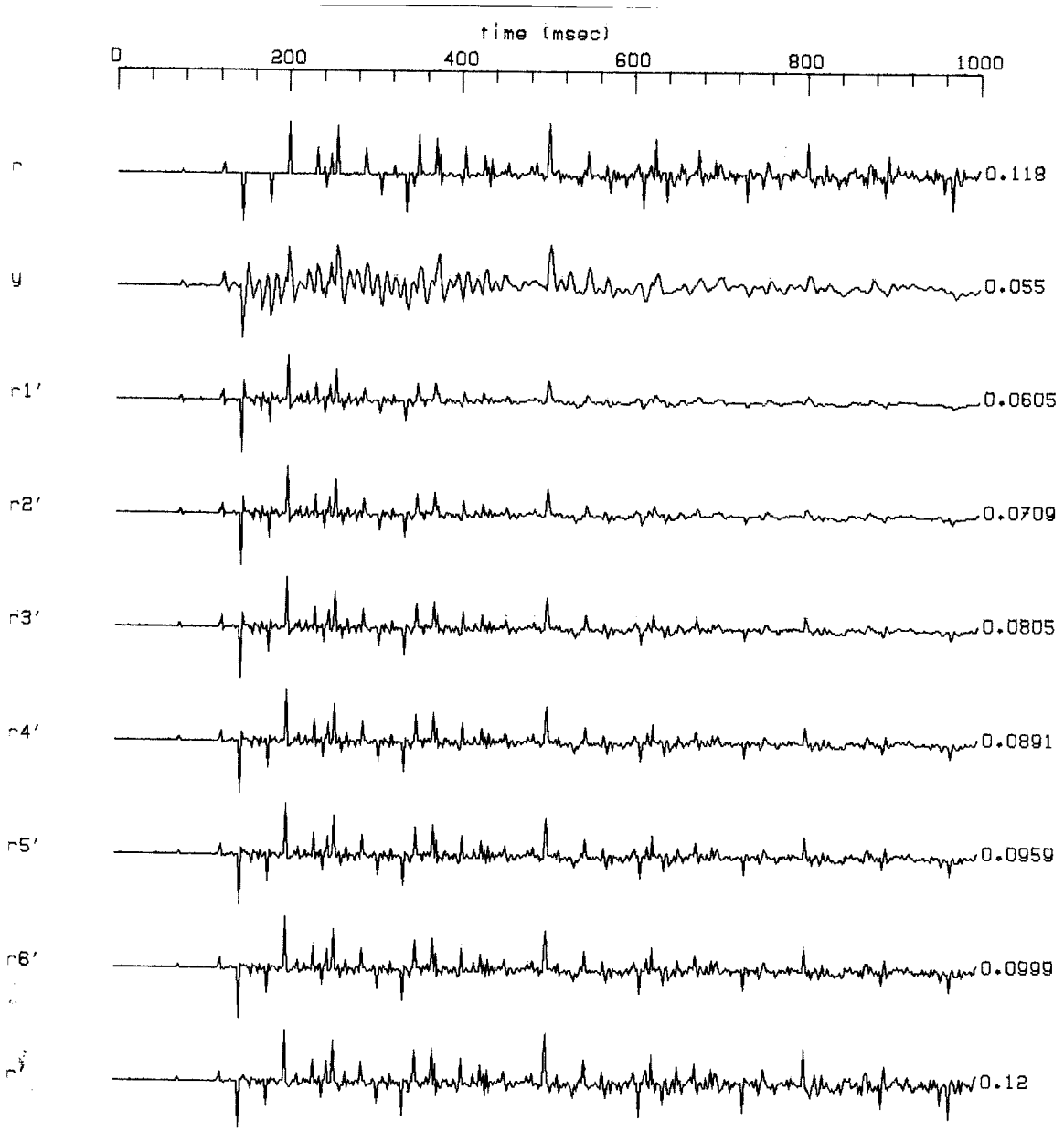

FIG. 5. Adaptive prediction error filtering with a Q-wise APEF. r is replotted from Figure 1. y is the divergence-corrected, noiseless, synthetic trace. r1′, r2′, ... are successive estimates of r produced by the iterative Q-wise APEF. r′ is the final estimate of r; the final estimate of Q was $Q = 99.6$.

a^0. We must not assume, however, that Q-wise APEF will always produce better estimates when applied to real seismic data. Performance will always on how well a particular set of data fits our model.

