Our previous three columns (1–3) contained discussion of the theoretical aspects of using derivatives in the analysis of spectroscopic data, followed by a discussion of the development of the Savitzky–Golay method of using convolution functions to compute derivatives, concluding with the presentation of a general method to create the set of convolution coefficients for any desired order of derivative, using any degree of polynomial fitting function and number of data points.

When performing quantitative calibrations using a derivative transform, several possible problems can arise. We’ve already noted that one of these is the possibility that the data used to compute the derivative will be affected by interfering materials. We can do little in a column such as this one to deal with such arbitrary and sample-dependent issues. Therefore we will concentrate on those issues that are amenable to mathematical analysis; this consists mostly of the behavior of the computed derivative when there is noise on the data.

Most of our discussion so far has centered on the use of the two-point-difference method of computing an approximation to the true derivative, but since we have already brought up the Savitzky–Golay method, it is appropriate here to consider both ways of computing derivatives, when considering how they behave when used for quantitative calibration purposes.

In fact, the two-point method can be considered a special case of the more general Savitzky–Golay concept because it can be considered the application of the set of convolution coefficients $-1, 0, 1$ to the data. Of course, these convolution coefficients were created ad hoc, and not according to the general scheme that produces the Savitzky–Golay set. Nevertheless, it is convenient to group them together for the purpose of further examination. We are also indebted to David Hopkins for invaluable discussions concerning the properties of the Savitzky–Golay convolution coefficients (4).

In our previous column we derived the expressions for the first and second derivatives of both the normal and Lorentzian band shapes (1). For the following discussion, however, we will address only the normal case because, as we will see, the Lorentzian case will parallel it closely.

In that previous column, we used the standard generic formula for the normal distribution, ignoring the aspect of using it to describe the situation for quantitative analysis. The quantity of concern now is the signal-to-noise ratio (S/N) of the data that we will use to perform the calibration calculations. To deal with this systematically, the S/N must now be divided into two parts: the magnitude of the signal, and the magnitude of the noise. Then different situations can be compared by independently computing the signal and noise contributions to the final S/N that is operative on the calibration.
We start with the simpler case, the signal. By investigating the behavior of the theoretical, ideal derivative, we avoid issues having to do with the different ways an approximation to the derivative can be obtained. The various approximations that can be obtained through the use of constructs such as the Savitzky–Golay convolutions allow us to make trade-offs between maximizing the signal, faithfully reconstructing the true derivative, and creating artifacts, but these issues are all obviated by considering the behavior of the theoretically ideal case. When we come to consider the noise, then as we shall see, the nature of the approximating method becomes very important, but for now we will ignore that.

If the concentration of a material can vary, however, then according to Beer’s law, the absorbance at any given wavelength will also be proportional to \( C \), the concentration. Therefore to take the concentration into account we must modify (including changing the generic \( Y \) variable to \( A \) to indicate absorbance) equations 1a, 6a, and 9a to

\[
A = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

Whereupon the first derivative becomes

\[
\frac{dA}{dX} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

And the second derivative is

\[
\frac{d^2A}{dX^2} =
\]

The “signal” part of S/N that concerns us is the way these expressions vary with the concentration of the analyte. Therefore, from equation 22 we obtain for the absorbance signal

\[
\frac{dA}{dC} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

For the first derivative we obtain

\[
\frac{d}{dC} \left( \frac{dA}{dX} \right) =
\]

And for the second derivative we obtain

\[
\frac{d}{dC} \left( \frac{d^2A}{dX^2} \right) =
\]

As we see from these equations, we have recovered the original expressions for the absorbance and the derivatives with respect to wavelength. The expression we used for the normal curve was the constant-area expression, but the continuation of the derivation for the change of the signal with respect to concentration will follow for the constant-height case, and for the Lorentzian curve, also.

As we saw in reference 1, the maximum value of the first derivative decreases as \( \sigma \) and the second derivative decreases as \( \sigma^2 \) compared with the absorbance, and therefore their derivatives with respect to concentration (which is the sensitivity to concentration changes) also decrease that way.

Therefore we now turn to the “noise” part of S/N. As we saw just above, the two-point derivative approximation can be put into the framework of the Savitzky–Golay convolution functions, and we will therefore not treat them as separate methods.

