Available online at www.sciencedirect.com
ScienceDirect
Advances in Engineering Software 39 (2008) 422–429
ADVANCES IN ENGINEERING SOFTWARE
www.elsevier.com/locate/advengsoft

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

* Tel.: +1 504 286 4511; fax: +1 504 286 4367.
E-mail address: tklasson@srrc.ars.usda.gov

0965-9978/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2007.03.006

Research, Champaign, IL) also have powerful curve-fitting routines and programming language.

In order to construct curves through experimental data points, cubic splines are often used. Here, each segment between two neighboring data points are connected with a third-order polynomial [1]; thus, for N data points, $N-1$ polynomials (splines) are used. Conditions are set 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, Klaus and Van Ness used an enhanced spline technique [2]. In that case, fewer than $N-1$ polynomials were used and a typical least square regression mechanism was applied to determine the "best" overall curve. Later, the method was further modified to include uncertainties in the experimental data and calculation of more suitable data points where the polynomials joined [3]. The method was complex and a FORTRAN program could be ordered from the authors. FORTRAN was also the choice of two other methods of smoothing data with spline functions, but the programs were too lengthy to be listed [4]. Another example of a method of data smoothing, also based on cubic splines, has previously been suggested in which the splines are constructed, not through the original experimental data points but between them, and the code for Q-BASIC program was provided [5].

While there are many computer software that can be (and has been) used for programming cubic spline equations, Excel was chosen in this work because of its familiarity among researchers and students. It offers data storage, data manipulation, graphical representation, and is available 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 problem-solving skills in students and expanding their capabilities [6,7]. While other more powerful software and programming languages exist (e.g., MATLAB, Octave, and Mathematica), spreadsheets continues to be successfully applied to a variety of scientific principles such as chemistry [8] and engineering [9,10]. In this work, a spreadsheet 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 [polynomials, $f(x)$] 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 polynomials, 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

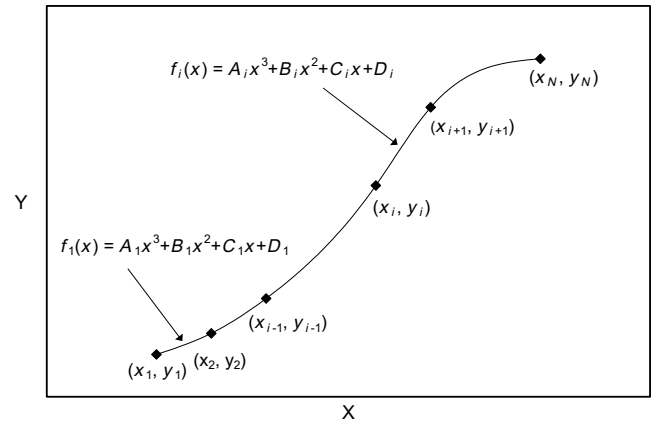

Fig. 1. Cubic spline construction.

polynomials. At the endpoints, $f_1''(x_1)$ and $f_{N-1}''(x_N)$ are set to zero [1] for normal cubic splines or they can be set to know values or to $f_1''(x_2)$ and $f_{N-1}''(x_{N-1})$ [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:

$$\begin{pmatrix} b_2 & c_2 & & & & \\ a_3 & b_3 & c_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} f_2'' \\ f_3'' \\ \vdots \\ f_i'' \\ \vdots \\ f_{N-1}'' \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ \vdots \\ r_i \\ \vdots \\ r_{N-1} \end{pmatrix}, \tag{1}$$

where

$$a_{i+1} = (x_{i+1} - x_i), \quad b_{i+1} = 2(x_{i+2} - x_i),$$
$$c_{i+1} = (x_{i+2} - x_{i+1}), \quad \text{and} \tag{2a, 2b, 2c}$$
$$r_{i+1} = 6\left(\frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right). \tag{3}$$

