MC3010: Differential Equations & Numerical Methods

Lecturers:
Dr. Padmanathan Kathirgamanathan & Mrs. Saranja Kiriparan

Department of Interdisciplinary Studies, Faculty of Engineering, University of Jaffna, Sri Lanka

NUMERICAL COMPUTATION

Lecture – 03: Interpolation

Polynomial

A polynomial in x is a function of the form

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots a_n x^n \qquad (a_n \neq 0, \quad n \text{ a non-negative integer})$$

where $a_0, a_1, a_2, \ldots,$ an are constants. We say that this polynomial p has degree equal to n. (The degree of a polynomial is the highest power to which the argument, here it is x, is raised.) Such functions are relatively simple to deal with, for example they are easy to differentiate and integrate.

Taylor polynomial approximations

Taylor series are a useful way of approximating functions by polynomials. The Taylor series expansion of a function $f(x)$ about $x = a$ may be stated

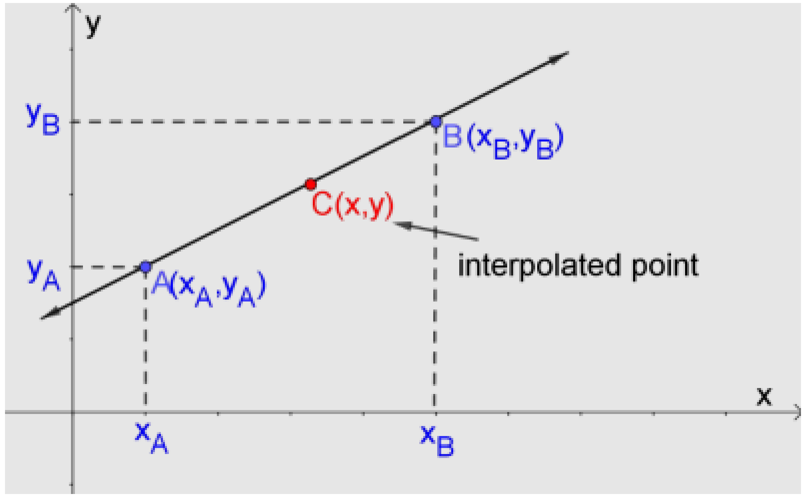
$$f(x) = f(a) + (x-a)f'(a) + \tfrac{1}{2}(x-a)^2 f''(a) + \tfrac{1}{3!}(x-a)^3 f'''(a) + \ldots.$$

The main drawback of this Taylor polynomial approximations is the **higher order derivatives must be known** (and **often they are either not available, or they are hard to compute**)

Suppose, we have a list of point values rather than knowing an expression for the function. Sometimes it is good enough to find a polynomial that passes near these points (like putting a straight line through experimental data). Such a polynomial is an approximating polynomial. We deal with the case where we want a polynomial to pass exactly through the given data, that is, an interpolating polynomial.

Numerical interpolation

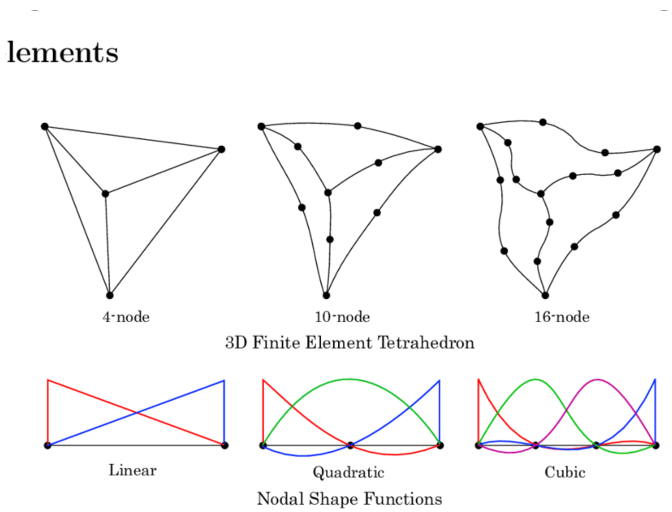
- Numerical interpolation refers to the process of estimating a value of a function between a given data points. In other words, it is the process of finding an approximate value of a function at a point within a range of values where the function is known.
- This is often done when **the exact value of the function is not available or when only a limited number of data points are available**.