We have derived previously (5, 6) that the following expression relates the noise on data to the noise of a multiple of that data:

\[
\frac{dA}{dC} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]
Chemometrics

\[
\text{Var}(aX) = a^2 \text{Var}(X) \quad [28]
\]

and, of course, we know that variances add. Therefore, if we have several variables, each of them contaminated with some noise (whose variance is \(\text{Var}(X)\)), and they are multiplied by some constants, then the variance of the result is:

\[
\text{Var}(X_{\text{net}}) = a_1^2 \text{Var}(X) + a_2^2 \text{Var}(X) + a_3^2 \text{Var}(X) + ... \quad [29]
\]

Therefore, if \(X\) represents the spectrum, the various \(a_i\) represent convolution coefficients and \(\text{Var}(X)\) represents a noise source that gives a constant noise level to the spectral values, then equation 29 gives the noise variance expected to be found on the computed resultant value, whether that is a smoothed spectral value or any order derivative computed from a Savitzky–Golay convolution. Equation 29 can be simplified slightly:

\[
\text{SD}(X_{\text{net}}) = \text{SD}(X) \sqrt{a_1^2 + a_2^2 + a_3^2 + ...} \quad [30]
\]

The expression under the radical gives the multiplying factor for the noise standard deviation for the computed derivative (or smoothed spectrum, but that is not our topic here; we will address only the question of the effect on derivatives), and can be computed solely from the convolution coefficients themselves, independently of the effect of the convolution on the “signal” part of S/N.

The nature of the convolution function matters, however, and so do the details of the way it is computed. To see this, let us begin by considering the two-point derivative we have been dealing with in most of this subseries of columns. For our first examination of the effect, let us consider that we are computing the derivative from adjacent data points spaced 1 nm apart (such as in our initial discussion of derivatives [1]).

As we mentioned, the two-point first derivative is equivalent to using the convolution function \([-1, 1]\). We also treated this in our previous column, but it is worth repeating here. Therefore the multiplying factor of the spectral noise variance is \(-1^2 + 1^2 = 2\), and the multiplying factor for the noise standard deviation is \(2^{1/2}\). Similarly, the second derivative is equivalent to a convolution function of \([1, -2, 1]\), making the multiplying factor for the standard deviation of the derivative calculated this way equal \(6^{1/2}\).

Because we have noted earlier that the magnitude of the “signal” parts of S/N \([\text{d}C/\text{d}(\text{d}X/\text{d}A)]\) decreases with increasing derivative order, at this point it would appear that because the signal decreases and the noise increases when you take a derivative, you wind up losing from both parts of S/N.

But things are not so simple. In this examination we have so far looked only at a derivative calculated from adjacent data points. What happens when we calculate a two-point derivative based on non-adjacent data points? We have already considered this question qualitatively in reference 3, where we noted that using the optimum spacing will result in an improved S/N for the derivative. Of course, “improved” in this case is in comparison to the derivative computed using adjacent data points; it must be determined on a case-by-case basis whether the improvement also carries over to the analytical signal.

We can express the improvement semiquantitatively in a graph, as in Figure 11. Here we show true spectrum as the straight line representing the true derivative, and the measured absorbance data as the large Xs. Because the measured data are contaminated with random noise, they do not fall on the line representing the true spectrum. The diagram is set up in such a way, however, that the “noise” on the data from the two wavelengths representing spacing = 1 and spacing = 2 is the same. It is clear from this diagram that the computed approximation to the true derivative is better for the case of spacing = 2, even though the noise is the same.

There are several ways to express this in words. One way is to note that the error is “spread” over a larger \(X\) distance, and therefore has less effect at any one point. Another way is to note that for a derivative computation, the effective “signal” is the value of \(\Delta Y\), and when \(\Delta X = 2\), \(\Delta Y\) is double the value...
of $\Delta Y$ when $\Delta X = 1$. Because the noise is the same, S/N therefore improves with an increase in the spacing. We learned in our prior column (3), however, that the improvement is linear with spacing only at very small values of $\Delta X$, at large values it decreases, levels off, and then eventually starts to get worse again.

