Construction of spline functions in spreadsheets to smooth experimental data

K. Thomas Klasson *

US Department of Agriculture, Agricultural Research Service, Southern Regional Research Center, 1100 Robert E. Lee Boulevard, New Orleans, LA 70124, United States

Received 11 January 2007; received in revised form 1 March 2007; accepted 12 March 2007
Available online 11 May 2007

Abstract

Experimental data are often in need of smoothing without a pre-determined trend line equation. This manuscript demonstrates how spreadsheet software can be programmed to smooth experimental data via cubic splines. Once the trend lines (splines) have been constructed, it is also simple to interpolate values and calculate derivatives and integrals. The formulas to carry out the calculation are listed and explained in the manuscript.

© 2007 Elsevier Ltd. All rights reserved.

Keywords: Smoothing; Splines; Spreadsheet; Derivatives; Integration

1. Introduction

Sometimes it is desirable to display experimental data with a trend line. There are several advantages to constructing smooth curves to indicate trends in data. For instance, if the equations for these curves are determined, interpolation may be done as well as finding derivatives and integrals anywhere along the trend curves. Commonly, the trend line is generated via some type of regression (e.g., a linear regression line or curve) and many commercial experimental data software has this capability but some of the standard software are limited to selected regression functions. In the case of Microsoft Office Excel 2003 (Redmond, WA), the built-in trend/regression types are linear, logarithmic, polynomial (up to sixth-order), power, and exponential; however, more complex (user-defined) equations are missing. SigmaPlot Version 8.02 (by Systat Software, San Jose, CA), another popular software, is far more powerful and has a larger number of built-in functions to which the data may be fitted. However, sometimes it is desirable to just indicate a trend in the data without forcing a particular curve function through the points. In Excel, data points can be connected by straight lines and by a curved line, but we really have no control over the curved line(s) nor can we use the equation of the curve for any calculation (as it is not available to the user). SigmaPlot offers much of the same features, data points can be connected by straight of curved lines but, again, the equations of the curve are not available.

The smooth curves used in Excel passes exactly through the data points which may not be desired if, for example, one desires to indicate an overall trend rather that connecting points. Thus, Excel can draw smooth curves but cannot smooth the experimental data other than with specific regression functions. SigmaPlot offers smoothing mechanisms that allows smoothing using polynomial regression with various weight functions to estimate the potential error within the data but the resulting equation is not displayed, rather sampling data on the curve is revealed. Other more powerful mathematical software such as MATLAB Version 7.3 (by MathWorks, Natick, MA), Octave Version 2.1.73 (a free command line program using a language compatible with MATLAB, University of Wisconsin, Madison), and Mathematica Version 5.2 (Wolfram...
In that case, fewer than programs were too lengthy to be listed [4]. Another exam-
methods of smoothing data with spline functions, but the
and a FORTRAN program could be ordered from the
where the polynomials joined [3]. The method was complex
imental data and calculation of more suitable data points
was further modified to include uncertainties in the exper-
ment. Later, the method
to determine the “best” overall curve. Later, the method
Klaus and Van Ness used an enhanced spline technique [2].

In order to create a curve which does not go exactly through the data points,
the values of \( f \) go exactly through the data points. In an effort to create
and to define the system so that all the constants in the
polynomials, \( f \) are connected with
points, cubic splines are often used. Here, each segment
in order to create smooth transitions between each segment
and to define the system so that all the constants in the
polynomial may be determined. The normal cubic splines
go exactly through the data points. In an effort to create
a curve which does not go exactly through the data points,
Klasson [5].

While there are many computer software that can be
and have been used for programming cubic spline equations,
Excel was chosen in this work because of its familiar-
ity among researchers and students. It offers data storage,
data manipulation, graphical representation, and is avail-
able on most personal computer systems already. Students
in both high school and college are exposed to spreadsheets
as part of their curricula. Educational research has shown
that spreadsheet applications assisted in promoting prob-
lem-solving skills in students and expanding their capa-
bilities [6,7]. While other more powerful software and
programming languages exist (e.g., MATLAB, Octave,
and Mathematica), spreadsheets continues to be success-
fully applied to a variety of scientific principles such as
chemistry [8] and engineering [9,10]. In this work, a spread-
sheet workbook is constructed based on the smoothing
technique described by Klasson [5].