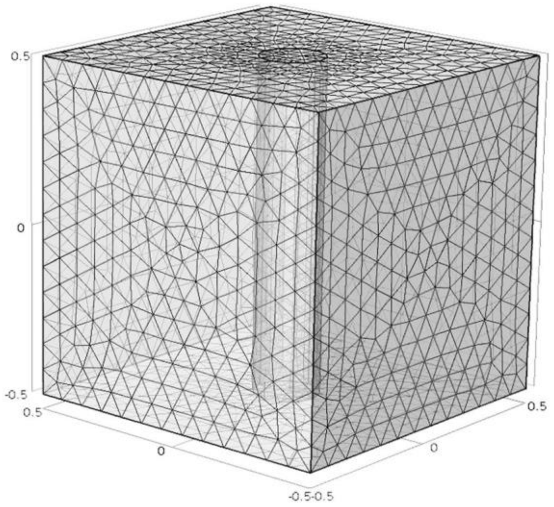


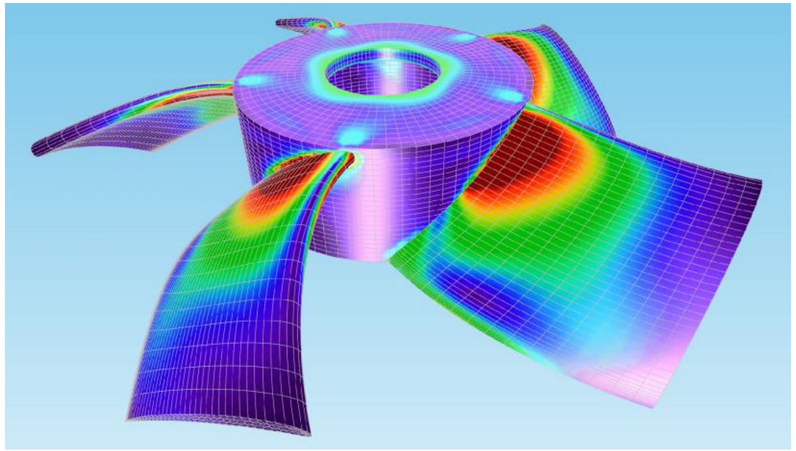
The formula for linear interpolation is:

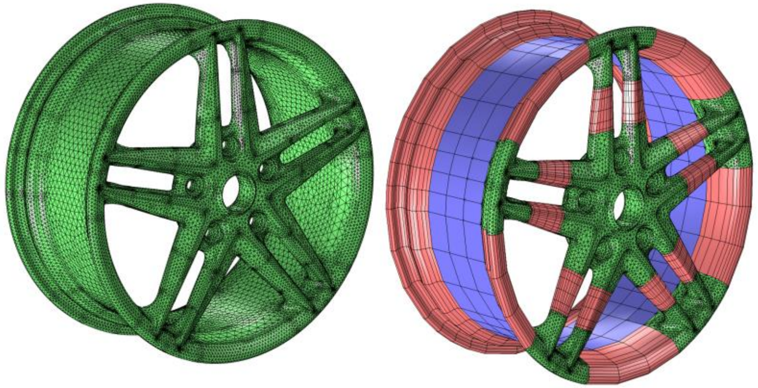
$$y = y_A + \frac{(y_B - y_A)(x - x_A)}{(x_B - x_A)}$$