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]:

$$f_i(x) = \frac{f_i''(x_{i+1} - x)^3 + f_{i+1}''(x - x_i)^3}{6(x_{i+1} - x_i)} + \left(\frac{y_i}{x_{i+1} - x_i} - \frac{f_i''(x_{i+1} - x_i)}{6}\right)(x_{i+1} - x) + \left(\frac{y_{i+1}}{x_{i+1} - x_i} - \frac{f_{1+1}''(x_{i+1} - x_i)}{6}\right)(x - x_i) \tag{4}$$

$$f_i'(x) = \frac{f_{i+1}''(x-x_i)^2}{2(x_{i+1}-x_i)} - \frac{f_i''(x_{i+1}-x)^2}{2(x_{i+1}-x_i)} + \frac{y_{i+1}-y_i}{x_{i+1}-x_i} - \frac{(f_{i+1}''-f_i'')(x_{i+1}-x_i)}{6} \tag{5}$$

$$\int_\alpha^\beta f_i(x)\mathrm{d}x = \left| \begin{array}{l} (x-x_i)^2\left(\frac{f_{i+1}''(x-x_i)^2}{24(x_{i+1}-x_i)} - \frac{f_{i+1}''(x_{i+1}-x_i)}{12} + \frac{y_{i+1}}{2(x_{i+1}-x_i)}\right) \\ -(x_{i+1}-x)^2\left(\frac{f_i''(x_{i+1}-x)^2}{24(x_{i+1}-x_i)} - \frac{f_i''(x_{i+1}-x_i)}{12} + \frac{y_i}{2(x_{i+1}-x_i)}\right) \end{array} \right|_\alpha^\beta . \tag{6}$$

If should be noted that in Eqs. (1)–(6), f_i'' denotes the second derivative evaluated at the ith point's x-value (x_i) using the ith spline, remembering that f_1'' and f_N'' 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 $\alpha = x_i$ and $\beta = x_{i+1}$ or when $\alpha = x_i$ 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 $\quad x_0 = x_1 - 0.05(x_2 - x_1)$

$$y_0 = y_1 + \frac{(y_2-y_1)(x_0-x_1)}{(x_2-x_1)} \tag{7a, 7b}$$

fictitious point 2 $\quad x_{N+1} = x_N + 0.05(x_N - x_{N-1})$

$$y_{N+1} = y_N + \frac{(y_{N-1}-y_N)(x_{N+1}-x_N)}{(x_{N-1}-x_N)} \tag{8a, 8b}$$

for $i = 0$ to N, $\quad x_{mi} = \frac{x_{i+1}+x_i}{2}$, $y_{mi} = \frac{y_{i+1}+y_i}{2}$. (9a, 9b)

This creates a total of $N+1$ mid-points that can be connected with N splines. Once the splines have connected the temporary points, interpolation is used to determine the smoothed y values at the original x-values. This sequence of calculations corresponds to a smoothness factor (sf) of 1 [5]. If the smoothed values of y at the x_i's are used as new starting points for another iteration with mid-points as described above, the smoothness factor is said to be equal to 2 and so on. We may also consider a smoothness factor with a value of <1 as a method to weigh the raw with the smoothed data. By this notion, a smoothness factor of 0.25 would represent a data point with a y-value calculated from the raw y-value and the smoothed y-value (with a smoothness factor of 1). The equation below demonstrates the principle of calculating a smoothed y-value with a smoothness factor of 0.25.

$$y_{\mathrm{sf}=0.25} = 0.75y_{\mathrm{raw}} + 0.25y_{\mathrm{sf}=1} \tag{10}$$

3. Numerical solution and spreadsheet development

Most of the equations above are simple to program in Excel. The one possible complication is solving the matrix (Eq. (1)) without VisualBasic programming; however, the matrix can be reduced to

$$\begin{pmatrix} bb_2 & c_2 & & & & \\ & bb_3 & c_3 & & & \\ & & \ddots & \ddots & & \\ & & & bb_i & c_i & \\ & & & & \ddots & \ddots \\ & & & & & bb_{N-1} \end{pmatrix} \begin{pmatrix} f_2'' \\ f_3'' \\ \vdots \\ f_i'' \\ \vdots \\ f_{N-1}'' \end{pmatrix} = \begin{pmatrix} rr_2 \\ rr_3 \\ \vdots \\ rr_i \\ \vdots \\ rr_{N-1} \end{pmatrix} \tag{11}$$

as a first step, where $bb_2 = b_2$, $rr_2 = r_2$, and

for $\quad i = 3$ to $N-1$,

$$bb_i = b_i - \frac{a_i c_{i-1}}{bb_{i-1}} \quad \text{and} \quad rr_i = r_i - \frac{a_i rr_{i-1}}{bb_{i-1}}. \tag{12a, 12b}$$

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

$$\begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} f_2'' \\ f_3'' \\ \vdots \\ f_i'' \\ \vdots \\ f_{N-1}'' \end{pmatrix} = \begin{pmatrix} rrr_2 \\ rrr_3 \\ \vdots \\ rrr_i \\ \vdots \\ rrr_{N-1} \end{pmatrix}, \tag{13}$$

where

for $\quad i = N-1$ to 2,

$$rrr_i = \frac{rr_i - c_i rrr_{i+1}}{bb_i} \quad (\text{note that } rrr_N = 0). \tag{14}$$

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 rrr_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).