2. Mathematical background

Cubic splines join adjacent data points with a third-
order polynomial as shown in Fig. 1, and with \( N \) number
of experimental data points, \( N - 1 \) number of splines [poly-
nomials, \( f(x) \)] define the overall curve [1]. The point where
two splines meet is sometimes referred to as a node.

In order to find the \( 4(N-1) \) constants for the polynomi-
als, a set of restrictive conditions are defined [1,5]. Briefly,
the values of \( f(x) \); the first derivative, \( f'(x) \); and the second
derivative, \( f''(x) \) at the nodes are set equal for the joining
polynomials. At the endpoints, \( f''(x_1) \) and \( f''(x_N) \) are
set to zero [1] for normal cubic splines or they can be set
to know values or to \( f''(x_2) \) and \( f''(x_{N-1}) \) [2]. With the
above conditions, the equations system is completely
defined with a smooth transition between splines at nodes.
The constants are then found by restructuring the standard
polynomial equation to another form [5]. Ultimately, the
equation system takes the following matrix form:

\[
\begin{bmatrix}
  b_2 & c_2 & \cdots & \cdots & a_N & b_N \\
  a_1 & b_1 & c_1 & \cdots & \cdots & b_2 \\
  \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
  \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
  a_{N-1} & b_{N-1} & c_{N-1} & \cdots & \cdots & b_N
\end{bmatrix}
\begin{bmatrix}
  f''_2 \\
  f''_3 \\
  \vdots \\
  \vdots \\
  f''_{N-1} \\
\end{bmatrix}
= \begin{bmatrix}
  r_2 \\
  r_3 \\
  \vdots \\
  \vdots \\
  r_{N-1}
\end{bmatrix},
\]

where

\[
\begin{align}
  a_{i+1} &= (x_{i+1} - x_i), \\
  b_{i+1} &= 2(x_{i+2} - x_i), \\
  c_i &= (x_{i+2} - x_{i+1}), \quad \text{and} \\
  r_{i+1} &= 6 \left( \frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right). 
\end{align}
\]

The equation system can be easily solved with Gaussian
elimination. Once the values of \( f''(x) \) have been determined;
interpolation, derivatives, and integrals can be done through
the following equations [5]:

\[
f_{i}(x) = f''_{i}(x_{i+1} - x)^3 + f''_{i}(x - x_i)^3
\]
\[
\quad + \left( \frac{y_{i}}{x_{i+1} - x} - \frac{f'_{i}(x_{i+1} - x)}{6} \right)(x_{i+1} - x)
\]
\[
\quad + \left( \frac{y_{i+1}}{x_{i+1} - x} - \frac{f'_{i+1}(x_{i+1} - x)}{6} \right)(x - x_i)
\]

Fig. 1. Cubic spline construction.
\[ f'_i(x) = \frac{f''_{i+1}(x - x_i)^2}{2(x_{i+1} - x_i)} - \frac{f''_i(x_{i+1} - x)^2}{2(x_{i+1} - x_i)} + \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \]
\[
\int_{x_i}^{x_j} f_i(x) \, dx = \left[ (x - x_i)^2 \left( \frac{c''_i((x-x_i)^2)}{24(x_{i+1} - x_i)^2} \right) - \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right]_{x_i}^{x_j}.
\]

If should be noted that in Eqs. (1)–(6), \( f''_i \) denotes the second derivative evaluated at the \( i \)th point’s \( x \)-value (\( x_i \)) using the \( i \)th spline, remembering that \( f''_i \) and \( f''_{i+1} \) are zero. Eq. (5) is simplified when the derivative is evaluated at \( x = x_i \). Likewise in Eq. (6), the expression is less complex when \( x = x_i \) and \( \beta = x_{i+1} \) or when \( x = x_i \) and \( \beta = x \).

