Piecewise Polynomial Interpolation

We often know or record data at a set of points, and it is required that we estimate the value that the data would have taken at other points. This may be expressed as knowing the value of a function $f(x)$ at a set of n points in a range $[a, b]$ with $a = x_1 < x_2 < \cdots < x_n = b$ and we are required to *approximate*[1] the function or determine an estimate of $f(x)$ at any value of $\in [a, b]$. The determination of the function that passed through all the points is called interpolation and the most popular form of function is the polynomial[2]. With polynomial interpolation[3], a set of n points $(x_i, f(x_i))$ for *i=1,2,…n* can be interpolated by a polynomial of degree *n-1*:

$$f(x) \approx p_{n-1}(x) = a_{n-1}\,x^{n-1} + \; a_{n-2}\,x^{n-2} + \; \ldots \; a_1\,x + \; a_0\,.$$

The functions $f(x)$ and $p_{n-1}(x)$ match at the interpolation points; $p_{n-1}(x_i) = f(x_i)$ for $= 1,2, \ldots n$.

However, unless the interpolation points are carefully selected then the accuracy of the interpolant does not necessarily improve as the number of interpolation points increases[3]. One method for overcoming this is to divide the domain of $f(x)$ into subdomains and to apply polynomial interpolation with a low degree polynomial on each individual subdomain. This technique is termed *piecewise polynomial interpolation* and this kind of representation of functions is embedded into methods such as the finite element method[4] and the boundary element method[5].

In this document we will consider some elementary forms of piecewise polynomial approximation – piecewise constant, piecewise linear and piecewise quadratic. For each method we will apply it to the following data.

x_i	0	1	2	3	4	5	6	7	8
$f(x_i)$	0	2	1	4	5	4	5	4	7

Piecewise Constant Interpolation

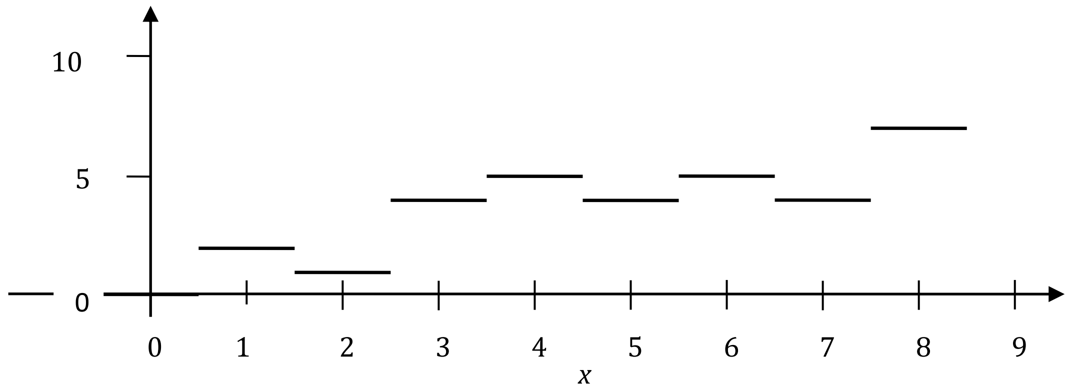
The simplest form of piecewise polynomial interpolation is piecewise constant interpolation. In this form of approximation the function $f(x)$ is approximated by a constant within each subdomain. That is

$$f(x) \approx p_0(x) = a_0\,.$$

in each subdomain.

[1] Approximation
[2] Polynomials
[3] Polynomial Interpolation
[4] Finite Element Method
[5] Boundary Element Method

With the data in the table this means that in the domain is divided into 9 subdomains, one for each point. Since the points are equally-spaced and one unit apart then the length of each subdomain is also unity. For example for the first point (0,0) the subdomain surrounding the point $x = 0$ could be to the left that is (-1,0], to the right [0,1) or with the data point in the middle, most typically a subdomain of (-0.5,0.5], as shown in the following graph of the approximating piecewise constant polynomial.



Piecewise constant approximation of the data in the table

Piecewise Linear Interpolation

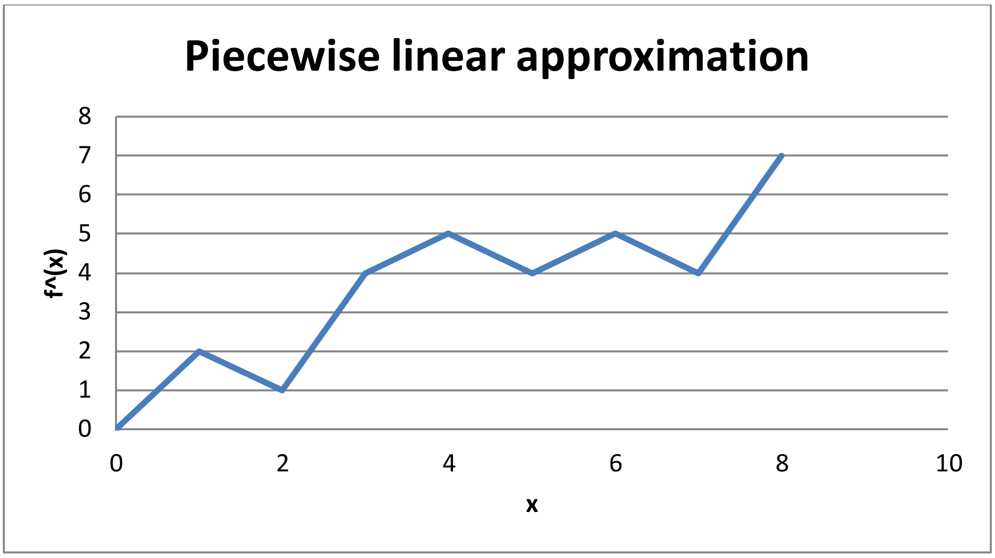
Another simple form of piecewise polynomial interpolation is piecewise linear interpolation. In this form of approximation the function $f(x)$ is approximated by a linear or straight line function[6] within each subdomain. That is

$$f(x) \approx p_1(x) = a_1 x + a_0$$

in each subdomain.

With the data in the table this means that in the domain is divided into 8 subdomains, with the approximation formed by linking the points at either side of the subdomain. Since the points are equally-spaced and one unit apart then the length of each subdomain is also unity.

[6] Equation of a Straight Line: Gradient and Intercept



Piecewise Quadratic Interpolation

The next step is to approximate the function $f(x)$ by a piecewise polynomial of degree 2 or a piecewise quadratic interpolation. In this form of approximation the function $f(x)$ is approximated by a linear or quadratic function[7] of the form

$$f(x) \approx p_2(x) = a_2 x^2 + a_1 x + a_0 .$$

within each subdomain. Since three ponts are required to define a quadratic then the domain of this problem [0,8] is divided into four subdomains [0,2], [2,4], [4,6], [6,8]. The resulting piecewise quadratic approximation is as follows.