Noise

Of the many assumptions and approximations that have been made above, two are particularly regrettable: (1) in Q-wise APEF, the assumption that Q is constant over a lengthy time (actually depth) interval and (2) in both methods, the assumption of a noiseless, divergence-corrected seismogram. Regarding the first assumption, Q-wise APEF as described in this paper can only estimate a Q (actually $1/Q$) averaged over the interval from which the estimate is derived. No attempt has yet been made to modify the algorithm for variable Q. Initial results with Q-ignorant APEF suggest that these algorithms are also unable to resolve rapid variations in Q (recall Figures 2b, 3b, and 4b).

The second assumption of no noise is even more troublesome. We applied both Q-ignorant and Q-wise APEF to the noise contaminated y_t in Figure 1 to obtain the results shown in Figures 6 and 7, respectively. At early times, when the signal level is high relative to that of the noise, results in Figures 6 and 3a (the noiseless results) are quite similiar; at late times, they are quite different as we should expect. Effectively, Q-ignorant APEF at late times stops learning about Q and starts learning about noise.

Q-wise APEF, on the other hand, cannot adapt to the time-variable signal-to-noise ratio. Hence, as shown in Figure 7, the estimates of r_t change little with each iteration; $\mathbf{r}' \approx \mathbf{r1}'$. In other words, Q-wise APEF shows little improvement over conventional spiking deconvolution. The estimated Q is $Q \approx 600$.

The results in Figures 6 and 7 suggest that "ignorance is bliss" when applying APEF to noisy seismic data. An APEF which "knows" everything about attenuation but nothing about noise may perform worse than an APEF which knows nothing at all.

Conclusions

Beginning with a simple model for an attenuated seismic trace, we can relate the time-variations of an adaptive prediction error filter to Q; i.e., we know how an APEF responding to attenuation effects should behave.

Of the two unconstrained, Q-ignorant APEF algorithms we have tested, APEF2 produces better estimates of both Q and r_t than does APEF1. We attribute the differences between results obtained with these two algorithms to be due to the ability of APEF2 to use future as

