

## RUNNING MEDIAN FILTERS AND A GENERAL DESPIKER

BY JOHN R. EVANS

Running mean filters have long been used as time domain, digital, low-pass filters. They are a convolution in the time domain with a boxcar function (or, more generally, with functions designed for better frequency characteristics). Running means are linear and easily described by their transfer functions in the frequency domain. They handle some kinds of high-frequency contamination well. However, faced with the spikey noise sometimes affecting electronic systems and phenomena subject to interference by lightning, cosmic rays, and timing tick marks running means usually perform poorly: spikes become low, wide, one-sided pulses affecting a broad region of the signal in the neighborhood of an originally short glitch. Since we can be generally confident of the signal except *during* the spike itself, a better filter would be one that does not affect the signal outside the spike, but replaces only the bad data with some interpolation of the good data. Stated another way, this filter would do what the human data editor does. Running *medians* nearly behave like this ideal filter; a simple extension of them is an ideal, general despiker.

Claerbout and Muir (1973) discussed the use of medians and the related  $L_1$  norm (which minimizes the sum of absolute errors) for a number of applications. Rabiner *et al.* (1975) applied running median filtering to speech processing problems, and Tukey (1977), apparently the originator of these methods, discussed them in statistical applications including scatter plot smoothing. The rarity of published uses of these nonlinear techniques in the geophysical literature prompts the following discussion. This letter discusses running median filter (RMF) "transfer functions," their similarity to the "human filter," and their special qualifications for filtering coded time signals. The use of RMFs only to identify spikes (which can then be removed by an appropriate interpolation) is also discussed.

### DEFINITIONS

An RMF of length  $m$  replaces each point in the time series with the median value in an  $m$ -point window of the original series centered on the current point. This letter largely deals with odd-length RMFs for which "middle point" is unambiguous in both time and amplitude.

When even-length windows are used, the median is arbitrarily defined as the *mean* of the *two* middle points. This definition implies a decentering of the time series by one-half sampling interval (to a point mid-way in time in the window). This decentering in time causes a phase shift which is proportional to frequency; the problem is circumvented by using even-length running medians (also even-length running means) in "complementary" pairs—the first RMF arbitrarily assigning the results to the point one-half sample interval too late, the second assigning the result one-half interval too early or *vice versa*. The final result does not exhibit phase problems.

### TRANSFER BEHAVIOR

Figure 1 compares the behavior of a 7-point running median and 7-point running mean on a mixture of white noise and spikes. Although both are good low-pass filters, the running median removes the spikes more effectively than the running mean. Less obviously, the running median produces a choppy-looking trace with

many small plateaus and sharp corners. The latter effect is not troublesome when there is a large difference in frequency between "signal" and this "numerical noise."

RMFs have no true transfer functions since they are nonlinear; strictly speaking, they behave differently for each new time series. However, their "transfer behavior" is essentially predictable. Figure 2 presents the power and phase response of a 5-point running median and of a running mean of the same length to roughly Gaussian, white noise. "Gaussian" refers to the distribution of points about the mean of the trace, and "white" to a constant power spectrum. Figure 2 was generated by ensemble averaging 15 individual estimates of the transfer behavior which were obtained by comparing spectra of input and output time series.

The point of interest is the similarity of the two functions. The transfer behavior

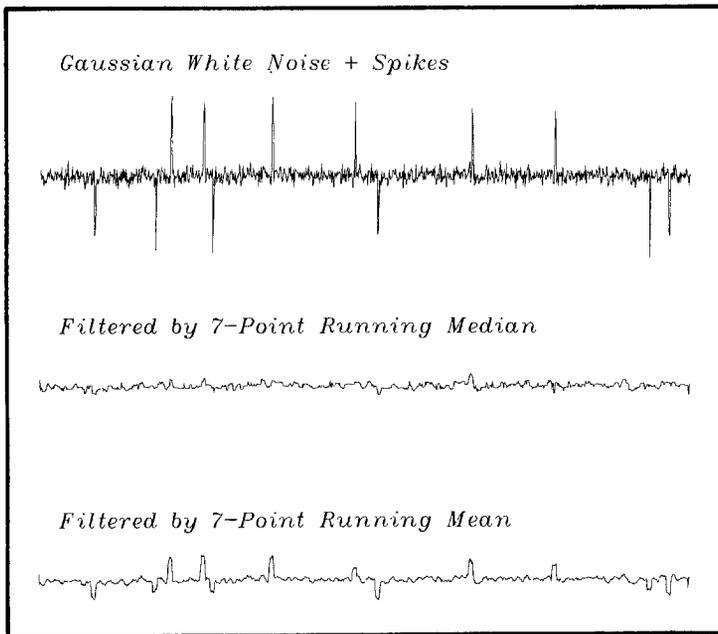


FIG. 1. An example of running medians and running means on a combination of Gaussian white noise and brief, random spikes. From top to bottom: 1024-point noise sample; sample filtered by running median of 7; sample filtered by a running mean of 7.

