Mathematical Modelling
Lecture 5 – Interpolation

Phil Hasnip
phil.hasnip@york.ac.uk

Overview of Course

- Model construction $\longrightarrow$ dimensional analysis
- Experimental input $\longrightarrow$ fitting
- Finding a 'best' answer $\longrightarrow$ optimisation
- Tools for constructing and manipulating models $\longrightarrow$ networks, differential equations, integration
- Tools for constructing and simulating models $\longrightarrow$ randomness
- Real world difficulties $\longrightarrow$ chaos and fractals

A First Course in Mathematical Modeling by Giordano, Weir & Fox, pub. Brooks/Cole. Today we're in chapter 4.

Aim

The difference between these two aims is one of emphasis:

- Model fitting: we expect some scatter in the experimental data, we want the best model of a given form – ‘theory driven’
- Model interpolation: the existing data is good, model is less important – ‘data driven’

Today we’ll be focussing on the second aim: interpolation.

Interpolation

Basic problem is that we have some good output data for various experimental inputs, and we want a good estimate of the output for some similar inputs.

One-term models

If we have a simple model with one term, then we can transform the data to 'linearise' it.

E.g. $y = Ae^{bx} \Rightarrow \ln y = \ln A + bx$

Plot $\ln(y)$ against x, and end up with a straight(-ish) line – simple to interpolate.

Example

Another example,

$$\begin{aligned} y &= Ax^2 \\ \Rightarrow \sqrt{y} &= \sqrt{A}x \end{aligned}$$

Plot $\sqrt{y}$ against x, and end up with a straight(-ish) line – simple to interpolate.

One-term models

What do these transformations do? Taking $y = x$ as our reference:

- $y = x^2$ curves up, above and away from the line. The transformation $\sqrt{y} = x$ bends the curve back onto the straight line.
- $y = \sqrt{x}$ curves down, below and away from the line. The transformation $y^2 = x$ bends the curve back onto the straight line.

There is a ladder of such transformations that rank them in order of how much 'bend' they undo.

Ladder of transformations

- e^y
- y^n $(n > 2)$
- y^2
- y no change
- $\sqrt{y}$
- $\sqrt[n]{y}$ $(n > 2)$
- $\ln y$
- $-\frac{1}{\sqrt{y}}$
- $-\frac{1}{y}$

Transformations

These transformations can work well when there is a single dominant term. Their simple form means easy analysis.

Unfortunately their simple form means they are often unable to capture key features $\longrightarrow$ need a multi-term model.

Multi-term models

Polynomials are an obvious next-step.

We know (N+1) data points can be fitted exactly with (at most) an Nth-order polynormial.

Easiest expression is the Lagrangian form.

Lagrangian form polynomials

Suppose our data points are labelled $(x_0, y_0), \ldots, (x_N, y_N)$. Then an Nth order polynomial of the form

$$P_N(x) = y_0 L_0(x) + y_1 L_1(x) + \ldots + y_N L_N(x)$$

is guaranteed to fit all the data, where

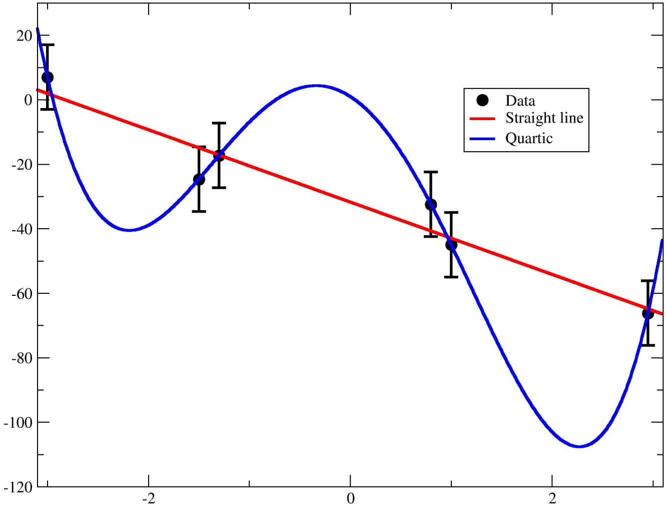
$$L_k(x) = \frac{(x - x_0) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_N)}{(x_k - x_0) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - x_N)}$$

so $P_N(x_k) = y_k$ and $\chi^2 = 0$.