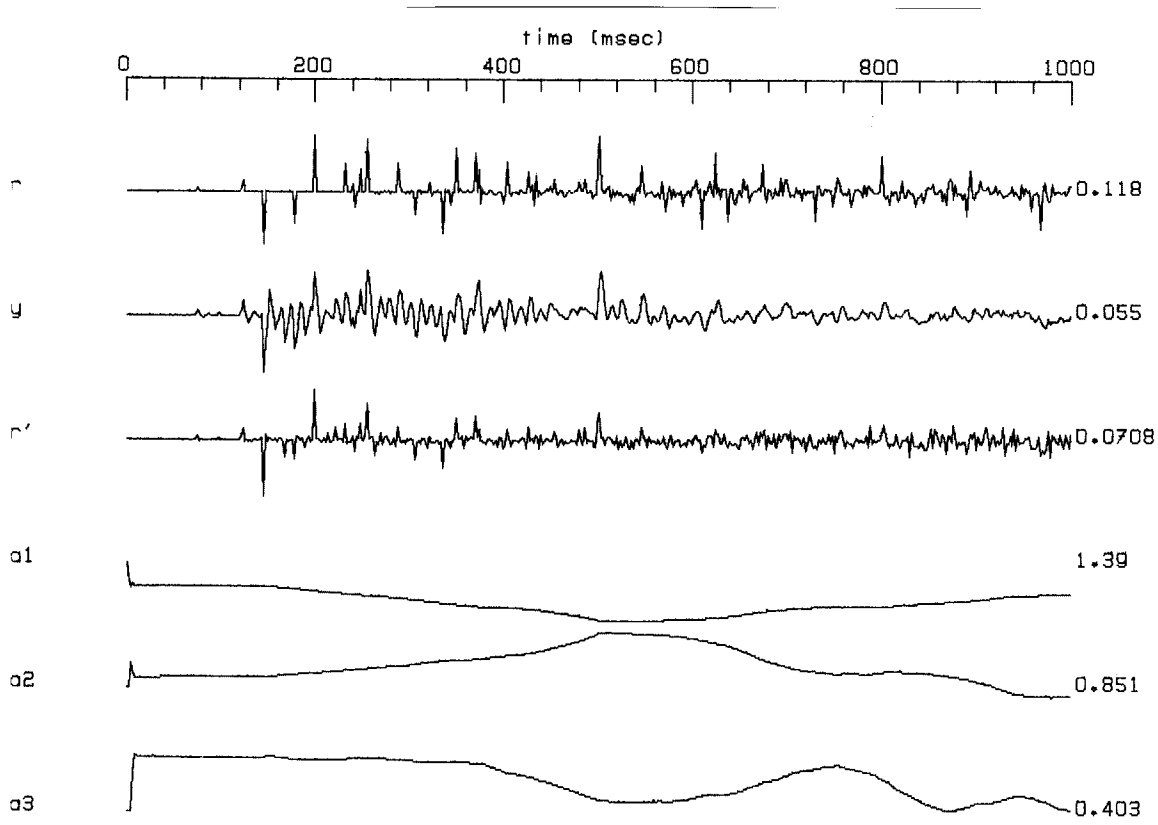

FIG. 6. Adaptive prediction error filtering (APEF2 with $\lambda = 0.97$) of noisy data. r is replotted from Figure 1; y is the noisy, divergence-corrected trace, also from Figure 1. r′ is the estimate of r and a1, a2, and a3 are the time-varying coefficients of the prediction error filter.

well as past data in designing a local prediction error filter. The basic fault of Q-ignorant algorithms stems from their adapting by weighting. Weighting not only forces us to neglect useful information about time-invariant components of the seismogram; it also requires that we choose a priori the adaptation rate (the weighting factor λ) of the APEF.

Weighting is unnecessary if we constrain the APEF to adapt only to attenuation effects. A Q-wise APEF can potentially yield better estimates of Q and r_t than can a Q-ignorant APEF. The Q-wise APEF is an iterative application of inverse Q-filtering and time-invariant spiking deconvolution; it is particularly efficient in filtering many traces successively, for then the estimate of Q from one trace is a good first guess of Q for the next trace. The basic fault of Q-wise APEF is that it cannot adapt to a time-varying signal-to-noise ratio. One untested solution to this problem may lie in estimating an ambient, white