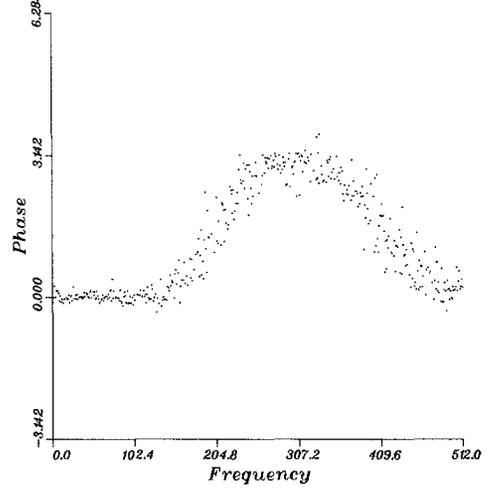
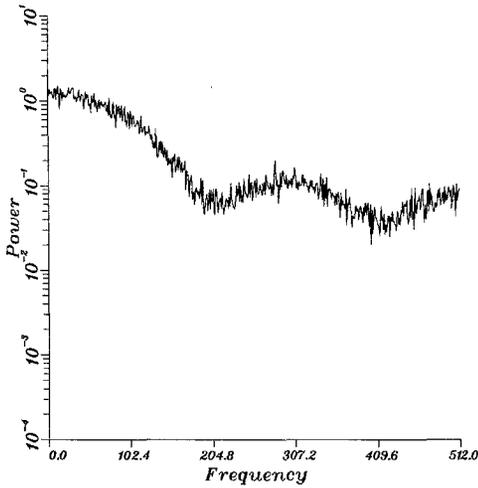
of running medians is very similar to, but has somewhat higher variance than, the transfer behavior of running means in all test cases explored. The power passed by the running median is higher at all frequencies than that passed by the running mean. The latter effect is larger at higher frequencies; at zero frequency the running median slightly exceeds unity transfer.

The variance of the running median transfer behavior is in direct proportion to the variance of the spectrum of the noise used in tests and depends on the distribution of the signal in the frequency domain. Tests with natural seismic noise samples show smoother responses at low frequencies (when substantial energy is present at those frequencies) and higher variance behavior at frequencies where the signal has less energy. By comparison with running means in the same situations, these effects are largely numerical.

The higher variance and higher power spectrum of the running median are probably due in part to modulation of energy from one frequency to another.

Velleman (1975) found that modulation is a major effect when simple (high-frequency) signals are filtered with RMFs, but that modulation is less significant for broad-spectrum noise and with even-length running medians. He also found that the transfer behaviors of even-length running medians used in complementary *pairs*

*White Noise Transfer Behavior  
of 5-Point Running Median*



*White Noise Transfer Behavior  
of 5-Point Running Mean*

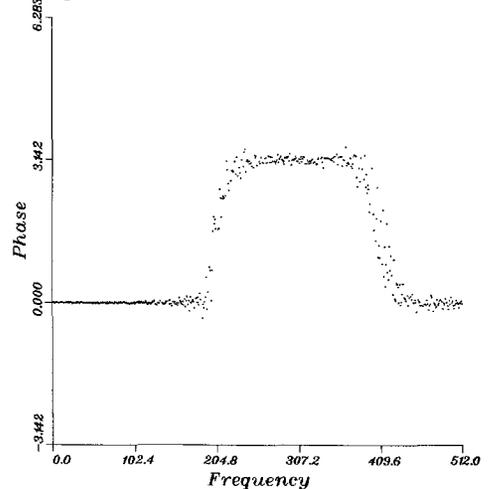
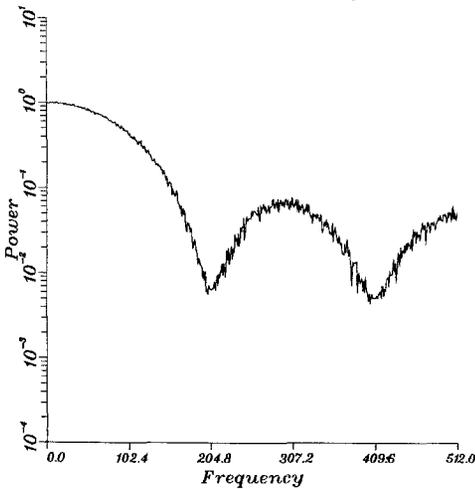