From a mathematical point of view, we can let $\delta X$ be the increment between adjacent measurement wavelengths. Then $\Delta X = n \times \delta X$, where $n$ is the number of wavelength increments over which the derivative is calculated. Then, because:

\[
\text{computed derivative} = \frac{\Delta Y}{\Delta X} = \frac{\Delta Y}{n \delta X}
\]  

[31]

Applying equation 28 to find the variation of the computed derivative we obtain:

\[
\text{Var(derivative)} = \frac{1}{n^2} \text{Var} \left( \frac{\Delta Y}{\delta X} \right)
\]  

[32]

and because $\delta X$ is a constant (with an assumed value of unity), and therefore its variance is zero, equation 32 becomes

\[
\text{Var(derivative)} = \frac{1}{n^2} \text{Var} (\Delta Y)
\]  

[33]

Converting to standard deviations:

\[
\text{SD(derivative)} = \frac{1}{n} \text{SD} (\Delta Y)
\]  

[34]

The nature of the values involved caused the entries to be difficult to compare directly, therefore we recomputed them to eliminate the normalization factors, making the coefficients more easily comparable; we present these in Table II. For Table II we also computed the sums of the squares of the coefficients and present them in the last row.

One trend is obvious: the more data included in the computation, the smaller the variance multiplying factor. This is

<table>
<thead>
<tr>
<th>Table I. Some of the Savitzky–Golay convolution coefficients using a quadratic fitting function.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>−4</td>
</tr>
<tr>
<td>−3</td>
</tr>
<tr>
<td>−2</td>
</tr>
<tr>
<td>−1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>Normal factor</td>
</tr>
</tbody>
</table>

The nature of the values involved caused the entries to be difficult to compare directly, therefore we recomputed them to eliminate the normalization factors, making the coefficients more easily comparable; we present these in Table II. For Table II we also computed the sums of the squares of the coefficients and present them in the last row.

One trend is obvious: the more data included in the computation, the smaller the variance multiplying factor. This is
expected for the case of smoothing; we know that the more
data included in even an ordinary running smooth (that is, a
running arithmetic average), the smaller the variance of the
smoothed (averaged) result (reducing as the square root of
the number of data points included in the average). There-
fore it is not surprising to find it also happening with a
weighted average, such as we find with a Savitzky–Golay
smooth.

We see a similar effect from the first derivative; this can
also be considered to be extended from the case of the two-
point derivative, where we showed earlier that the farther
apart the points used are, the smaller the variance of the re-
sulting derivative value. In the case of the Savitzky–Golay
convolution functions, however, the mechanism leading the
reduced variance is slightly different than that of the two-
point derivative. In the Savitzky–Golay case, the reduced
variance is caused by the implicit smoothing effect of com-
puting the function over multiple data points, just as it is in
the case of explicit smoothing.

<table>
<thead>
<tr>
<th>Index</th>
<th>5-point smooth</th>
<th>7-point smooth</th>
<th>9-point smooth</th>
<th>5-point first derivative</th>
<th>7-point first derivative</th>
<th>9-point first derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td>−4</td>
<td>−0.0952</td>
<td>0.06060</td>
<td>−0.0909</td>
<td>−0.10714</td>
<td>−0.06666</td>
<td></td>
</tr>
<tr>
<td>−3</td>
<td>−0.0857</td>
<td>0.14825</td>
<td>0.16883</td>
<td>−0.2</td>
<td>−0.07142</td>
<td>−0.03333</td>
</tr>
<tr>
<td>−2</td>
<td>0.34285</td>
<td>0.28571</td>
<td>0.23376</td>
<td>−0.1</td>
<td>−0.03571</td>
<td>−0.01667</td>
</tr>
<tr>
<td>−1</td>
<td>0.48571</td>
<td>0.33333</td>
<td>0.25541</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.34285</td>
<td>0.28571</td>
<td>0.23376</td>
<td>0.1</td>
<td>0.03571</td>
<td>0.01667</td>
</tr>
<tr>
<td>1</td>
<td>−0.0857</td>
<td>0.14825</td>
<td>0.16883</td>
<td>0.2</td>
<td>0.07142</td>
<td>0.03333</td>
</tr>
<tr>
<td>2</td>
<td>−0.0952</td>
<td>0.06060</td>
<td>−0.0909</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.48571</td>
<td>0.33333</td>
<td>0.25541</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.34285</td>
<td>0.28571</td>
<td>0.23376</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SSK* 0.48571 0.33333 0.25541 0.14 0.03571 0.01667

*See text for the meaning of SSK.

