Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
http://www.simfit.org.uk

Given a set of observations there are several ways to fit a piecewise cubic spline curve in order to generate a best-fit smoothed approximation than can then be used to visualize trends in the data, to perform calculations such as estimating derivatives, or to use as a calibration curve. It is important to appreciate that the best-fit spline is very dependent on the technique used so this must be explained.

Definition of a piecewise spline curve

A spline curve consist of contiguous sections separated at junction points called called knots, where a distinct cubic polynomial is defined for each section. If the data points x, y are in nondecreasing order of x with only one observation y_i at each value x_i then such a piecewise cubic spline curve has $(k + 4)/2$ knots in all, with four of these at the first x value, four at the last x value, and $(k - 12)/2$ at interior x values. For each interval defined by the extreme x values and the interior knots there will be a cubic polynomial $p(x)$ given by

$$p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3$$

so that the model equation is actually a set of cubic polynomials with one for each interval. This model is fitted to minimize some objective function using the model evaluated for x_i as the value of the corresponding cubic polynomial defined for the interval containing x_i. Cubic polynomials that are adjacent must have the same function value and derivative where they meet at a knot, but there are numerous ways to define or calculate knot positions, and to define the objective function to be minimized.

Note that, as spline fitting procedures require only one observation y_i at each distinct x_i value, replicate observations in data sets supplied to SIMFIT are transformed internally to data sets with means replacing replicates for fitting, and weights calculated from replicates if no weights were supplied.

Program **spline** which can be opened using the [A/Z] option from the SIMFIT main menu offers two options.

- **Input a new data file for fitting**

  This can then be fitted to generate a set of spline knots and calculate spline coefficients which then become the defaults for all subsequent procedures.

- **Input a previously saved spline knots file**

  This must contain a set of knots and coefficients to define a default best-fit spline curves that can be used directly for all subsequent procedures.

In addition there are the following main fitting techniques available.

1. Knots defined by users
2. Knots calculated automatically given a smoothing factor F
3. Knots between data points defined by a smoothing parameter ρ
4. Knots between data points with ρ defined by cross-validation

There are no hard and fast rules but the SIMFIT program **calcurve** uses method 1 which is the easiest to understand and gives users complete control over defining a calibration curve, program **compare** uses method 2 to define splines that can be used to compare profiles for similarity and differences, program **csafit** uses method 1 to approximate flow cytometry profiles, while program **spline** is for those who wish to investigate several methods for estimating derivatives, arc lengths, curvature, displaying trends in data, or for calibration analysis.

The SIMFIT test file that will be used to demonstrate spline fitting procedures is `compare.tf1` containing the following data.

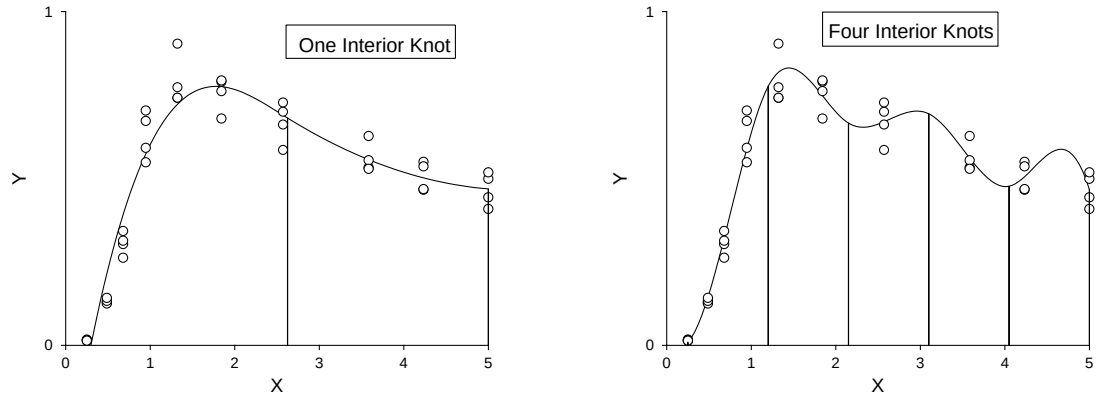
x	y	se
0.25000	0.017267	1
0.25000	0.015585	1
0.25000	0.014268	1
0.25000	0.014136	1
0.48647	0.12861	1
0.48647	0.12536	1
0.48647	0.13339	1
0.48647	0.14230	1
0.67860	0.26261	1
0.67860	0.34277	1
0.67860	0.30364	1
0.67860	0.31373	1
0.94662	0.67252	1
0.94662	0.70382	1
0.94662	0.59192	1
0.94662	0.54850	1
1.3205	0.90417	1
1.3205	0.74158	1
1.3205	0.77353	1
1.3205	0.74208	1
1.8420	0.79030	1
1.8420	0.79384	1
1.8420	0.67971	1
1.8420	0.76176	1
2.5695	0.58575	1
2.5695	0.66178	1
2.5695	0.70023	1
2.5695	0.72772	1
3.5844	0.53286	1
3.5844	0.62744	1
3.5844	0.55484	1
3.5844	0.52923	1
4.2334	0.55003	1
4.2334	0.46641	1
4.2334	0.46840	1
4.2334	0.53647	1
5.0000	0.49920	1
5.0000	0.51847	1
5.0000	0.44355	1
5.0000	0.40895	1

