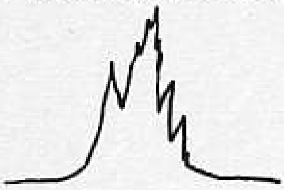
SIGNAL PROCESSING FOR AESTHETIC REASONS

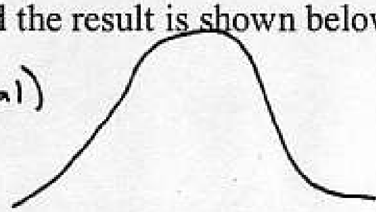
Today's talk confronts the fundamental issue of the difference between what is real and what you are allowed to see. Consider the following signal.

smooth

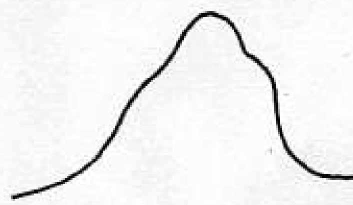


It appears to be a nice collection of three peaks. But what if you only wanted to see one peak? Is there a way to PROCESS the SIGNAL to make only one peak appear? Sure. It is called SMOOTHING and the result is shown below.

(increase signal)

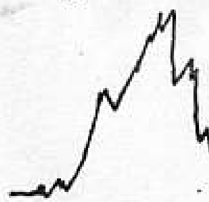


Or maybe you have what appears to be one peak



sharpen

But what if you want to see three peaks? You can perform the opposite mathematical process and ENHANCE the RESOLUTION, and boom, three peaks:



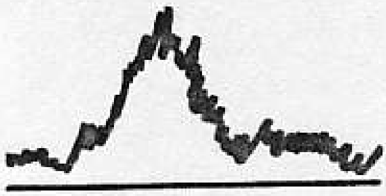
(increase resolution)

Now maybe you don't think it is right to have these kinds of SIGNAL PROCESSING to alter the original signal you see. The fact is that you NEVER see the original signal these days what with computers rearing their awesome power. Be aware that before data gets to you, someone has messed with it to improve the AESTHETICS of the data.

Maybe you think this doesn't apply to you. But how often have you decided to alter the knobs on your stereo to turn up the bass? You are changing the original data. And how often have you asked the photo developers to remove the red dot in the eyeball when you were having your pictures developed? You are changing the original data. Again, you are altering the data for AESTHETIC reasons.

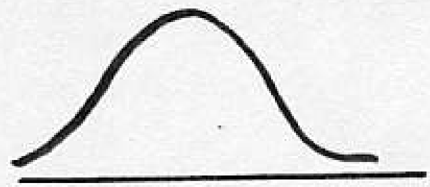
Increasingly as you work with digital images on the computer, you will be given numerous SIGNAL PROCESSING options. Behind every one of these is the process of either improving the signal, improving the resolution, or altering the shape of the data. You need to be able to understand the mathematical process behind this work. As a first effort, we will look at the simple process of SMOOTHING a signal, of removing the noise and increasing the SIGNAL

TO NOISE. Next week we will look at a more general collection of techniques that involve the Fourier transform.



Bad S/N

smoothing →



Good S/N

SMOOTHING DATA

Today we examine the technique of data smoothing: the manipulation of previously collected data to improve the signal-to-noise of the data. Recognize that smoothing is the most common example of cosmetic surgery on data. Yet in the absence of new information, the improved appearance of the data must come at the expense of resolution (increased broadening of the peaks).

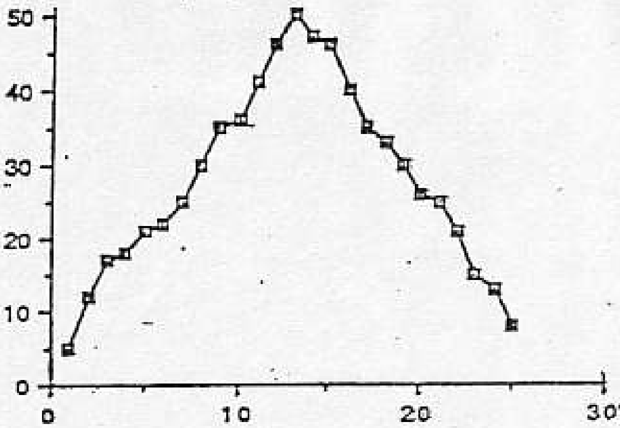
The particular approach to smoothing we will study uses a process involving convolution integers in a MOVING WINDOW or MOVING AVERAGE. This method is a standard component of most data processing software packages including Excel. In learning about the MOVING WINDOW SMOOTH, it will be useful to note that the mathematical foundation borrows from the LINEAR LEAST SQUARES algorithm for modeling we just learned. This is because the MOVING AVERAGE is based upon the assumption that the data to be processed fits a certain function (for example, a second order polynomial). This means that if the true lineshape for our data isn't really a quadratic lineshape (and most are not), then we are actually distorting the data!! Thus the sole purpose for smoothing the data is to make it look pretty.