Applications of Interpolation

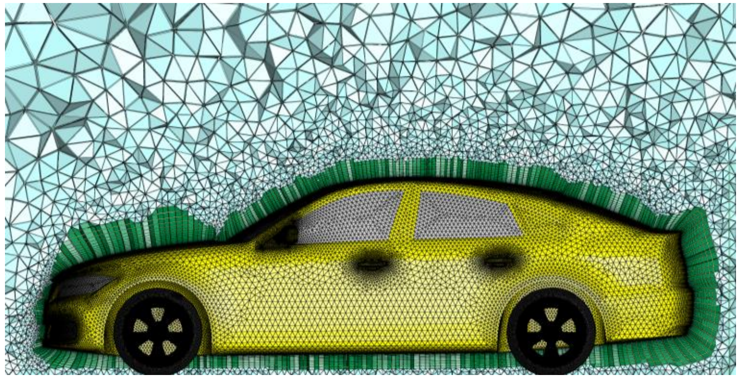


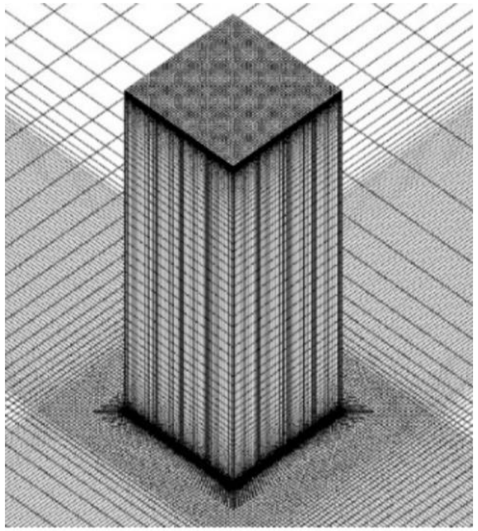






In FEA we discretize the solution region into finite elements. To conduct the analysis we assume a displacement model to approximately indicate the variations of the displacement within the element. The polynomial chosen to interpolate the field variables over the element are called shape functions.





Applications of Interpolation

A census of the population of the United States is taken every 10 years. The following table lists the population, in thousands of people, from 1950 to 2000.

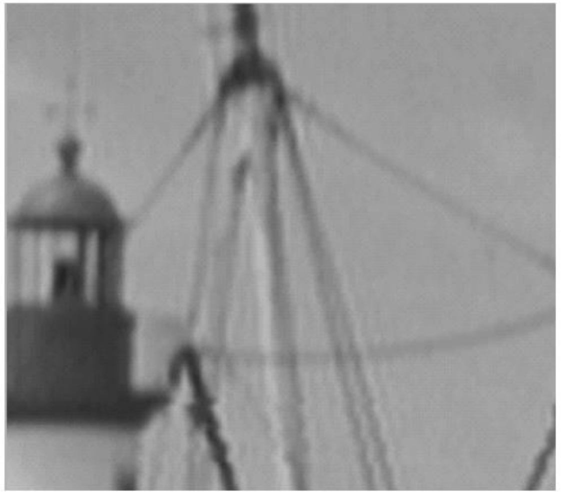
Year	1950	1960	1970	1980	1990	2000
Population (in thousands)	151,326	179,323	203,302	226,542	249,633	281,422

We can predict the population in 1975 or even in 2020 by using some kind of interpolating polynomial

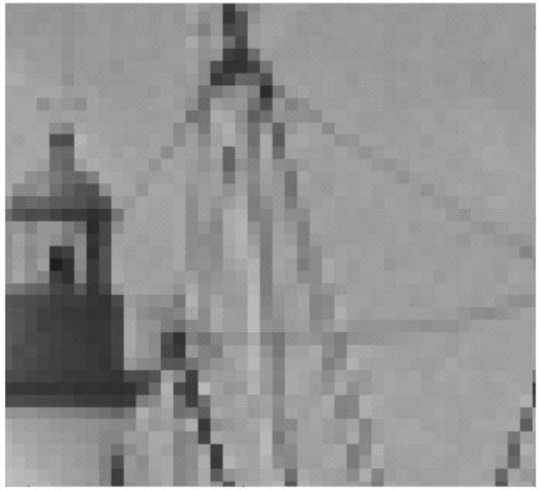
Applications of Interpolation

17. Original boat image (512×512).

Figure 18. Enlarged image by bicubic interpolation.