Table III. More Savitzky–Golay convolution coefficients.

<table>
<thead>
<tr>
<th>Index</th>
<th>5-point first derivative with quartic fitting function</th>
<th>5-point first derivative with cubic fitting function</th>
<th>5-point second derivative, quadratic fitting function</th>
</tr>
</thead>
<tbody>
<tr>
<td>−3</td>
<td>0.12987</td>
<td>0.66667</td>
<td>−0.14285</td>
</tr>
<tr>
<td>−2</td>
<td>0.32467</td>
<td>0.66667</td>
<td>−0.14285</td>
</tr>
<tr>
<td>−1</td>
<td>0.56709</td>
<td>0.66667</td>
<td>−0.14285</td>
</tr>
<tr>
<td>0</td>
<td>0.32467</td>
<td>0.66667</td>
<td>−0.14285</td>
</tr>
<tr>
<td>1</td>
<td>−0.12987</td>
<td>0.66667</td>
<td>−0.14285</td>
</tr>
<tr>
<td>2</td>
<td>−0.12987</td>
<td>−0.083333</td>
<td>0.2857</td>
</tr>
<tr>
<td>3</td>
<td>0.56709</td>
<td>0.9027</td>
<td>0.2857</td>
</tr>
</tbody>
</table>

SSK 0.5670 0.9027 0.2857

There are several directions in which the convolutions can
be varied; one is to increase the amount of data used, by
using longer convolution functions as we demonstrated
above. Another is to increase the degree of the fitting polyno-
mial, and the third is to compute higher-order derivatives. In
Table III we present a very small selection of the effect of po-
tential variations.

What can we learn from Table III? We can compare those
sums of squared coefficients with the corresponding one in
Table II using the same number of data points, and either:
1) The same order derivative with a lower-degree fitting
polynomial, or
2) The same degree polynomial, for a lower-order derivative.

For comparison one, we find two cases: seven-point
smooth with quartic versus quartic fitting function, and
five-point first derivative with quartic versus cubic fitting
function. From these two comparisons we find that the
noise of the derivative (of the same order and number of
data points) increases as the degree of the fitting function
increases.

For comparison two, we find one case: five-point first de-
rivative versus five-point second derivative, both using a
quadratic fitting function. Here again, the noise increased
with increasing derivative order.

In fact, we see that the five-point first derivative using a
cubic fitting function will have almost as high a noise level as
the original data. Couple this with the fact we saw above, that
the sensitivity to concentration of the first derivative is re-
duced compared to the sensitivity of the absorbance data it-
self, and we see that in this particular case, depending on the
value of \( \sigma \) for the absorbance band, use of this form of com-
puting the derivative may be worse than using the ab-
sorbance data, while using a different computation, such as a
quadratic fitting function, may be better than the absorbance
data. Therefore, the effect of using derivatives will depend very much, on a case-by-case basis, whether a particular computation will be beneficial or detrimental.

For this reason, readers will find another very interesting exercise to compute the sums of the squares of the coefficients for several of the sets of coefficients, to extend these results to both higher order derivatives and higher degree polynomials, to ascertain their effect on the variance of the computed derivative for extended versions of these tables. Hopkins (7) has performed some of these computations, and has also coined the term RSK/Norm for the $\sum((\text{coeff/normalization factor})^2)$ in the Savitzky–Golay tables. Because here we predivide the coefficients by the normalization factors, and we are not taking the square roots, we use the simpler term SSK (sum squared coefficients) for our equivalent quantity. Hopkins in the same paper has also demonstrated how the two-point computation of derivatives can also have an equivalent value of the RSK/Norm, with results essentially equivalent to the ones we presented earlier. Table III in reference 7, particularly, shows how differences in the application of the derivative computation can cause the noise level of the computed derivative to be either greater or less than the noise of the absorbance spectrum from which they are computed.

**Acknowledgement**

The authors thank David Hopkins for valuable discussions regarding several aspects of the behavior of Savitzky–Golay derivatives, and also for making sure we spelled Savitzky correctly!

**References**