Technique 1. Boxcar Average

The simplest approach to smoothing is boxcar averaging in which data is divided into groups and each group yields a single centered data point. Random noise is averaged to zero as the number of points increases. However, lost in the process is a great deal of resolution, since many fewer data points are actually plotted.

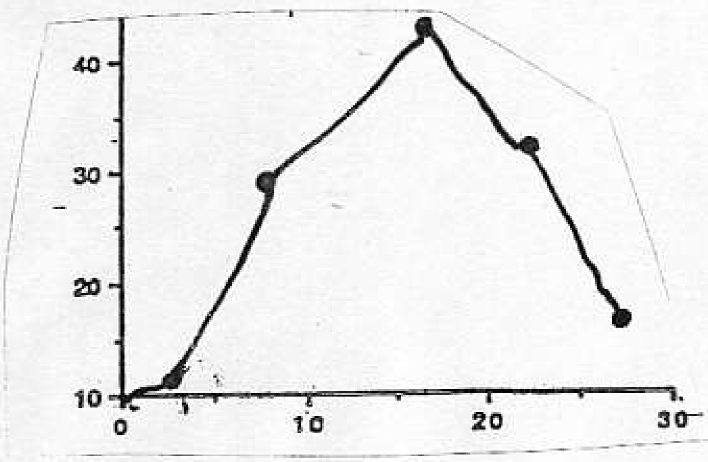
Example: Boxcar of width = 5 applied to 25 raw data points.

|5,12,17,18,21,|22,25,30,35,36|41,46,50,47,46|40,35,33,30,26|25,21,15,13,8|



Notice that in this data there appears to be a single broad peak, but the ebbs and flows in the slope suggest either a noisy signal or the presence of more peaks. What can be done to get rid of the noise? AVERAGE it out using a BOXCAR AVERAGE. Boxcars are easy, just take a running sequence of numbers and add them together to get a single data point that averages out the noise. In the case below, a five-point BOXCAR is added together, in other words, every five data points are added together.

| 10.6 | 29.6 | 46.0 | 32.8 | 16.4 |



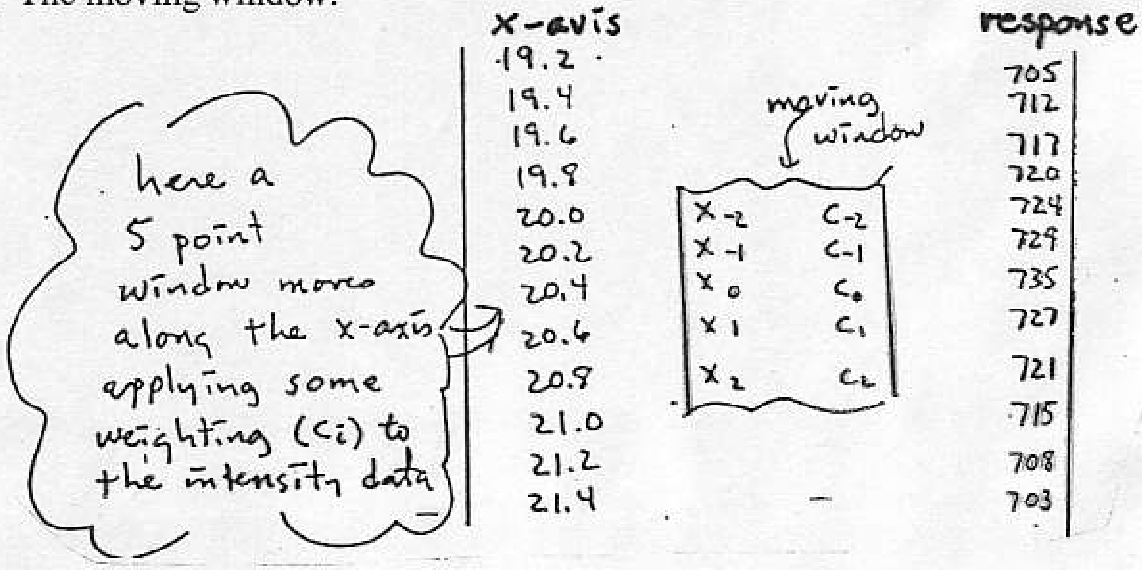
Note that the result is a smoother, less noisy result. But notice how much information is lost in the process.