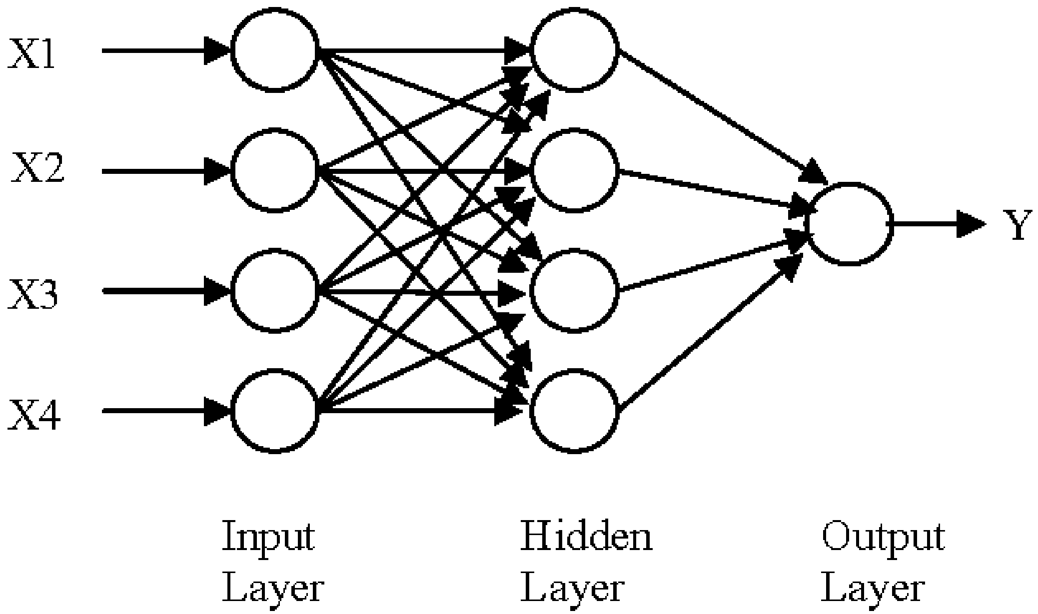
igure 19. Enlarged image by bilinear interpolation.



ure 20. Enlarged image by nearest neighbour.

Applications of Interpolation

The Lagrange Interpolation Polynomial for Neural Network Learning



- **Data Analysis and Visualization**: to estimate missing or incomplete data values, or to smooth out noisy data.
- **Geographic Information Systems (GIS):** to estimate the values of geographic features, such as elevation, precipitation, or temperature, at locations where measurements are not available.
- **Image Processing:** to enhance the quality of images, such as resizing or rescaling, or to fill in missing pixels in images.
- **Computer Graphics:** to create smooth animations, morphing effects, or to interpolate between key frames in animations.

Types of Interpolation

There are three major types of numerical interpolation .

1. Polynomial Interpolation (Lagrange Interpolation)
    a) Linear Interpolation
    b) Quadratic Interpolation
2. Spline Interpolation

1. Polynomial Interpolation

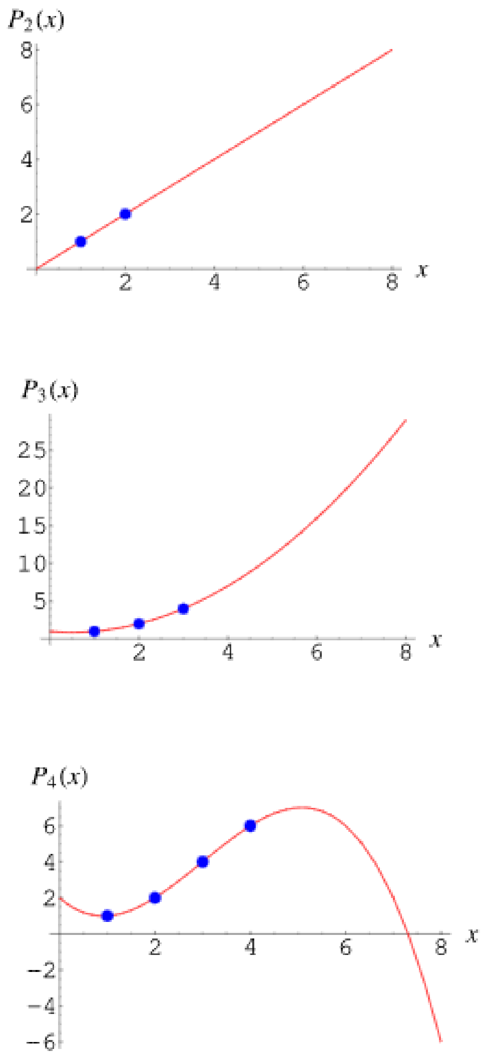
Polynomial interpolation is a method of approximating a function by a polynomial of a specified degree that passes through a set of given points.