FIG. 2. "Transfer behavior" of 5-point running medians and running means on 1024-point synthetic Gaussian white noise samples. These are ensemble averages of 15 separate estimates. Note overall similarity of the filters and higher variance and side lobes of the nonlinear filter. Frequency 512 is the Nyquist, and phase is in radians.

have lower side lobes than of those of single odd-length RMFs. The present work suggests that the same reduction of side lobes occurs when odd-length RMFs are paired. The resulting transfer behavior is roughly the product of the two separate "behaviors," although the advantage gained is not as large as with running means.

RMFs and compounds of these filters behave much like their linear counterparts

when used to filter broad-spectrum noise and typical natural noise samples. RMF behavior is somewhat more erratic than that of linear filters of the same length, and side lobes are slightly higher. However, RMFs have important advantages when faced with spikey noise. Any large excursion of the time series whose duration is no larger than  $(m - 1)/2$  samples, where  $m$  is the window length of the filter, will be replaced by the nearest well-supported values. A 3-point RMF will eliminate 1-point spikes; a 7-point RMF eliminates 1-, 2-, or 3-point spikes, a 6-point RMF eliminates 1- or 2-point spikes. As defined above, even-length RMFs will round off spikes longer than  $(m - 2)/2$  points, because a mean of two points is involved, whereas odd-length RMFs leave wide spikes unbroadened and often clipped. This despiking is the main reason for using RMFs on geophysical data; they are rapid, automated data editors.

#### FILTERING CODED SQUARE-WAVES

The following discussion applies to any sequence which is monotonic within every  $m$  length segment of the series or constant within  $(m + 1)/2$  length segments; I use the specific example of WWVB radio time code which is often recorded with seismic data.

WWVB is controlled by the Cesium clocks of the National Bureau of Standards and is transmitted at exactly 60 kHz. Radio interference and the electronics involved in reducing this amplitude-modulated signal to a coded square-wave commonly introduce spikes into the code. A human who knows that it should look like can commonly read even very noisy traces; on the other hand, machines have a terrible time with it.

Odd-length RMFs are an exceptionally well-adapted tool for cleaning up this code without introducing systematic errors into the result; they remove the spikes the "human filter" removes. Figure 3 illustrates the effects of running medians and the undesirable effects of running means on a typical sample of noisy WWVB code. A typical window length and an extremely long window length are shown for both filters. The longer window length simply serves to amplify the effects discussed.

All four filters effectively remove spikey noise from trace. However, the running means reduce the sharp changes in signal level, which mark time, to gradual slopes centered on the original steps. A systematic bias is therefore introduced by the linear filter. Countering this bias by using a point half-way down these slopes is incorrect because the exact second is the point at which the trace *just* begins to drop in the original time series (the drop is not instantaneous).

Odd-length RMFs have *no* effect on steps in the absence of noise. In the presence of noise, their worst effect is to draw nearby spikes into the step (usually making it a double step), or to round off the corners when there is a wide distribution of values near the step. These errors are not so systematic as those caused by running means and can be eliminated by further robust reduction (e.g., using the median time between steps for "one second"). Errors introduced into "time of earliest step descent" by rounding off corners will be smaller than those in "time of middle of descent."

The length of the longest RMF window that can be used is determined by the length of the shortest nonnoise segment. If the minimum nonnoise length is  $N$  sec and  $S$  is the sampling interval of the digitizer, then for an odd-length RMF, the largest window length allows is

$$m_{\max} = 2(N/S - 3/2).$$

Any window larger than this theoretical maximum would treat short nonnoise segments as spikes and destroy them.