Technique 2. Moving Window Average (This is so common you can do it in Excel).

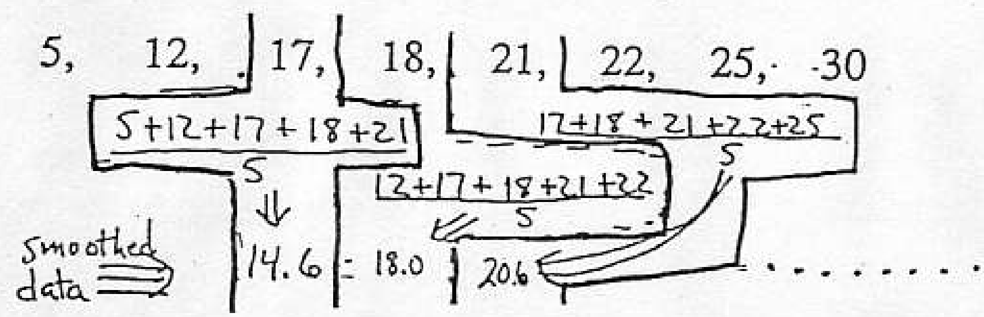
A better approach to smoothing makes use of a moving average in which for a fixed number of points, the y-values are summed and divided by the number of points to obtain an average. However rather than dropping all the data points as in the boxcar, only the data point at the end of the group is dropped and the next point in the sequence is added to calculate a new average. You can see why this is referred to as a running average or moving window average. This approach has the advantage that a new data point replaces the old data point (a one-for-one trade) so that some potential exists for retaining resolution as we smooth.

What is being described is the moving window is the basis for CONVOLUTION as it occurs in nature. CONVOLUTION is a special way that NATURE mixes or folds together two functions. It is really important. So as you watch people in front of the classroom walking by and shaking hands, realize this is how nature mixes functions together too!!

Figure 1. The moving window.



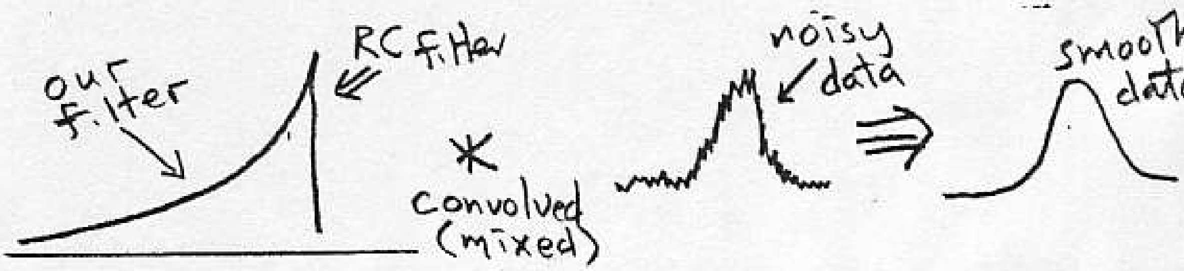
In Figure 1, the C_i's represent convolution integers (actually weighting coefficients, a_i, just like in LEAST SQUARES) and are applied over a set of abscissa values here ranging from -2 to +2 in a 5-point window. In its simplest case, all the coefficients equal one. Thus we have an UNWEIGHTED MOVING WINDOW. To perform a convolution of the ordinate numbers with convolution integers C_i, here equal to 1, the products of $C_i = 1$ times the ordinate value are added together and divided by 5. Let's apply this UNWEIGHTED MOVING WINDOW SMOOTH to our original data set:



Weighted Moving Window Average

We've seen that it is possible to apply a moving average and that one approach is to use an unweighted filter (all the coefficients are 1.) Surely this isn't the best possible set of coefficients. Is there a choice of moving average coefficients that works best for our data? The answer is yes, and it depends on what kind of model function describes our data. So if we think we have a quadratic equation, we can apply weighted coefficients for our model function. If we think the model function is an exponential, we can apply a model function for an exponential. Do we have to know the coefficients--of course not, the software we are using will select them for us. However, in an effort to take you behind the computer screen, presented below are the coefficients used in a weighted moving window for a parabola.