Given a set of $n+1$ distinct data points (x_0, y_0), (x_1, y_1), ….., (x_n, y_n), there is only one polynomial of degree at most n in the form

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots \ldots . + a_n x^n$$

That goes through all the points.

Polynomial interpolation involves finding a polynomial of degree n or less that passes through these points. The resulting polynomial can be used to approximate the original function at other points within the range of the given data points.

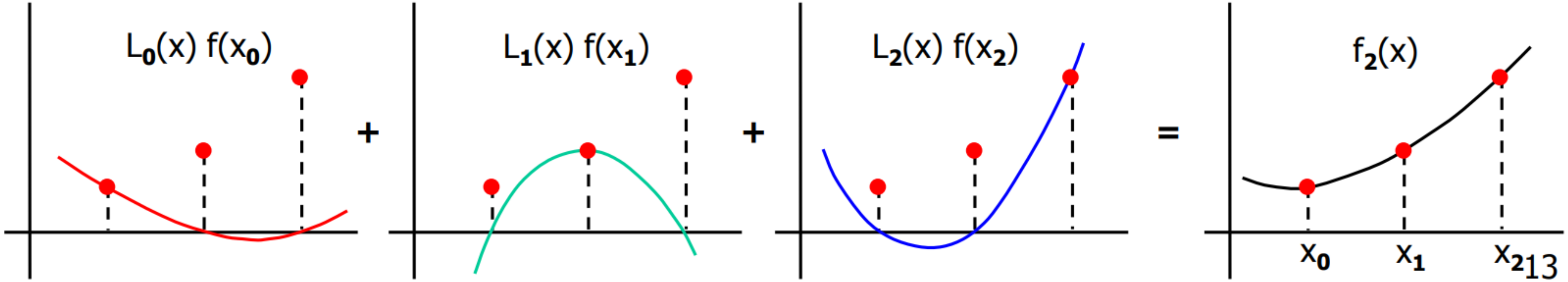


Lagrange Interpolation - General

Given $(x_0,\ f_0),\ (x_1,\ f_1),\ \ldots\ldots\ (x_n,\ f_n)$ with arbitrarily spaced x_j, Lagrange had the idea of multiplying each f_j by a polynomial that is 1 at x_j and 0 at the other n nodes and then taking the sum of these n+1 polynomials.

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i) \qquad \text{where} \qquad L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

$$\text{Lagrange functions} \quad L_i(x) = \begin{cases} 1 & \text{at} \quad x = x_i \\ 0 & \text{at} \quad \text{all other data points} \end{cases}$$



Department of Interdisciplinary Studies, Faculty of Engineering, University of Jaffna, Sri Lanka

1.1 Lagrange Interpolation: Linear Interpolation

Linear interpolation (first-degree Lagrange interpolating polynomial) is interpolation by the straight line through (x_0, f_0), (x_1, f_1);
Thus, the linear Lagrange polynomial, p_1 is a sum

