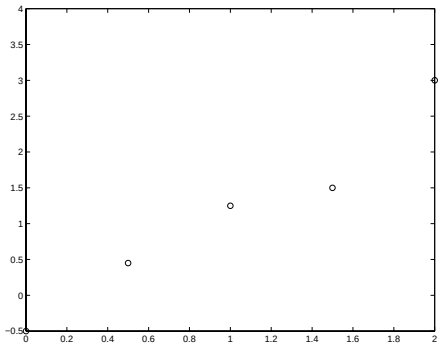
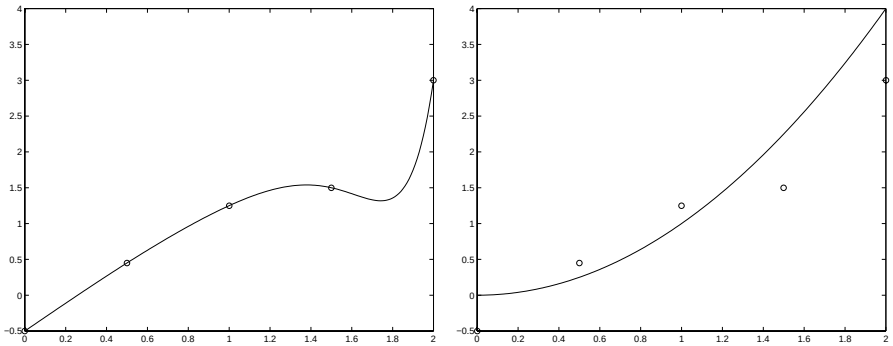
Interpolation and Curve Fitting

In working with application problems, it is not unusual to lack an analytic expression for the function you're working on. In fact, it's probabably *more* common to have your result be a large set of data values, with no expression in sight! However, this make certain things difficult, such as evaluating your 'function' at a point other than a data point, or differentiating the function. Because of this, we would like to take that data and find an analytic function that best describes it. Before we start discussing specific methods, consider the following data points on a graph:



Which of the plots below better 'fits' the data?



The answer to that is 'it depends'. The first graph hits each data point exactly, but it's got a strange dip at the end; if we were trying to use this function to estimate a derviative, we might get a negative value for the slope when the overall trend is increasing. The second graph misses all the data points, but it does pass a nice curve through them; it's following the general shape of the data.

What we're seeing here are two competing methods that we're going to discuss – one is **interpolation** (specifically, polynomial interpolation), where we construct a polynomial which hits each of the data points exactly. The catch is that with large numbers of data points, we need to use higher degree polynomials, and high degree polynomials suffer from a phenomenon with a highly technical mathematical term: 'polynomial wiggle'. nth degree polynomials have up to $n-1$ critical values, meaning potentially $n-1$ max's and min's – the 'hills and valleys' look. While the interpolating polynomial hits the data points exactly, its behavior between data points, and the derivative values everywhere, are suspect. The other is a form of **curve fitting** where our goal is to set the form of the curve in advance (such as a straight line, a parabola, or an exponential function), and produce the curve that 'best fits' the data in the sense that the differences between the data values and the function values are minimized.