As the example on the last page of these notes illustrates, the use of $C_i=1$ tends to blunt the end of the peak. Unweighted coefficients are obviously not the best choice. Consider other possibilities: the exponential which



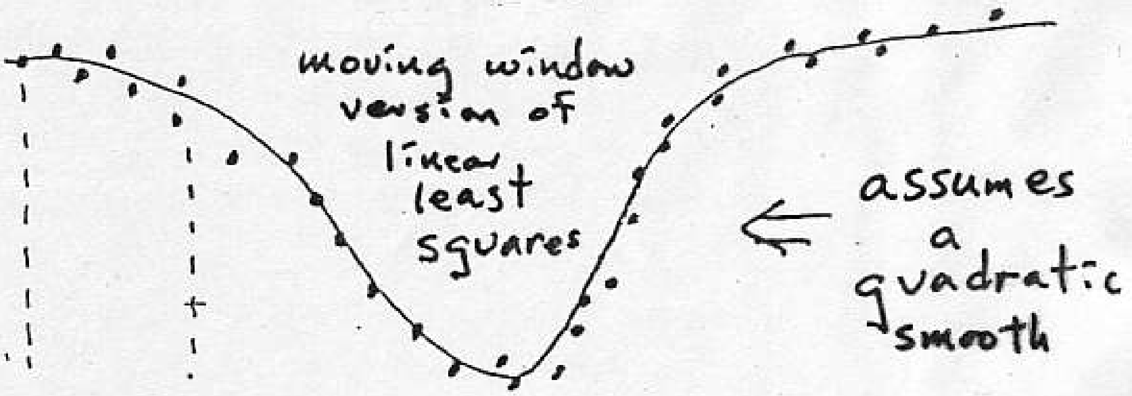
simulates an RC analog time constant. Here the most recent data point is given the greatest weight. Future data have no influence. Thus a unidirectional distortion of the data is introduced. Did you realize that the RC time constant used to filter noisy data (like noisy reception in your stereo or AM radio), is a CONVOLUTION function!! Of course it is, because convolutions are how nature mixes things.

Now in contrast with the exponential smooth applied by an RC filter during data acquisition an obvious advantage of data processing (AFTER DATA ACQUISITION) is the ability to base data treatment on future as well as past events. We can make use of the convolution functions such as exponents, two side exponents, and triangle functions for smoothing purposes. Each, like the moving window, introduces distortion either through broadening shifts in peak maximal, or reduction in peak intensity.

Is there some best convolution function? If I plotted data by hand and tried to fit a line to it, drawing my curve as close to the data points as I could, I would be trying to obtain some best fit of the data. But we already know what this (MODELING) and how to do it mathematically (LEAST SQUARES). In particular, we've learned how to do this for polynomials of the form:

$$y = a_0 + a_1 x + a_2 x^2 + \ldots$$

In this case the coefficients, a_i, are equivalent to the coefficients, c_i, used in the moving window smooth.



Shown above is a pictorial representation of the attempt to apply a seven-point smooth of a quadratic function to data. In each case as we move along the data, we create a new window in which we make the assumption that our raw data, in the absence of noise, is part of a quadratic function. Now if this is really true, then application of the quadratic function will yield our TRUE LINE SHAPE. If it is not true, we are actually distorting our lineshape, although we are making it smoother.

You are probably all wondering how we generate the appropriate coefficients for the moving window. Fortunately, those coefficients are available in software smoothing programs (of course, I could make you produce them using the least squares ideas we developed last week.

As an example, the 5-point smooth which assumes the data is a quadratic function ($y=x^2$) uses the integers {-3,12,17,12,-3}. Note how this is applied in a moving window smooth of a line shape that is a true quadratic. Applying the moving window does nothing more than return the data with which we started. This makes sense, of course, because if we had no noise, then the weighted averaging should yield a quadratic using the weighting functions for a quadratic.

X	Y	Y*	mw 1	mw 2	mw 3	mw 4	5pt quadratic
-4	16	-	-3				
-3	9		12	-3			
-2	4	4	17	12	-3		-48+108+68+12-0/35=4
-1	1	1	12	17	12	-3	-27+48+17+0-3/35=1
0	0	0	-3	12	17	12	-12+12+0+12-12/35=0
1	1			-3	12	17	
2	4				-3	12	
3	9					-3	
4	16						

LOOK, our convolution (smooth) is exactly equal to the raw data!

note our raw data is $y=x^2$, i.e. there is no noise!!

In our remaining example, we apply the 5-point quadratic smooth to our original noisy data and compare it to unweighted and to 9-point smooths.