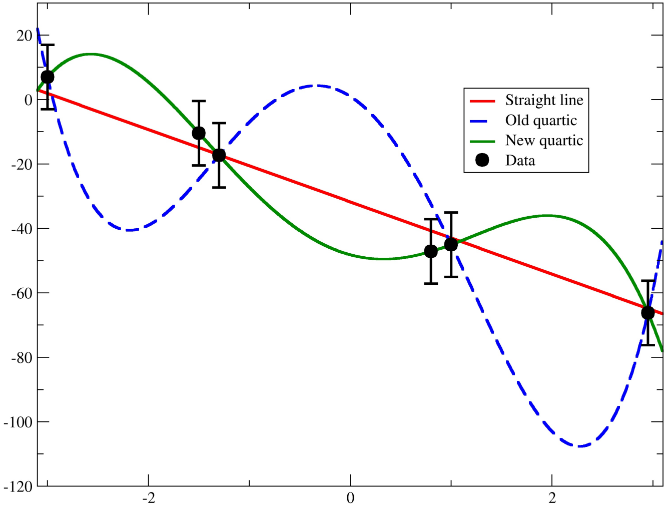
Polynomials

Easy to integrate and differentiate, but they do have problems...

Oscillations



Sensitivity



Model fitting

How can we eliminate these oscillations and sensitivity of coefficients?

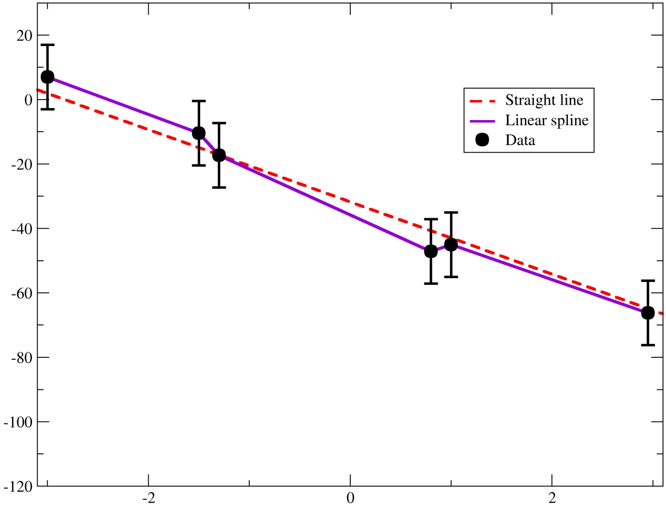
- Smooth data points – use lower order polynomial
- Calculate χ^2 and use model fitting

Splines

There is an alternative to all this polynomial fitting $\longrightarrow$ splines.

We represent the behaviour separately in between each pair of data points, and fit it piecewise.

Linear splines



Linear splines

Linear splines are effectively what we're using when we do linear interpolation. Between points (x_k, y_k) and (x_{k+1}, y_{k+1}) we represent the data as a linear spline $S_k(x)$:

$$S_k(x) = a_k + b_k x$$

In fact:

$$S_k(x) = y_k + \left(\frac{y_{k+1} - y_k}{x_{k+1} - x_k}\right)(x - x_k)$$

Cubic splines

Linear splines are easy to use, but they give a very jagged fit. A better choice is cubic splines.

This is just the same idea as before, but because cubics have two extra parameters we can add two extra conditions:

- First derivative of S_{k+1} matches S_k at $x = x_{k+1}$ (same slope)
- Second derivative of S_{k+1} matches S_k at $x = x_{k+1}$ (same curvature)

Boundary conditions

At the end points we now have some choice, since there are no adjacent splines to match – we have to choose some boundary conditions. Either:

- Natural spline assume curvature is zero at end points. This is the most common choice.
- Clamped spline assume some given slope at end points.

Summary

When we have good experimental data but little idea of a suitable model...

- Ladder of Transformations – use to guess a simple one-term model
- Polynomials – multi-term, but problems with:
  - Oscillations
  - Sensitivity

  Often better to use a low-order polynomial and fit.
- Splines
  - Linear – linear interpolation between data points
  - Cubic – smoother fit; natural or clamped? Need to think about boundary conditions