For example, the shortest pulse of WWVB is 0.2 sec. If the trace is sampled at 200 samples/sec as in Figure 3,  $m_{\max}$  is  $2(0.2/0.005 - 1.5) = 77$ . The long RMF window of Figure 3 is near this theoretical maximum. In practice, noise and computational

*WWVB Time Code (200 Samples per Second)*

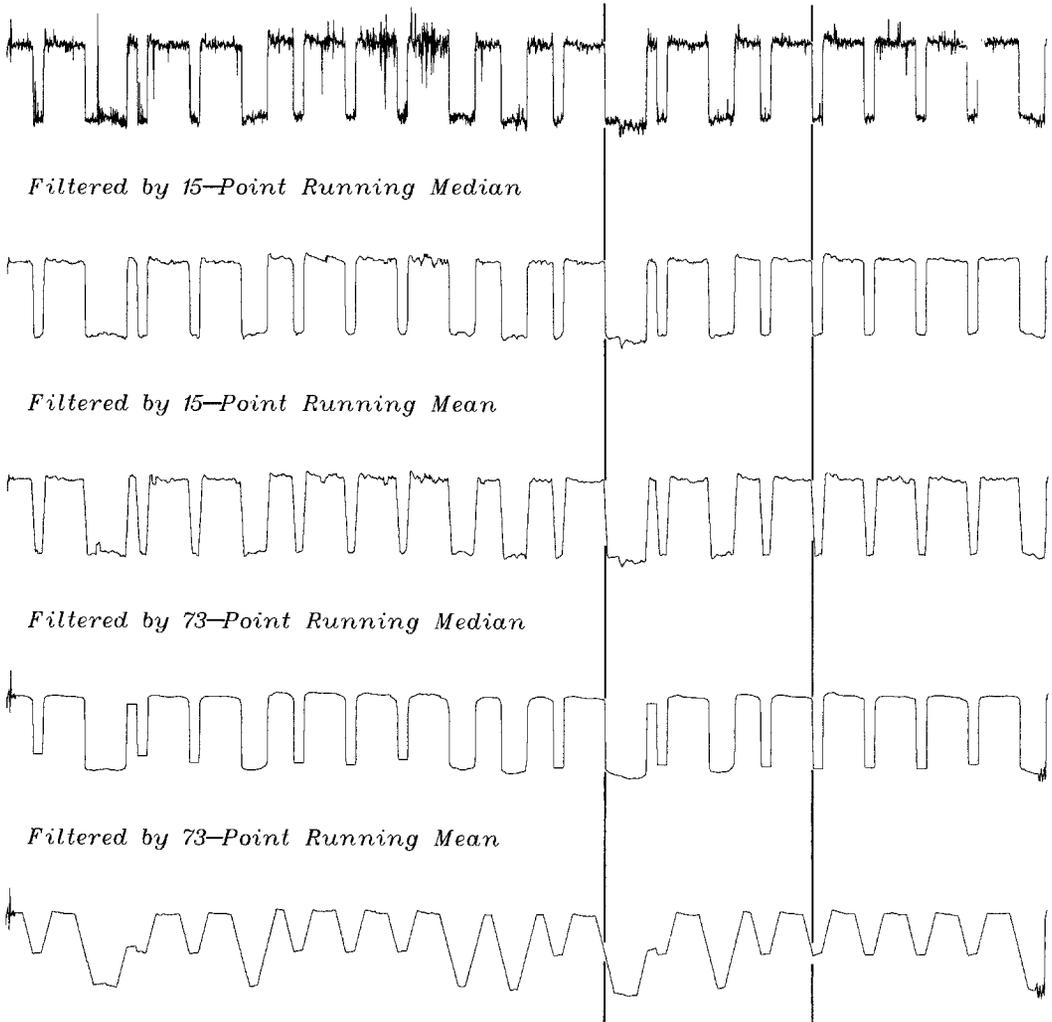


FIG. 3. Effects of running medians and running means on WWVB radio time code. This 20-sec sample was digitized at 200 samples/sec. Note that points "copy on" within a half-window width of either end of traces because the filter output is not defined there. Note slopes introduced by the linear filters where the nonlinear filters leave sharp steps. The two vertical lines aid comparison between traces.

expense may shorten the practical maximum from the theoretical one. Note that only *odd*-length RMFs are appropriate for this kind of time series. Even-length filters involve a mean and will introduce the same sort of slopes as running means.

#### GENERAL DESPIKING ALGORITHM

The "numerical noise" associated with RMFs can sometimes be a worse problem than the original spikes (W. L. Ellsworth, personal communication, 1979). A simple

extension of RMFs produces a “gentle” despiker, one which modifies the trace only where spikes are found. The absolute difference between the original trace and that trace filtered by an RMF is small except where spikes are clipped by the RMF. Comparison of this differential trace to a predetermined threshold allows identification of spikes.