	A	B	C	D	E	F	G	H	I
1	Enter x and y here		Smoothness Factor =	0		Arbitrary	Calculated	Calculated	Calculated
2	x	y	Smooth y	dy/dx	∫ y dx	x	y	dy/dx	∫ y dx
3									
4									
5									
6									
7									

Main Calc

Fig. 2. Structure of Main worksheet in an Excel workbook.

	A	B	C	D	E	F	G	H	I	J	K
1	i	x	y	ys	a	b	c	r	bb	rr	rrr
2											
3											
4											
5											
6											
7											

Main Calc

Fig. 3. Structure of Calc worksheet in an Excel workbook.

B3 =Main!A3 Transfer of the first x-value (x_1).

B4 =IF(A4="",B3+0.05*(B3−B2),Main!A4) If the counter column is empty, calculate the x-value (x_{N+1}) of the "high" fictitious data point from Eq. (8a), otherwise transfer the ith x-value (x_i).

C2 =Main!B3+(Main!B4−Main!B3)*($B2−$B3)/($B4−$B3) Calculation of the y-value (y_0) of the "low" fictitious data point from Eq. (7b).

C3 =Main!B3 Transfer of the first y-value (y_1).

C4 =IF(A4="",C3+(C2−C3)*(B4−B3)/(B2−B3),Main!B4) If the counter column is empty, calculate the y-value (y_{N+1}) of the "high" fictitious data point from Eq. (8b), otherwise transfer the ith y-value (y_i).

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.

E5 =B5−B4 This calculates the first a-value (a_3) from Eq. (2a) for Eq. (1) matrix; the rest of the a_i-values are calculated in the same way.

F4 =2*(B5−B3) This calculates the first b-value (b_2) from Eq. (2b) for Eq. (1) matrix; the rest of the b_i-values are calculated in the same way.

G4 =B5−B4 This calculates the first c-value (c_2) from Eq. (2c) for Eq. (1) matrix; the rest of the c_i-values are calculated in the same way.

H4 =IF($A5="",0,6*(D5−D4)/(B5−B4)−6*(D4−D3)/(B4−B3)) If this is the last point, set r_N to zero, otherwise r_i is calculated from Eq. (3) for Eq. (1) matrix.

I4 =F4 $bb_2 = b_2$ when the matrix in Eq. (1) is reduced to Eq. (11) matrix.

I5 =F5−E5/I4*G4 bb_i is calculated from Eq. (12a) for Eq. (11) matrix.

J4 =H4 $rr_2 = r_2$ when the matrix in Eq. (1) is reduced to Eq. (11) matrix.

J5 =IF($A6="",0,H5−E5/I4*J4) If this is the last point, set rr_N to zero, otherwise rr_i is calculated from Eq. (12b) for Eq. (11) matrix.

K3 =0 $rrr_1 = f_1''(x_1)$ is set to zero.

K4 =IF($A5="",0,(J4−G4*K5)/I4) If this is the last point, $rrr_N = f_{N-1}''(x_N)$ is set equal to zero, otherwise, $rrr_i = f_i''(x_i)$ is calculated from Eq. (14) for Eq. (13) matrix.

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 work sheet. 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 start 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.

L2 =(B3+B2)/2

M2 =(C3+C2)/2

	L	M	N	O	P	Q	R	S	T	U
1	xm	ym	a	b	c	r	bb	rr	rrr	ys
2										
3										
4										
5										
6										
7										

Main Calc

Fig. 4. Further construction of the Calc worksheet.

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−P3*T4)/R3)

The formula in column U deserves 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_s) 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.

U2 =U3+(U4−U3)*($B2−$B3)/($B4−$B3) This calculation is used to set up the y-value for the next mid-point, when the smoothness factor is 2 and it corresponds to a new low fictitious point.
U3 =(T2*(L3−$B3)^3+T3*($B3−L2)^3)/6/(L3−L2)+(M2/(L3−L2)−T2*(L3−L2)/6)*(L3−$B3)+(M3/(L3−L2)−T3*(L3−L2)/6)*($B3−L2) This calculates the first y-value (y_{s1}) from Eq. (4); the rest of the y_{si}-values are calculated in the same way.