The columns contain data in the following format.

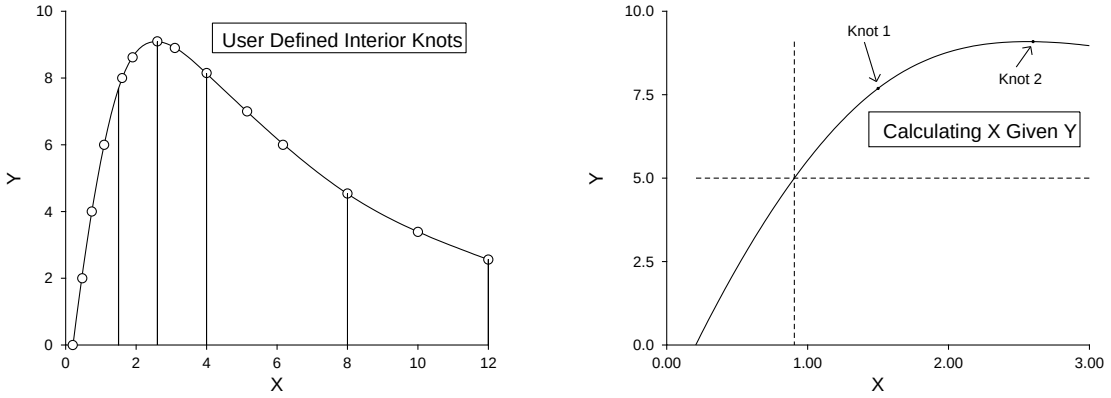
1. **Column 1**: the variable x which must be in non-decreasing order.
2. **Column 2**: the response y presumed to be dependent on x.
3. **Column 3**: the value of 1 indicates that the replicates will be used to calculate the sample standard deviations at each x-replicate value to be used for weighting.
   This column can be omitted or set to a positive value se if it is wished to supply weighting factors w directly which would then be used as $w = 1/se^2$.

Knots defined by user

Here the user must specify the number of interior knots and their spacing in such a way that genuine dips, spikes or asymptotes in the data can be modeled by clustering knots appropriately. Four knots are added automatically to correspond to the smallest x value, and four more are also added to equal the largest x value. If the data are monotonic and have no such spike features, then equal spacing can be resorted to, so users only need to specify the actual number of interior knots. The programs **calcurve** and **csafit** offer users both of these techniques, as knot values can be provided after the termination of the data values in the data file, while program **spline** provides the best interface for interactive spline fitting. Fixed knot splines have the advantage that the effect of the number of knots on the best fit curve is fully intuitive; too few knots lead to under-fit, while too many knots cause over-fit. The next figure illustrates the effect of changing the number



of equally spaced knots when fitting the data in `compare.tf1` by this technique. The vertical bars at the knot positions were generated by replacing the default symbols (dots) by narrow (size 0.05) solid bar-chart type bars. It is clear that the the fit with one interior knot is quite sufficient to account for the shape of the data, while using four gives a better fit at the expense of excessive undulation. To overcome this limitation of fixed knots SimFIT provides the facility to provide knots that can be placed in specified patterns and, to illustrate this, the next figures display several aspects of the fit to `e02baf.tf1`.