The above discussion addresses normal cubic splines through the nodes determined by the experimental data points without smoothing. Smoothing of data can, for instance, be accomplished by the method described by Klasson [5], where two fictitious points [at \((x_0, y_0)\) and \((x_{N+1}, y_{N+1})\)] are first created outside the range of the original experimental data. Then another set of data points is extracted “between” the available points; this new set (denoted mid-points) is used to create splines for smoothed trend lines. The equations for this are as follows:

- **fictitious point 1**
  \[ x_0 = x_1 - 0.05(x_2 - x_1) \]
  \[ y_0 = y_1 + \frac{(y_2 - y_1)(x_0 - x_1)}{(x_2 - x_1)} \]  
  \[ b_{b1} = \frac{-a_i c_{i-1}}{b_{b1-1}} \]
  \[ r_{r1} = \frac{-a_c r_{r1-1}}{b_{b1-1}}. \]

As the second step, the above matrix is further reduced to

\[
\begin{bmatrix}
1 \\
1 \\
\ddots \\
1
\end{bmatrix}
\begin{bmatrix}
l_{f_2} \\
l_{f_3} \\
\vdots \\
l_{f_{N-1}}
\end{bmatrix} =
\begin{bmatrix}
r_{r2} \\
r_{r3} \\
\vdots \\
r_{r_{N-1}}
\end{bmatrix}.
\]

where

\[
for \quad i = N - 1 \quad to \quad 2,
\]
\[
r_{rr_i} = \frac{r_{rr_i-1}}{b_{b_i}} \quad (note \quad that \quad r_{rr_N} = 0). \]

This represents an easily programmable Gaussian elimination [1] and the matrix in Eq. (13) gives the solution to the equation system as \( f''_i \) is equal to \( r_{rr_i} \). We can now begin developing a suitable Excel workbook structure (Fig. 2), which contain two worksheets (titled Main and Calc) in the same workbook.

As suggested in Fig. 2, the raw data of \( x \) and \( y \) is entered in columns A and B of the Main worksheet. Next we must begin entering some of the functions for the calculations into the Calc worksheet, as follows (see Fig. 3):

A2 =0 The A column contain the counter i.
A3 =IF(Main!A3="","",A2+1) If there is not an x-value in the Main worksheet, the cell will be blank, otherwise the counter will be increased by 1.
B2 =Main!A3−0.05*(Main!A4−Main!A3) Calculation of the x-value (\( x_0 \)) of the “low” fictitious data point from Eq. (7a).
At this point, we can copy the formulas in A3, B4, and C4 to A4:A100, B5:B100, and C5:C100, respectively. Eventually, the smoothed y-values at the original x-values will be calculated in column D in the Calc worksheet but it is easier to skip entering the formula until the rest of the Calc worksheet has been explained. Next we can enter the formulas that will construct and solve the matrix (Eq. (1)) and calculate the second derivatives \( f''_i \) for the splines at the nodes.

\[
\begin{align*}
\text{L2} &= \frac{\text{B3} + \text{B2}}{2} \\
\text{M2} &= \frac{\text{C3} + \text{C2}}{2}
\end{align*}
\]

The formulas in E4, F4, G4, H4, I5, J5, and K4 should be copied to E5:E100, F5:F100, G5:G100, H5:H100, I6:I100, J6:J100, and K5:K100, respectively. We have now solved the matrix system in Eq. (1) and have the information needed to interpolate, find derivatives, and integrals for the spline functions through the raw data. However, we lack the smoothing mechanism. We can accomplish this by expanding the calculations that take place in the Calc worksheet. Fig. 4 shows the structure of this expansion.