The content of L2, M2, N4, O3, P3, Q3, R4, 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:

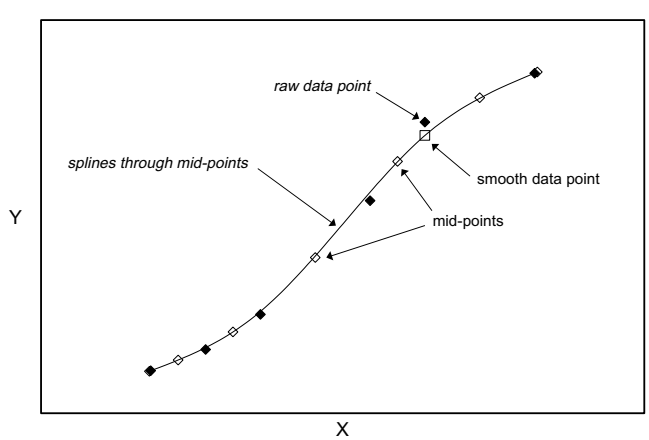

Fig. 5. Construction of splines through mid-points. There are two mid-points that are difficult to note in the figure, as they are located slightly beyond the first and last raw experimental data point.

V2 =L2 The mid-point x-values remain the same.
W2 =(U3+U2)/2 New mid-point y-values are calculated from the smoothed y-values in column U.

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.

D2 =IF(AND(A2="",A1=""),0,IF(Main!$D$1=0,C2,IF(Main!$D$1<1,Main!$D$1*U2+(1− Main!$D$1)*C2,IF(Main!$D$1=1,U2,IF(Main! $D$1=2,AE2,AO2)))))

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:

AP2 =IF(Main!F3="","",Main!F3) If there is an arbitrary x-value for which an y-value is needed, it will be copied here.
AQ2 =INDEX(B$2:B$100,MATCH(AP2, B$2:B$100,1)−IF(AP2=MAX(Main!A$3:A$100),1,0)) This function looks up the raw data x-value, less than the arbitrary x-value.
AR2 =LOOKUP(AQ2,B$2:B$100,D$2:D$100) This function looks up the corresponding y-value to the AQ2 x-value.
AS2 =LOOKUP(AQ2,B$2:B$100,K$2:K$100) This function looks up the secondary derivative at the point defined by AQ2 and AR2.
AT2 =INDEX(B$2:B$100,MATCH(AP2,B$2:B$100, 1)+IF(AP2=MAX(Main!A$3:A$100),0,1)) This

	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
1	xm	ym	a	b	c	r	bb	rr	rrr	ys	xm	ym	a	b	c	r	bb	rr	rrr	ys
2																				
3																				
4																				
5																				
6																				
7																				

Main Calc

Fig. 6. Further construction of the Calc worksheet to create smoother trendlines.

	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
1	any x	xi	yi	rrri	xi+1	yi+1	rrri+1	y for any x	(xi+1)-(xi)	step ∫ y dxdt
2										
3										
4										
5										
6										
7										

Main Calc

Fig. 7. Further construction of the Calc worksheet for interpolation, derivatives, and integration.

function looks up the raw data x-value, greater than the arbitrary x-value.

AU2 =LOOKUP(AT2,B$2:B$100,D$2:D$100) This function looks up the corresponding y-value to the AT2 x-value.

AV2 =LOOKUP(AT2,B$2:B$100,K$2:K$100) This function looks up the secondary derivative at the point defined by AT2 and AU2.

AW2 =(AS2*(AT2−AP2)^3+AV2*(AP2−AQ2)^3)/6/(AT2−AQ2)+(AR2/(AT2−AQ2)−AS2*(AT2−AQ2)/6)*(AT2−AP2)+(AU2/(AT2−AQ2)−AV2*(AT2−AQ2)/6)*(AP2−AQ2) 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 =IF(F3=”’’,”’’,Calc!AW2) in the Main worksheet.

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.

D3 =IF(A3=”’’,”’’,IF(A4=”’’,(C3−C2)/(A3−A2)+Calc!K2*(A3−A2)/6,−Calc!K3*(A4−A3)/2+(C4−C3)/(A4−A3)−(Calc!K4−Calc!K3)/6*(A4−A3))) If there is an x-value in column A, the derivative is calculated using Eq. (5).

E3 =IF(A3=”’’,”’’,0) At the first point there is no area under the curve.

E4 =IF(A4=”’’,”’’,0.5*(A4−A3)*(C3+C4)−1/24*(A4−A3)^3*(Calc!K3+Calc!K4)+E3) The integral is calculated between the x-value in column A and the previous x-value according to Eq. (6) and this area segment is added to the previous area calculated in the cell above.