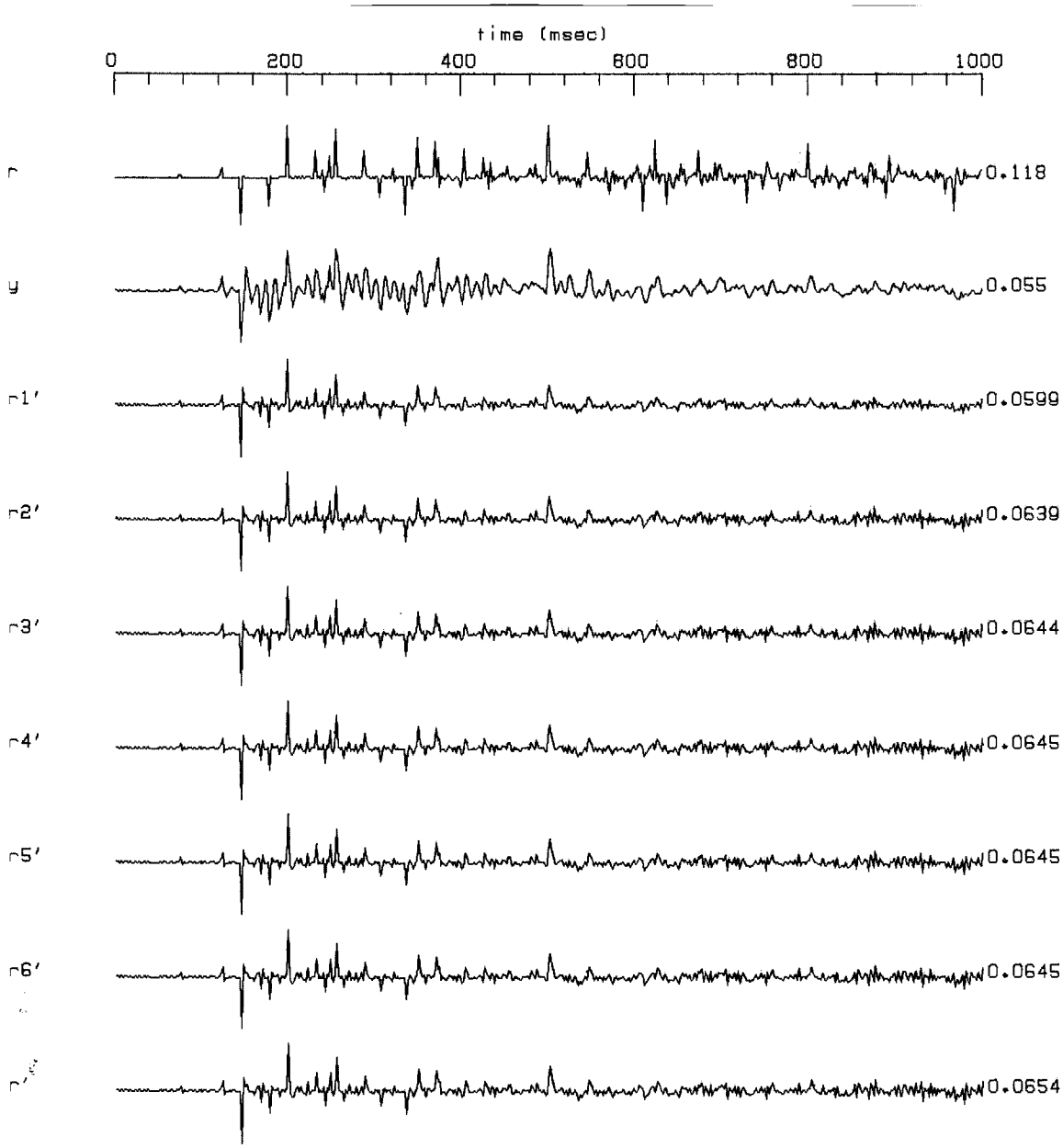

FIG. 7. Q-wise adaptive prediction error filtering of noisy data. r is replotted from Figure 1; y is the noisy, divergence-corrected trace, also from Figure 1; r1′, r2′, ... are successive estimates of r produced by the iterative Q-wise APEF. r′ is the final estimate of r; the final estimate of Q was $Q \approx 600$.

noise level along with auto-regressive coefficients and Q; another may involve the use of a bandlimited inverse Q-filter.

Neither Q-ignorant nor Q-wise APEF appears capable of resolving rapid depth variations in Q; APEF is not yet likely to be useful in detecting an abrupt Q-contrast associated with a gas sand. Lateral, or rather, trace-to-trace variations in Q may be more reliably estimated.

REFERENCES

Burg, J.P., 1975, Maximum entropy spectral analysis: Ph.D. thesis, Stanford University, California.

Claerbout, J.C., 1976, Fundamentals of geophysical data processing with applications to petroleum prospecting: New York, McGraw-Hill Book Company.

Griffiths, L.J., Smolka, F.R., and Trembly, L.D., 1977, Adaptive deconvolution: a new technique for processing time-varying seismic data: Geophysics, v. 42, p. 742-759.

Hale, D., 1981, An inverse Q-filter: Stanford Exploration Project Report 26, p. 231-243.

Lee, D.T.L., Morf, M., and Friedlander, B., 1981, Recursive least squares ladder estimation algorithms: IEEE Trans. Circuits Syst., v. CAS-28, p.467-481.

Widrow, B., 1970, Adaptive filters, in Aspects of network and system theory: New York, Holt, Rinehart and Winston, Inc.

Appendix: an adaptive prediction error filter with foresight

Let y_t denote the input time series known for $t = 0, 1, \cdots, n-1$; and let $a_{t,k}$ denote the kth order prediction error filter with Z-transform

$$A_k(Z) = 1 + a_{1,k}Z + a_{2,k}Z^2 + \cdots + a_{k,k}Z^k$$

We guarantee $a_{t,k}$ to be minimum-phase for all k by building $a_{t,k}$ with the Levinson recursion:

$$A_k(Z) = A_{k-1}(Z) - c_k Z^k A_{k-1}(Z^{-1}) \qquad \text{(A1)}$$

starting with $A_0(Z) = 1$. If we know c_k for $k = 1, 2, \cdots, m$, then we can find, using equation (A1) recursively, the prediction error filters of order $k = 1, 2, \cdots, m$. Now define both forward and backward prediction errors:

$$f_{t,k} \equiv \sum_{s=0}^{k} a_{s,k}\, y_{t-s} \qquad \text{(A2a)}$$

$$b_{t,k} \equiv \sum_{s=0}^{k} a_{s,k}\, y_{t-k+s} \quad ; \quad t = k, k+1, \cdots, n-1 \qquad \text{(A2b)}$$