In the columns displayed in Fig. 4, we define the splines that will trend the experimental data with a smoothness factor of 1. As previously described, the process starts by creating the mid-point values, which will then be the starting points for the smoothed trend lines. The spreadsheet formulas to accomplish this closely follow the same pattern as in the basic cubic spline functions above. The only difference is that we have one more data point to work with due to the addition of fictitious data points followed by creations of mid-points.

\[
\begin{align*}
\text{E} &= \frac{\text{B3} + \text{B2}}{2} \\
\text{M} &= (\text{C3} + \text{C2})/2
\end{align*}
\]
N4 =L4–L3
O3 =2*(L4–L2)
P3 =L4–L3
Q3 =IF($A4='''',0,6*(M4–M3)/(L4–L3)–6*(M3–M2)/(L3–L2))
R3 =O3
R4 =O4–N4/R3*P3
S3 =Q3
S4 =IF($A5='''',0,Q4–N4/R3*S3)
T2 =0
T3 =IF($A4='''',0,(S3
T4:T100, and U4:U100, respectively. As mentioned before, the U column contains the smoothed
values in column U. The formulas in column U deserve a little explanation. This column contains values of y (at
the original raw x-values) which lay on the smoothed trend line defined by the spline functions through the
mid-points. The concept is demonstrated in Fig. 5.

As is noted in Fig. 5, the coordinates (x,y) of the smoothed data point must be calculated via the interpolation
equation described in Eq. (4). Below are the equations that calculate these smoothed data values in the Excel)
worksheet.

\[ \begin{align*}
U2 &= U3+(U4–U3)*((B2–B3)/(B4–B3)) \\
U3 &= (T2*(L3–B3)^3 + T3*(B3–L2)^3)/6/(L3–L2) +
\frac{(M2/(L3–L2)–T2*(L3–L2)/6)}{(L3–L2)–T3*(L3–L2)/6}*(B3–L2)
\end{align*} \]

The formulas in D2 should be copied to D3:D100. Fig. 2 also contains an area in the Main worksheet that is set
aside for the calculation of the y-value (in column G) from an arbitrary x-value in column F. The calculation is fairly
simple according to Eq. (4), once it has been determined which of the splines should be used. This task is also carried
out in the Calc worksheet (see Fig. 7).

The formulas needed to look up which spline need be used (defined by the coordinates of the nodes on both sides
and corresponding second-order derivatives) can be written as follows:

\[ \begin{align*}
AP2 &= \text{IF(Main!F3='''',''', Main!F3)} \\
AQ2 &= \text{INDEX(BS2:BS100,MATCH(AP2,BS2:BS100,1))} \\
AR2 &= \text{LOOKUP(AQ2,BS2:BS100,DS2:DS100)} \\
AS2 &= \text{LOOKUP(AQ2,BS2:BS100,KS2:KS100)} \\
AT2 &= \text{INDEX(BS2:BS100,MATCH(AP2,BS2:BS100,1))} + \text{IF(AP2=MAX(Main!A$3:A$100),0,1)}
\end{align*} \]

The content of L2, M2, N4, O3, P3, Q3, R3, S4, T3, and U3 should be copied to L3:L100, M3:M100, N5:N100,
O4:O100, P4:P100, Q4:Q100, R5:R100, S5:S100,
T4:T100, and U4:U100, respectively. As mentioned before,
the U column contains the smoothed y-values at the original
x-values as well as the two fictitious x-values, and if we
wanted to find an even smoother trend line, we would use
these smoothed y-values as starting points. With this
knowledge we can complete the rest of the Calc worksheet for smoother curves (with sf = 2 and sf = 3) as shown
in Fig. 6. The formulas that should be entered are as follows:

\[ \begin{align*}
V2 &= L2 \text{ The mid-point x-values remain the same.} \\
W2 &= (U3+U2)/2 \text{ New mid-point y-values are calculated from the smoothed y-values in column U.}
\end{align*} \]

To continue, the formulas in V2 and W2 should be copied to V3:V100 and W3:W100, respectively, and the formulas
in N2:U100 should be copied to X2:AE100. The formulas in V2:AE100 should be then be copied to AF2:AO100 and
this completes all the entries for smoothing the raw (y) data in three iterative steps for smoothness factors of 1, 2 and 3. As noted in Fig. 2, the smoothness factor is entered in cell D1 in the Main worksheet and this
information is needed to select one of the three smoothed
values that were calculated in the Calc worksheet, columns
U, AE, and AO. The formula in column D in the Calc worksheet accomplishes this. It also uses Eq. (10) for a
smoothness factors of less than 1.

