Savitzky-Golay filter derivation

1) See also the derivation in Numerical Recipes (available online) section 14.8, p. 651.

2) a nice article is On the Frequency-Domain Properties of Savitzky-Golay Filters

Ronald W. Schafer,  HP Laboratories,  HPL-2010-109

Let's say we want to fit a kth order polynomial to a frame of size F samples, where F is odd.  We can set up a matrix **A** where each column is a polynomial (1st column is all ones, 2nd column is proportional to t, 3rd proportional to t^2, etc).  Then to find the polynomial coefficients c, we must solve

**A** **c** = **y**

Where **A** is size F x k, **c** is kx1, and the data **y** is Fx1.  The solution is:

**c** = (**A**T**A**)-1 **A**T**y** **.**

We could solve for the coefficients directly -for example to calculate curvature from the polynomial coefficients.  However, for fitting  we really don't need the coefficients.  Instead, we are interested in the smoothed fit to the data.  If we call the fit **yhat**, then

**yhat = Ac**

= **A**(**A**T**A**)-1 **A**T**y** (substitute c)

**= B y** (define B)

The matrix **B** above, size FxF, can be calculated ahead of time - it only depends on the polynomial shapes.  This is the matrix calculated by the 'sgolay' command in matlab:  b = sgolay(k,f).

From the equation above, we can see the first element of **yhat** (filter output for first sample) is just the first column of **B** dot-multiplied with **y**.  The second element of **yhat** is the 2nd column of **B** dot-multiplied with **y** , etc.

Assuming our total data length is longer than F, we can use the first (F-1)/2 columns of **B** to handle the left edge, the middle column to handle the middle of the data (away from edges) by just sliding the data window, and the last (F-1)/2 columns of **B** to handle the right edge.