Calculation of 5-point Moving Window Smooth for a Quadratic Function

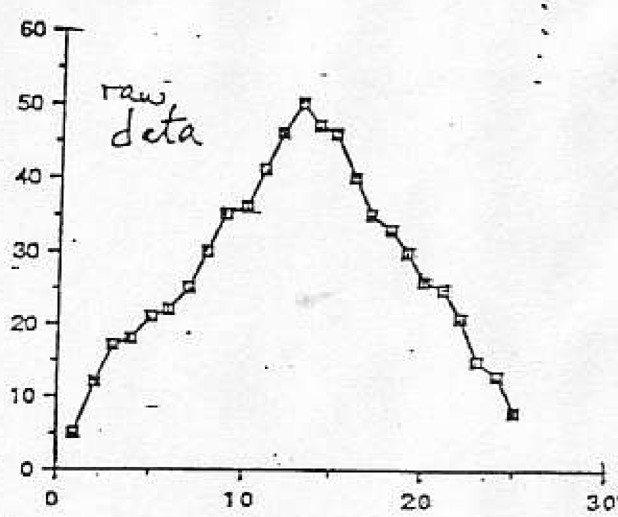
the raw data	5pt. weighted quadratic smooth
5	
12	
17	16.3
18	18.9
21	20.3
22	22.3
25	25.2
30	30.2
35	34.0
36	37.0
41	40.8
46	46.4
50	48.7
47	48.4
46	44.9
40	40.3
35	35.5
33	33.3
30	29.7
26	26.9
25	24.0
21	20.6
15	16.1
13	
8	

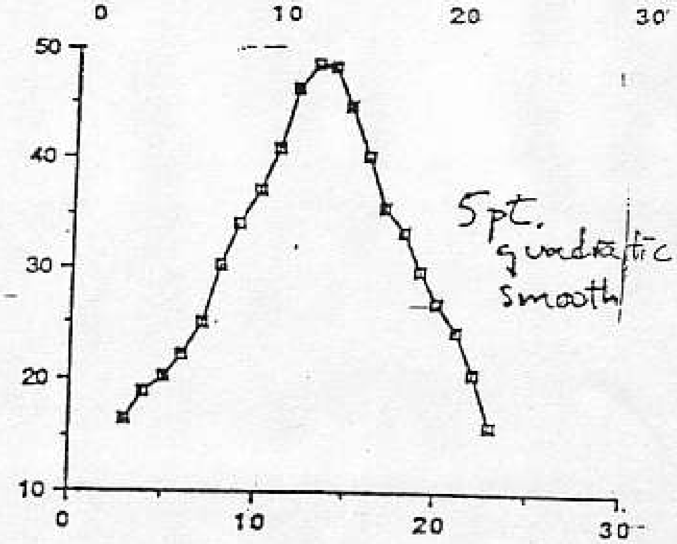
Weights: -3, 12, 17, 12, -3

math examples for first two smoothed points

$$16.3 = \frac{[(-3)(5) + (12)(12) + (17)(17) + (12)(18) + (-3)(21)]}{35}$$

$$18.9 = \frac{[(-3)(12) + (12)(17) + (17)(18) + (12)(21) + (-3)(22)]}{35}$$





notice the smoothed nature of the data
<u>and</u> the loss of resolution

Examples of Moving Window Smooths of Original Data.

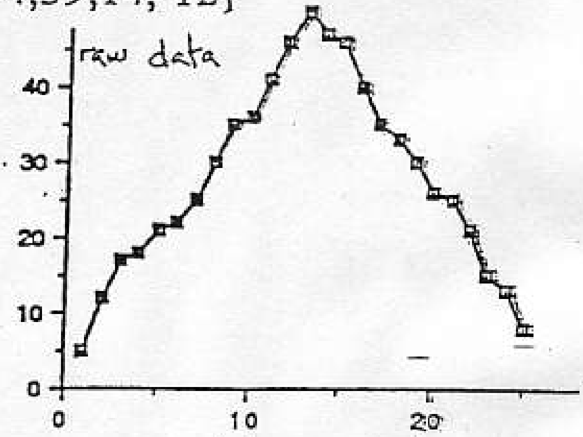
Results are shown for

the unweighted smooth with coefficients = {1,1,1,1,1}

for the 9-point quadratic smooth with

coefficients = {-21,14,39,54,59,54,39,14,-12}

Raw data	unweighted 5pt	quadratic 9pt
5	–	–
12	–	
17	14.6	
18	18.0	
21	20.6	20.6
22	23.2	24.0
25	26.6	27.2
30	29.6	30.4
35	33.4	34.0
36	37.6	36.9
41	41.6	39.6
46	44.0	41.2
50	46.0	41.8
47	45.8	41.6
46	45.6	40.9
40	40.2	39.2
35	36.8	36.9
33	32.8	35.3
30	29.8	30.1
26	27.0	26.4
25	23.4	22.9
21	18.0	
15	16.4	
13		
8		



unweighted
5pt moving window $c_i=1$