\[ \begin{align*}
D2 &= \text{IF(AND(A2='''',A1=''''),0,IF(Main!$DS1=0,C2,IF(Main!$DS1<1,Main!$DS1*U2+(1–Main!$DS1)*C2,IF(Main!$DS1=1,U2,IF(Main!$DS1=2,AE2,AO2))))}
\end{align*} \]

The formula in D2 should be copied to D3:D100. Fig. 2 also contains an area in the Main worksheet that is set
aside for the calculation of the y-value (in column G) from an arbitrary x-value in column F. The calculation is fairly
simple according to Eq. (4), once it has been determined which of the splines should be used. This task is also carried
out in the Calc worksheet (see Fig. 7).

The formulas needed to look up which spline need be used (defined by the coordinates of the nodes on both sides
and corresponding second-order derivatives) can be written as follows:

\[ \begin{align*}
AP2 &= \text{IF(Main!F3='''',''', Main!F3)} \\
AQ2 &= \text{INDEX(BS2:BS100,MATCH(AP2,BS2:BS100,1))} \\
AR2 &= \text{LOOKUP(AQ2,BS2:BS100,DS2:DS100)} \\
AS2 &= \text{LOOKUP(AQ2,BS2:BS100,KS2:KS100)} \\
AT2 &= \text{INDEX(BS2:BS100,MATCH(AP2,BS2:BS100,1))} + \text{IF(AP2=MAX(Main!A$3:A$100),0,1)}
\end{align*} \]
function looks up the raw data x-value, greater than the arbitrary x-value.

\[ AU2 = \text{LOOKUP}(AT2, BS2:BS100, DS2:DS100) \]  
This function looks up the corresponding y-value to the AT2 x-value.

\[ AV2 = \text{LOOKUP}(AT2, BS2:BS100, KS2:KS100) \]  
This function looks up the secondary derivative at the point defined by AT2 and AU2.

\[ AW2 = \left( AS2 \left( \frac{AT2}{C0}\right)^3 + AV2 \left( \frac{AP2}{C0}\right)^3 \right) / 6 / \left( \frac{AT2}{C0} \right) + \frac{AR2 / \left( \frac{AT2}{C0} \right)^2 + \left( \frac{AP2}{C0} \right)^2}{6} \]  
The y-value at the arbitrary x-value is calculated from Eq. (4).

This calculated y-value can now be transferred to the Main worksheet into column G by

\[ G3 = \text{IF}(F3 = '', '', \text{Calc!AW2}) \]  
The formula in cell G3 in the Main worksheet can be copied to G4:G100. The workbook is now just missing formulas for derivatives and integrals. The derivative at the nodes and the integral under the curve between two nodes may be calculated from Eqs. (5) and (6), which result in the formulas in column D and E of the Main worksheet.

The formulas in D3 and E4 should be copied to D4:D100 and E5:E100, respectively. Columns AX and AY in the Calc worksheet (see Fig. 7) contains formulas that will be used for calculating the area under the curve between the first arbitrary x-value and another arbitrary x-value, entered into column F in the Main worksheet. These formulas are

\[ AX2 = AT2 - AQ2 \]  
This calculates the x distance between the two nodes of the spline, on which the arbitrary x-value in column AP is located as this value is needed often in Eq. (6).

\[ AY2 = \frac{1}{24} / AX2 * (AV2 * (AP2 - AQ2)^4 - AS2 * (AT2 - AP2)^4 + AS2 * AX2^4 + 0.5 * \left( \frac{AU2 / AX2 - AV2 * AX2/6 + (AP2 - AQ2) * 2 - (AR2 / AX2 - AS2 * AX2/6 + (AT2 - AP2)^2 + (AR2 / AX2 - AS2 * AX2/6 + AX - 2)^2 \right)} {2} \]  
The integral is calculated between the arbitrary x-value in column AP and the node at the beginning of the spline used.