Figure 4 illustrates the effect of a gentle despiker on a real seismogram. A synthetic, random spike train was added to a known signal giving a signal to noise ratio of 1.2 (maximum signal amplitude over maximum spike amplitude). Using a

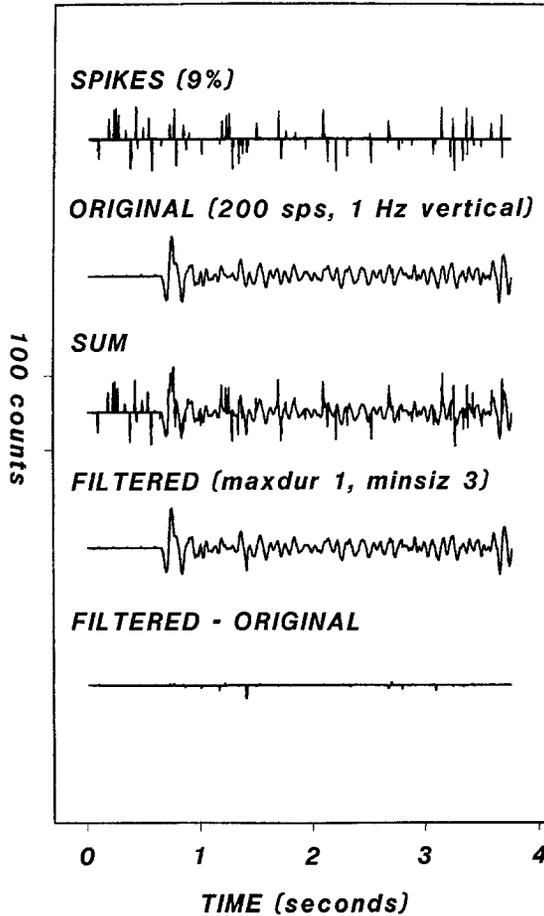


FIG. 4. The effect of general despiking subroutine DESPIK (Evans, 1981) on a real seismogram with synthetic spikes added (*middle trace*). Integral of “filtered-original” is 92 per cent less than the integral of the synthetic spikes. Signal/noise is improved from 1.2 to 2.9.

minimum spike duration of one sample ( $m = 3$ ) and threshold at about 3 per cent of the peak-to-peak signal amplitude, subroutine DESPIK (Evans, 1981) reduced the total error of the trace by 92 per cent [ $\Sigma(\text{filtered-original})/\Sigma(\text{spikes}) = 0.08$ ]. Interpolation across spikes identified by the routine was done with the piecewise continuous cubic polynomial, weighted-average-slope method of Wiggins (1976).

Of the 750 points in the trace, 9 per cent were spikes; 83 per cent of those were identified correctly. Most spikes missed were 2 points wide or smaller than the predetermined threshold. Seventeen good points were incorrectly identified as

spikes; most "false alarms" neighbored real spikes where the spikes fell on slopes of the signal. If a spike goes up in a rising segment of the trace, then the return to normal can look to the routine like a downward spike.

Finally, the DESPIK algorithm should be used as one tool for despiking and should not be considered a panacea for noisy data. Any special knowledge of the data should be used with it to produce despikers specific to each data type.

#### SUBROUTINES AND COMPUTATION TIMES

The nonlinear filtering for this study was done with an old version of subroutine MEDFLT and with subroutine DESPIK (Evans, 1981). The current version of MEDFLT (Evans, 1981) takes advantage of an algorithm by K. R. Anderson (written communication, 1979), which relies on the fact that each new filtered point (after the first one) can be gotten *without* completely sorting that point's neighbors. The sorted set of neighbors for the previous point need only have one old neighbor removed and one new one dropped in at the appropriate place in order to produce a completely sorted set for the new point. This algorithm requires  $O(m)$  ("the order of  $m$ ") steps to execute rather than  $O(m^2)$  for a complete bubble-sorting algorithm. A complete sorting algorithm using the Quicksort method requires  $O(m \log_2 m)$  operations (Hoare, 1962). As implemented by the current subroutine (which does odd-length RMFs only), Anderson's (1978) algorithm was roughly twice as fast as a bubble-sorting algorithm for  $m = 3$  and roughly 5 times faster for  $m = 11$  in bench mark tests I have made.

