# Interpolation of Tabular Data with Cubic Splines

A spline is a thin wooden or plastic instrument used by draftsmen to draw smooth curves through sets of x and y data points called "knots". In the early years of aircraft design, "loftsmen" working in spacious lofts would use long splines to design the wing, fuselage, and other curved surfaces of an airplane.

Mathematically, a cubic spline is a third-order curve applied to subsets of user-defined pairs of $x$ and $y$ data points or knots. A cubic spline has minimum oscillatory behavior which results in smooth transitions between data points. This property makes the curve visually pleasing.

The general form of a third-order or cubic polynomial is given by:

$$f(x) = ax^3 + bx^2 + cx + d$$

where $a$, $b$, $c$ and $d$ are constant coefficients. These coefficients are determined from several equations that reflect the properties of the cubic spline. These conditions involve such things as function and derivative values. For example, the function or y values must be equal at the interior knots, and the first and last functions must pass through the endpoints. For smoothness the first derivatives at the interior knots must also be equal. Constraints on the endpoints also determine the type of cubic spline. The spline might be natural, clamped or even cyclic.

This report describes a ***Numerit*** application called `demosfit` which demonstrates how to interpolate tabular data of the form $y = f(x)$ using cubic splines. The following is an outline of the major steps required to process a simple ASCII data file, create the cubic spline coefficient arrays and interpolate the data at user-defined $x$ data values. This program reads and fits a data file called `solar.dat` which is a typical solar activity data file. The x data of this file consists of the calendar date and the y data of this file is the solar flux.

The cubic spline process involves three individual functions. The ***Numerit*** function which performs each operation is shown in courier font. The steps are as follows:

(1) read the ASCII data file
(2) find the total number of x and y data points in this file
(3) sample and load data arrays for the spline operations (`spdata`)
(4) generate the spline coefficient arrays (`spcoef`)
(5) perform interpolation for one or more user-defined x data values (`spval`)

The errors of the curve fit can be determined by examining the difference between the original $y$ data and the `yfit` array calculated by the `spval` function.

An excellent reference book about splines is *Handbook on Splines for the User* by E. V. Shikin and A. I. Plis, published by the CRC Press. This book includes a companion floppy disk with both Fortran and C source code.

The following is a source code listing of the main driver for this program. It illustrates the program logic and procedures for interacting with each of the cubic spline functions.

```
` name of data file

fname = "solar.dat"

` read solar activity data file

readsol(fname, 1, sdate, sf10, sap)

` number of data points in each array

ndata = length(sdate)

` load data arrays

for i = 1 to ndata
    x[i] = sdate[i]
    y[i] = sf10[i]

` sample and load data arrays for spline operations

nsp = 5

spdata(x, y, ndata, nsp, xk, yk, npts)

` generate spline coefficient arrays

spcoef(xk, yk, npts, c1, c2, c3)

` evaluate spline fit at every third data point

nspj = 3

i = 0

for j = 1 to ndata
    if (j mod nspj <> 0)
       ` null
    else
       i = i + 1
       xval = x[j]
       spval(xval, xk, yk, npts, c1, c2, c3, sx)
       xfit[i] = xval
       yfit[i] = sx
```

The last part of this driver program evaluates the fit at every third data point so that it can be plotted and compared to the original data.

The following is a plot of the original data (solid line) and the cubic spline interpolation of the selected data points (dots). The "fit" is quite good for this data file.

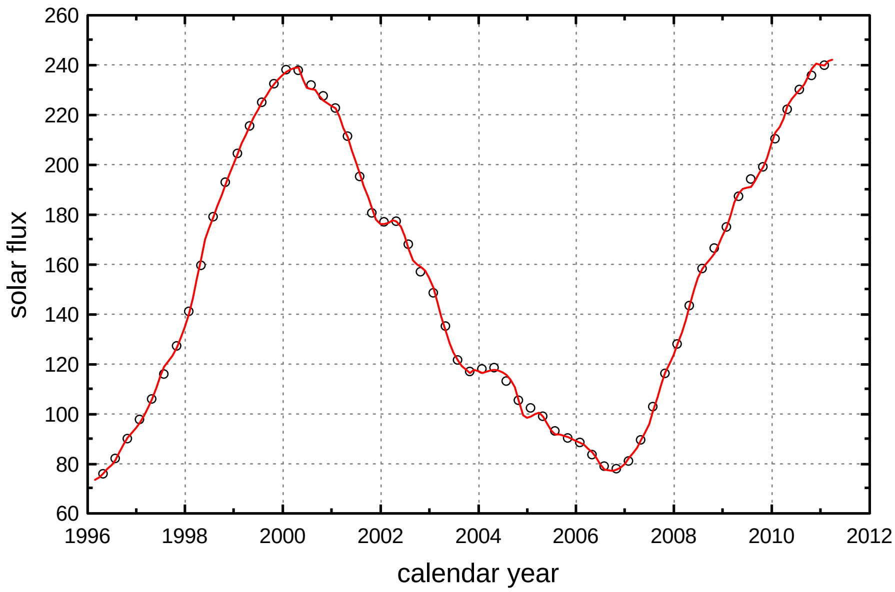


Figure 1. Cubic Spline Interpolation of Tabular Data