Using equation (1), one can derive the following recursive forms for equations (A2):

$$f_{t,k} = f_{t,k-1} - c_k b_{t-1,k-1}$$

$$b_{t,k} = b_{t-1,k-1} - c_k f_{t,k-1} \quad ; \quad t = k, k+1, \cdots, n-1$$

with initial conditions

$$f_{t,0} = b_{t,0} = y_t \quad ; \quad t = 0, 1, \cdots, n-1$$

Next define an exponentially weighted sum of squared forward and backward errors:

$$E_t(c_k) \equiv \sum_{s=k}^{n-1} \lambda^{|t-s|}(f_{s,k}^2 + b_{s,k}^2)$$

$$= \sum_{s=k}^{n-1} \lambda^{|t-s|}[(f_{s,k-1} - c_k b_{s-1,k-1})^2 + (b_{s-1,k-1} - c_k f_{s,k-1})^2]$$

where $0 \ll \lambda \leq 1$. We choose c_k to minimize E_t. Setting $\partial E_t / \partial c_k = 0$ yields

$$c_{t,k} = \frac{2\sum_{s=k}^{n-1} \lambda^{|t-s|} f_{s,k-1} b_{s-1,k-1}}{\sum_{s=k}^{n-1} \lambda^{|t-s|}(f_{s,k-1}^2 + b_{s-1,k-1}^2)} \qquad \text{(A3)}$$

The subscript t on $c_{t,k}$ is appropriate because $c_{t,k}$ is obtained by minimizing E_t; for $\lambda < 1$, $c_{t,k}$ depends heavily on the forward and backward prediction errors centered near the time t, less so on the errors far from t. A different value of t would imply a different (shifted) weighting of the errors, resulting in a different value of $c_{t,k}$. If desired, a different prediction error filter $A_{t,k}(Z)$ can be constructed for each $t = k, k+1, \cdots, n-1$. For $\lambda = 1$, $c_{t,k}$ will be constant with respect to t.

The choice of exponential weighting in equation (A3) leads to a computationally efficient method for computing each of the $c_{t,k}$ for $t = k, k+1, \cdots, n-1$. Consider just the numerator of equation (A3) and, for now, drop the subscript k. Define

$$N_t \equiv 2\sum_{s=k}^{n-1} \lambda^{|t-s|} f_s b_{s-1}$$

$$= 2\sum_{s=k}^{t-1} \lambda^{t-s} f_s b_{s-1} + 2\sum_{s=t}^{n-1} \lambda^{s-t} f_s b_{s-1}$$

$$\equiv N_t^- + N_t^+$$

For the first sum

$$N_{t+1}^- = 2\lambda\sum_{s=k}^{t} \lambda^{t-s} f_s b_{s-1}$$

$$= 2\lambda(f_t b_{t-1} + \sum_{s=k}^{t-1} \lambda^{t-s} f_s b_{s-1})$$

$$= \lambda(2 f_t b_{t-1} + N_t^-) \quad ; \quad N_k^- = 0$$

and, likewise, for the second sum

$$N_{t-1}^+ \equiv 2\lambda \sum_{s=t-1}^{n-1} \lambda^{s-t} f_s b_{s-1}$$

$$= 2\lambda(\lambda^{-1} f_{t-1} b_{t-2} + \sum_{s=t}^{n-1} \lambda^{s-t} f_s b_{s-1})$$

$$= 2 f_{t-1} b_{t-2} + \lambda N_t^+ \quad ; \quad N_n^+ = 0$$

In the same manner, the denominator of equation (A3) can be split into two terms, each with its own recursion:

$$D_t \equiv \sum_{s=k}^{n-1} \lambda^{|t-s|} (f_s^2 + b_{s-1}^2)$$

$$= \sum_{s=k}^{t-1} \lambda^{t-s} (f_s^2 + b_{s-1}^2) + \sum_{s=t}^{n-1} \lambda^{s-t} (f_s^2 + b_{s-1}^2)$$