It should be possible to further reduce execution time by doing a "binary search" of the almost-sorted set of the current algorithm. One would compare the new point to the middle point in a partition of the almost-sorted set, cutting that partition in half at each step. Such an algorithm should take  $O(\log_2 m)$  operations (T. Goforth, written communication, 1981).

Unweighted running means can be calculated by an algorithm requiring  $O(1)$  operations because only one old point need be subtracted and one new one added to an unweighted running mean, regardless of window length, to produce a new output point. MEDFLT is not as fast as this  $O(1)$  algorithm. The running median took about twice as long as an unweighted running mean for  $m = 3$  and about 5 times longer for  $m = 11$  in bench mark tests.

The general despiking subroutine DESPIK has not been optimized for speed. Execution time required to identify spikes is proportional to the number of points in the trace and the window length passed to MEDFLT. Interpolation times are generally proportional to the number of spikes found and the total number of points in the trace.

Velleman and Hoaglin (1978) give FORTRAN subroutines for other kinds of nonlinear filtering including even-length RMFs.

#### CONCLUSIONS

Running medians and running means affect white noise similarly. Both have side lobes at the same frequencies, both can be compounded (several used in sequence) in order to "multiply" their frequency representations, and both are low-pass filters. Running means have slightly lower side lobes, smoother phase and amplitude responses, and analytical representations. Running medians respond more like the human data editor, have far more resilience in the face of spikey noise, and have special properties that are very useful for filtering square-wave data.

The design of robust filters is straightforward because of these similarities to running means and because of an RMF's predictable response to spikes and steps. The desired overall response of the filter can be designed in a linear version (overdesign is wise), and resistance to spikes added by changing at least the leading element of a compound filter to a running median. Preserving steps requires using *only* odd-length running medians and *no* running means.

I hope that this letter stimulates the use of RMFs and other robust techniques. They can be very powerful tools when applied to otherwise problematic data.

#### ACKNOWLEDGMENTS

This work was originally inspired by Jim McBride of Princeton and encouraged by Jason Morgan, Peter Bloomfield, H. M. Iyer, Paul Velleman, Ken Anderson, and many others. I am very grateful for their support. I also wish to thank Bill Bakun, David Oppenheimer, Paul Velleman, Ken Anderson, and Bob Nowack for their helpful reviews.

Part of this work was supported by NSF Grant DES-7515209 (while the author was at Princeton) and the remainder by the U.S. Geological Survey.

#### REFERENCES

- Anderson, K. R. (1978). Automatic processing of local earthquake data, *Ph.D. Thesis*, Massachusetts Institute of Technology, Cambridge, Massachusetts, 173 pp.
- Claerbout, J. F. and F. Muir (1973). Robust modeling with erratic data, *Geophysics* **38**, 826.
- Evans, J. R. (1981). FORTRAN computer programs for running median filters and a general despiker, *U.S. Geol. Surv., Open-File Rept. 81-1091*.
- Hoare, C. A. R. (1962). Quicksort, *The Computer Journal* **5**, 10-15.
- Rabiner, L. R., M. R. Sambur, and C. E. Schmidt (1975). Applications of a nonlinear smoothing algorithm to speech processing, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23, 552.
- Tukey, J. W. (1977). *Exploratory Data Analysis*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Velleman, P. F. (1975). Robust non-linear data smoothing, Technical Report No. 89, Series 2, Department of Statistics, Princeton University, Princeton, New Jersey.
- Velleman, P. F. (1977). Robust non-linear data smoothers: definitions and recommendations, *Proc. Natl. Acad. Sci., U.S.A.* **74**, 434.
- Velleman, P. F. (1978). Definition and comparison of robust non-linear data smoothing algorithms, Economic and Social Statistics Technical Report Series, (778/017), New York State School of Industrial and Labor Relations, Cornell University.
- Velleman, P. F. and D. C. Hoaglin (1978). Implementation of some resistant non-linear smoothers, Economic and Social Statistics Technical Report Series, (878/018), New York State School of Industrial and Labor Relations, Cornell University.
- Wiggins, R. A. (1976). Interpolation of digitized curves, *Bull. Seism. Soc. Am.* **66**, 2077.

U.S. GEOLOGICAL SURVEY  
MS 77  
345 MIDDLEFIELD ROAD  
MENLO PARK, CALIFORNIA 94025

Manuscript received August 14, 1981