13.9 Smoothing of Data

The concept of "smoothing" data lies in a murky area, just beyond the fringe of these better posed and more highly recommended techniques:

- Least squares fitting to a parametric model, which is covered in some detail in Chapter 14.
- Optimal filtering of a noisy signal, see §12.6.
- Letting it all hang out. (Have you considered showing the data as they actually are?)

On the other hand, it is useful to have some techniques available that are more objective than

- Draftsman's license. "The smooth curve was drawn by eye through the original data." (Through each individual data point? Or through the forest of scattered points? By a draftsman? Or by someone who knows what hypothesis the data are supposed to substantiate?)

Data smoothing is probably most justified when it is used simply as a graphical technique, to guide the eye through a forest of data points all with large error bars. In this case, the individual points and their error bars should be plotted on the same graph, and no quantitative claims should be made on the basis of the smoothed curve. Data smoothing is least justified when it is used subjectively to massage the data this way and that, until some feature in the smoothed curve emerges and is pounced on in support of an hypothesis.

Data smoothing is what we would call "semi-parametric." It clearly involves some notion of "averaging" the measured dependent (y) variable, which is parametric. Smoothing a set of values will not, in general, be the same as smoothing their logarithms. You have to think about which is closer to your needs. On the other hand, smoothing is not supposed to be tied to any particular functional form $y(x)$, or to any particular parametrization of the x axis. Smoothing is art, not science.

Here is a program for smoothing an array of ordinates (y's) that are in order of increasing abscissas (x's), but without using the abscissas themselves. The program pretends that the abscissas are equally spaced, as they are if reparametrized to the variable "point number." It removes any linear trend, and then uses a Fast Fourier Transform to low-pass filter the data. The linear trend is reinserted at the end. One user-specified constant enters: the "amount of smoothing," specified as the number of points over which the data should be smoothed (not necessarily an integer). Zero gives no smoothing at all, while any value larger than about half the number of data points will render the data virtually featureless. The program gives results that are generally in accord with the notion "draw a smooth curve through these scattered points," and is at least arguably objective in doing so. A sample of its output is shown in Figure 13.9.1.

```
SUBROUTINE SMOOFT(Y,N,PTS)
    Smooths an array Y of length N, with a window whose full width is of order PTS neighboring
    points, a user supplied value. Y is modified.
PARAMETER(MMAX=1024)          Maximum size of padded array.
DIMENSION Y(MMAX)
```

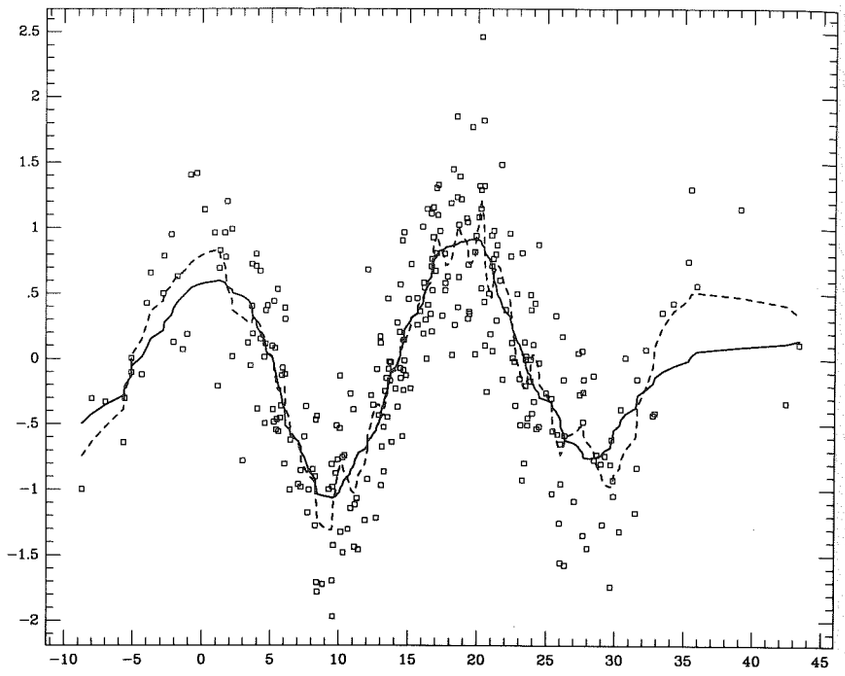

Figure 13.9.1. Smoothing of data with the routine SMOOFT. The open squares are noisy data points, nonuniformly sampled with the greatest sampling density towards the middle of the abscissa. The dotted curve is obtained with smoothing parameter PTS= 10.0 (averaging approximately 10 points); the solid curve has PTS= 30.0.

```
      M=2
      NMIN=N+2.*PTS                    Minimum size including buffer against wrap around.
    1 IF(M.LT.NMIN)THEN                Find the next larger power of 2.
          M=2*M
      GO TO 1
      ENDIF
      IF(M.GT.MMAX) PAUSE 'MMAX to small'
      CONST=(PTS/M)**2                 Useful constants below.
      Y1=Y(1)
      YN=Y(N)
      RN1=1./(N-1.)
      DO 11 J=1,N                      Remove the linear trend and transfer data.
          Y(J)=Y(J)-RN1*(Y1*(N-J)+YN*(J-1))
        11 CONTINUE
      IF(N+1.LE.M)THEN                 Zero pad.
          DO 12 J=N+1,M
              Y(J)=0.
          12 CONTINUE
      ENDIF
      MO2=M/2
      CALL REALFT(Y,MO2,1)             Fourier transform.
      Y(1)=Y(1)/MO2
      FAC=1.                           Window function.
      DO 13 J=1,MO2-1                  Multiply the data by the window function.
```



```
        K=2*J+1
        IF(FAC.NE.0.)THEN
            FAC=AMAX1(0.,(1.-CONST*J**2)/MO2)
            Y(K)=FAC*Y(K)
            Y(K+1)=FAC*Y(K+1)
        ELSE                           Don't do unnecessary multiplies after window function is zero.
            Y(K)=0.
            Y(K+1)=0.
        ENDIF
      13 CONTINUE
      FAC=AMAX1(0.,(1.-0.25*PTS**2)/MO2)   Last point.
      Y(2)=FAC*Y(2)
      CALL REALFT(Y,MO2,-1)            Inverse Fourier transform.
      DO 14 J=1,N                      Restore the linear trend.
          Y(J)=RN1*(Y1*(N-J)+YN*(J-1))+Y(J)
      14 CONTINUE
      RETURN
      END
```

There is a different smoothing technique that we have found useful for data whose error distribution has very broad tails: At each data point, construct a "windowed median" of ordinates, that is the median of that point's ordinate and the ordinates of the $2M$ data points nearest in abscissa, M on each side. Near the edge of the graph, where there are fewer than M points available on one side, instead use more points on the other side, so that your medians are always of $2M$ points. Choose M to taste. The windowed median is not smooth (in the sense of differentiable), but it does fluctuate substantially less than the raw data, and it will often track otherwise noisy trends quite well. It is also genuinely nonparametric, i.e. invariant under reparametrizations of both abscissa and ordinate.

REFERENCES AND FURTHER READING:

Bevington, Philip R. 1969, *Data Reduction and Error Analysis for the Physical Sciences* (New York: McGraw-Hill), §13.1.