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 =1/24/AX2*(AV2*(AP2−AQ2)^4−AS2*(AT2−AP2)^4+AS2*AX2^4)+0.5*((AU2/AX2−AV2 * AX2/6)*(AP2−AQ2)^2−(AR2/AX2−AS2*AX2/6)*(AT2−AP2)^2+(AR2/AX2−AS2*AX2/6)*AX-2^2) 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 =IF(F3=”’’,”’’,0) The integral is zero at the first arbitrary point.

I4 =IF(F4=”’’,”’’,LOOKUP(Calc!AQ3, A$3:A$100, E$3:E$100)+Calc!AY3−LOOKUP(Calc!AQ$2, A$3:A$100,E$3:E$100)−Calc!AY$2) 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.

	A	B	C	D	E	F	G	H	I
1	Enter x and y here		Smoothness Factor =	0.5		Arbitrary	Calculated	Calculated	Calculated
2	x	y	Smooth y	dy/dx	∫ y dx	x	y	dy/dx	∫ y dx
3	3	2.5	2.497231786	-1.1528488	0	4	1.486379679	-0.72685867	0
4	4.5	1	1.247197473	-0.194371	2.62860735	5	1.289319518	0.3219816	1.2956727
5	7	2.5	2.200080793	-0.0652611	6.87046048	6	1.841510844	0.61889084	2.8363454
6	9	0.5	0.503315019	-1.2399438	9.96541716	8	1.645368557	-0.9462731	6.9102963
7									

Main / Calc

Fig. 8. Example of calculations using the Excel workbook with a smoothness factor of 0.5.

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].

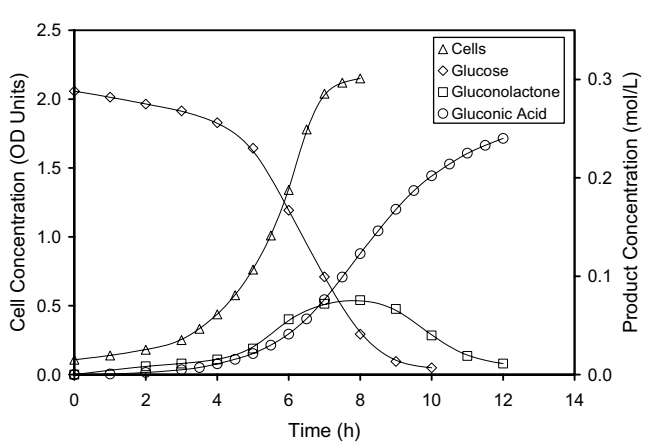

Fig. 9. Fermentation profiles to illustrate smooth cubic splines with a smoothness factor of 0.5. The curves on the graph were generated by entering large number of arbitrary x-values into the Excel work book (in the F-column of the Main worksheet).

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].

[2] Klaus RL, Van Ness HC. An extension of the spline fit technique and application to thermodynamic data. AIChE J 1967;13(6):1132–6.

[3] Nunhez JR, Mori M, d'Avila SG. Fitting thermodynamic data using the modified spline technique. Comput Chem Eng 1993;17(11): 1091–9.

[4] Lyche T, Schumaker LL, Sepehrnoori K. Fortran subroutines for computing smoothing and interpolating natural splines. Adv Eng Software 1983;5(1):2–5.

[5] Klasson KT. Experimental data analysis. An algorithm for determining rates and smoothing data. Appl Biochem Biotechnol 1997;63/65:339–48.

[6] Abramovich S, Nabors W. Spreadsheets as generators of new meanings in middle school algebra. Comp Schools 1997;13(1–2):13–25.

[7] Molyneux-Hodgson S, Rojano T, Sutherland R, Ursini S. Mathematical modeling: the interaction of culture and practice. Educ Stud Math 1999;39(1–3):167–83.

[8] Billo JE. Excel for chemists. Hoboken, NJ: John Wiley & Sons; 2004.

[9] Billo JE. Excel for scientists and engineers. Hoboken, NJ: John Wiley & Sons; 2007.

[10] Filby G. Spreadsheets in science and engineering. New York, NY: Springer-Verlag; 1998.

[11] Koga S, Burg CR, Humphrey AE. Computer simulation of fermentation systems. Appl Microbiol 1967;15(4):683–9.

[12] Tanner RD. An enzyme kinetic model for describing fermentation processes. Biotechnol Bioeng 1970;12:831–43.