$$\equiv D_t^- + D_t^+$$

The recursions are

$$D_{t+1}^- = \lambda(f_t^2 + b_{t-1}^2 + D_t^-) \quad ; \quad D_k^- = 0$$

$$D_{t-1}^+ = f_{t-1}^2 + b_{t-2}^2 + \lambda D_t^+ \quad ; \quad D_n^+ = 0$$

In summary, the order (k) recursions are

$$f_{t,k} = f_{t,k-1} - c_{t,k} b_{t-1,k-1}$$

$$b_{t,k} = b_{t-1,k-1} - c_{t,k} f_{t,k-1}$$

where (replacing the subscript k)

$$c_{t,k} = \frac{N_{t,k}}{D_{t,k}} = \frac{N_{t,k}^- + N_{t,k}^+}{D_{t,k}^- + D_{t,k}^+}$$

and the time (t) recursions are

$$N_{t+1,k}^- = \lambda(2 f_{t,k-1} b_{t-1,k-1} + N_{t,k}^-) \quad ; \quad N_{k,k}^- = 0$$

$$N_{t-1,k}^+ = 2 f_{t-1,k-1} b_{t-2,k-1} + \lambda N_{t,k}^+ \quad ; \quad N_{n,k}^+ = 0$$

$$D^{-}_{t+1,k} = \lambda(f^2_{t,k-1} + b^2_{t-1,k-1} + D^{-}_{t,k}) \quad ; \quad D^{-}_{k,k} = 0$$

$$D^{+}_{t-1,k} = f^2_{t-1,k-1} + b^2_{t-2,k-1} + \lambda D^{+}_{t,k} \quad ; \quad D^{+}_{n,k} = 0$$

A Fortran77 subroutine to perform time-variable prediction error filtering using the above equations is provided below. Note that one need not compute the time-variable prediction error filters $A_{t,k}(Z)$ to compute the prediction errors f_t (i.e., the "deconvolved" time series).

```
c subroutine to perform time-variable prediction error filtering
c       via a modified Burg algorithm
c SYNOPSIS:
c call tvburg (y,n,k,lambda,f,b,nm,dm)
c PARAMETERS:
c inputs:
c       y       input time series
c       n       length of y,f,b,nm,dm arrays
c       k       length of prediction error filter
c       lambda  adaptation rate parameter - lambda = 1 implies no adaptation,
c               lambda = .5 implies extremely rapid adaptation, a typical
c               value might be lambda = .97.
c outputs:
c       f       prediction error filtered y (forward prediction errors)
c       b       workspace (backward prediction errors)
c       nm      workspace (necessary to avoid roundoff errors)
c       dm      workspace (necessary to avoid roundoff errors)
c
        subroutine tvburg (y,n,k,lambda,f,b,nm,dm)
        implicit undefined (a-z)
        integer n,k
        real y(n),f(n),b(n),nm(n),dm(n),lambda
        integer i,j
        real c,np,dp,fold

        do 10 i = 1,n
        f(i) = y(i)
10      b(i) = y(i)
        do 30 j = 2,k
        nm(j) = 0.
        dm(j) = 0.
        do 20 i = j,n-1
        nm(i+1) = lambda * (nm(i) + 2. * f(i) * b(i-1))
20      dm(i+1) = lambda * (dm(i) + f(i) * f(i) + b(i-1) * b(i-1))
        np = 0.
        dp = 0.
        do 30 i = n,j,-1
        np = lambda * np + 2. * f(i) * b(i-1)
        dp = lambda * dp + f(i) * f(i) + b(i-1) * b(i-1)
        c = (nm(i) + np) / (dm(i) + dp)
        fold = f(i)
        f(i) = fold - c * b(i-1)
30      b(i) = b(i-1) - c * fold
        return
        end
```

RESPONSIBILITIES - Sept. 1981

Problem	Primary Responsibility	Backup Responsibility
crashes	Jeff	Larry/Rick/Hale/Chuck
AP	Larry	Jeff
S.I.	Larry=S0	Rick=S1
air conditioning	Kai	Hale
RP06,CDC air filters	Kai/Alfonso	Pat
RP07, getdev	Jeff	Hale/Chuck
Add-on memory	Alfonso	Chuck
plotters	Chuck	Larry
line printer	Bert	Hale
Gigis	Jon/Pat	-
Datamedia	Pat/Kai	-
backups	Hale	Larry
AED	Chuck	Rick
tape drive	Bert	Rick
tape archives	Rick	-
cleaning - room 467	Okaya	COCORP
cleaning - room 471	Thierry	COCORP
plot program maintenance	Rick	Jon