The left hand figure shows the result when spline knots were input from the spline file `e02baf.tf2`, while the right hand figure shows how program **spline** can be used to predict X given values of Y. Users simply specify a range of X within the range set by the data, and a value of Y, whereupon the intersection of the dashed horizontal line at the the specified value of Y is calculated numerically, and projected down to the X value predicted by the vertical dashed line. Note that, after fitting `e02baf.tf1` using knots defined in `e02baf.tf2`, the best fit spline curve was saved to the file `spline.tf1` which can then always be input again into program **spline** to use as a deterministic equation between the limits set by the data in `e02baf.tf1`.

Automatically calculated knots

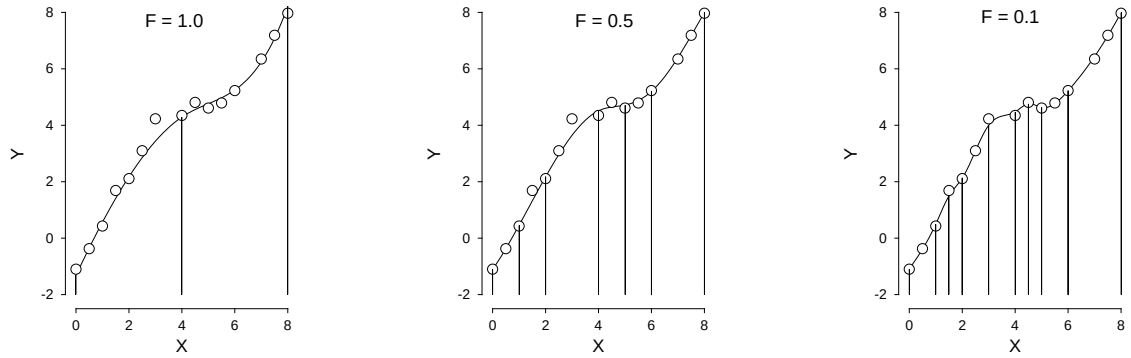
Here the knots are generated automatically and the spline is calculated to minimize

$$\eta = \sum_{i=5}^{m-5} \delta_i^2,$$

where δ_i is the discontinuity jump in the third derivative of the spline at the interior knot i, subject to the constraint

$$0 \leq WSSQ_N \leq F$$

where F is user-specified. If F is too large there will be under-fit and best fit curve will be unsatisfactory, but if F is too small there will be over-fit. For example, setting $F = 0$ will lead to an interpolating spline passing through every point, while choosing a large F value will produce a best-fit cubic polynomial with $\eta = 0$ and no internal knots. In weighted least squares fitting $WSSQ$ will often be approximately a chi-square variable with degrees of freedom equal to the number of experimental points minus the number of parameters fitted, so choosing a value for $F \approx N$ will often be a good place to start. The programs **compare** and **spline** provide extensive options for fitting splines of this type. The next figure, for example, illustrates the effect of fitting `e02bef.tf1` using smoothing factors of 1.0, 0.5, and 0.1.



In between knots: ρ input

Here there is one knot between each distinct x value and the spline $f(x)$ is calculated as that which minimizes

$$WSSQ_N + \rho \int_{-\infty}^{\infty} (f''(x))^2 \, dx.$$

As with the automatically generated knots, a large value of the smoothing parameter ρ gives under-fit while $\rho = 0$ generates an interpolating spline, so assigning ρ controls the overall fit and smoothness. As splines are linear in parameters then a matrix H can be found such that

$$\hat{y} = H\bar{y}$$

and the degrees of freedom ν can be defined in terms of the leverages h_{ii} in the usual way as

$$\begin{aligned} \nu &= \text{Trace}(I - H) \\ &= \sum_{i=1}^{N} (1 - h_{ii}). \end{aligned}$$

This leads to two ways to specify the spline coefficients which depend on ρ being fixed by the user or estimated in some way.

The spline can be fixed by specifying the value of ρ. To use this option, the value of ρ is input interactively, and the resulting fit inspected graphically until it is acceptable. This way users have complete control over the amount of smoothing required.

In between knots: ρ by generalized cross validation

Alternatively, ρ can be estimated by minimizing the generalized cross validation GCV, where

$$GCV = N\left(\frac{\sum_{i=1}^{N} r_i^2}{(\sum_{i=1}^{N}(1-h_{ii}))^2}\right).$$

As this leads to a unique estimate for ρ, users have no control if the spline leads to either over-smoothing or over-fitting.

Here are the results for fitting `compare.tf1` which has 10 distinct data points using cross validation, and plotted with 95% confidence range error bars calculated from replicates, and with large dots to indicate the knot positions.

ρ	1.907E-03
DOF	3.489E-01
$WSSQ$	1.249E-04