The formulas in AX2 and AY2 can be copied to AX3:AX100 and AY3:AY100, respectively. The final formulas that need to be entered are those in column I in the Main worksheet. These formulas are

\[ I3 = \text{IF}(F3 = '', '', 0) \]  
The integral is zero at the first arbitrary point.

\[ I4 = \text{IF}(F4 = '', '', \text{LOOKUP}(\text{Calc!AQ3, AS3:AS100, ES3:ES100}) + \text{Calc!AY3 - LOOKUP(\text{Calc!AQ52, AS3:AS100, ES3:ES100})} \]  
The area under the spline curve(s) is calculated from the area under the curve from the first arbitrary x-value in the F-column to the present value.
The programming above may appear tedious but is quite logical, if familiarity with basic Excel functionality exists. Fig. 8 shows the result of using the above spreadsheet with four data points. The figure is presented as a method by which to check the programming above. Verification that the spreadsheet was constructed correctly was performed using Octave, modified to allow the built-in SPLINE.m function to use normal cubic splines (data not shown).

4. Numerical application

The applications for using smooth curves to interpolate data or determining derivatives are numerous and appear in many fields of science. In the area of biotechnology or chemical engineering, derivatives are needed to calculate reaction rates, and in the area of chemistry, derivatives are necessary for determination of thermodynamic properties. To illustrate an example in the biotechnology area, we have plotted data from Koga et al. [11,12], who carried out a time experiment where bacteria converted glucose to gluconolactone and then to gluconic acid in a fermentation study. The various concentrations were recorded in time and are shown in Fig. 9. To illustrate the usefulness of splines we have constructed cubic splines as described above with a smoothness factor of 0.5. As expected, the splines represent the data quite well. This approach could have been used to find derivatives needed to describe the kinetics of the fermentation. Koga et al. used simple finite difference methods to carry out their analysis of data [11].

5. Conclusions and limitations

As shown in this manuscript, spreadsheet functions can be used to construct cubic splines to trend, smooth, interpolate, integrate, and derivate experimental data. The theory of splines is not new, just its implementation into a spreadsheet environment without intricate programming. Just as with other methods using cubic splines, there are limitations of their implementation. As a general guideline, the experimental data should not be too closely spaced; for example, duplicate readings of y at the same x-value must be reduced to a single value. Data too closely spaced will, in general, cause large fluctuation in the shape of the overall trend line and may cause improbable estimates of interpolated values as well as derivatives and integrals if smoothing is not performed [5]. In the above spreadsheet, the raw data should also be entered in the order of increasing x-values. Another limitation (or strength) of the smoothing method described above is that it seeks to straighten the overall trend curve to a straight line between the first and last data point [5]. Some of the other published smoothing methods used in combination with cubic splines also have limitations. For example, the capability of the extended spline fit technique to describe the data depends on which nodes to keep and which nodes to eliminate when finding the spline functions [2,3]. Regardless of their limitation, they all present solid methods for smoothing of experimental data and providing a method by which to find derivatives and integrals.

There are other mathematical software packages like MATLAB or Octave that have built-in subroutines for spline functions: provided with a set of x- and y-values, interpolation can be performed via splines and trend lines can be constructed and visually displayed. A variety of smoothing mechanisms also exist within these programs. The above spreadsheet was not developed to replace these powerful software programs—it was developed to show the capabilities of Excel and provide a simple method for evaluating experimental data. The wide-spread use of Excel makes the program easy to transport and distribute.
References

[1] Chaney W, Kincaid D. Numerical mathematics and computing. Monterey, CA: Brooks/Cole; 1980 [It should be noted that there is a typographical error on page 227 of this reference. A statement \[G(N) = G(N)/E(N)\] should be added to both the FORTRAN and BASIC programs listed on this page. The statement should be placed before the beginning of the last loop].