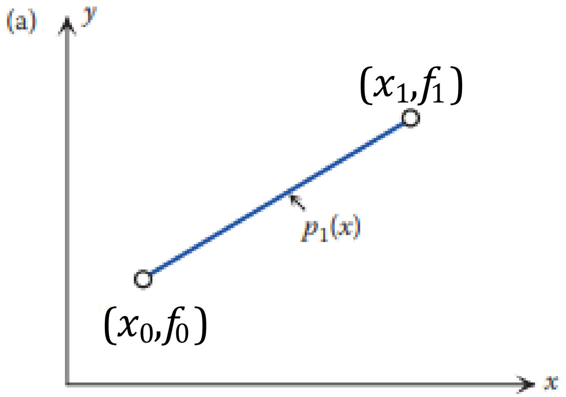
$$p_1(x) = L_0(x)f_0 + L_1(x)f_1$$

with $L_0(x)$ and $L_1(x)$ are the Lagrange coefficient functions
Where $L_0(x_0) = 1$ *and* $L_0(x_1) = 0$; similarly, $L_1(x_0) = 0$ *and* $L_1(x_1) = 1$

$$L_0(x) = \frac{x - x_1}{x_0 - x_1}, \qquad L_1(x) = \frac{x - x_0}{x_1 - x_0}.$$

This gives the linear Lagrange polynomial

$$p_1(x) = L_0(x)f_0 + L_1(x)f_1 = \frac{x - x_1}{x_0 - x_1} \cdot f_0 + \frac{x - x_0}{x_1 - x_0} \cdot f_1.$$



Example : 1

Compute a 4D-value of ln 9.2 from ln 9.0 = 2.1972, ln 9.5 = 2.2513 by linear Lagrange interpolation and determine the error, using ln 9.2 = 2.2192 (4D).

Solution. $x_0 = 9.0, x_1 = 9.5, f_0 = \ln 9.0, f_1 = \ln 9.5$. Ln (2) we need

$$L_0(x) = \frac{x - 9.5}{-0.5} = -2.0(x - 9.5), \qquad L_0(9.2) = -2.0(-0.3) = 0.6$$

$$L_1(x) = \frac{x - 9.0}{0.5} = 2.0(x - 9.0), \qquad L_1(9.2) = 2 \cdot 0.2 = 0.4$$

$$\ln 9.2 \approx p_1(9.2) = L_0(9.2) f_0 + L_1(9.2) f_1 = 0.6 \cdot 2.1972 + 0.4 \cdot 2.2513 = 2.2188.$$

The error is $\epsilon = a - \tilde{a} = 2.2192 - 2.2188 = 0.0004$. Hence linear interpolation is not sufficient here to get 4D accuracy; it would suffice for 3D accuracy.

1.2 Lagrange Interpolation: Quadratic interpolation

Quadratic interpolation (Second-degree Lagrange interpolation) is interpolation of given three points (x_0, f_0), (x_1, f_1), (x_2, f_2) by a second degree polynomial $p_2(x)$, in the form;

$$p_2(x) = L_0(x)f_0 + L_1(x)f_1 + L_2(x)f_2$$

with $L_0(x_0) = 1, L_1(x_1) = 1, L_2(x_2) = 1$, and $L_0(x_1) = L_0(x_2) = 0$, etc. We claim that

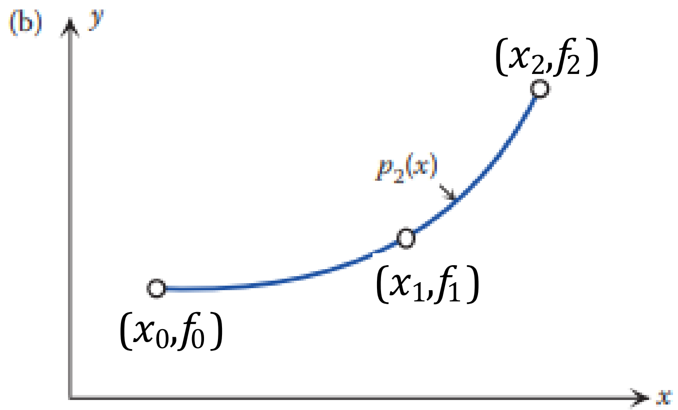
$$L_0(x) = \frac{l_0(x)}{l_0(x_0)} = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

$$L_1(x) = \frac{l_1(x)}{l_1(x_1)} = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$L_2(x) = \frac{l_2(x)}{l_2(x_2)} = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

$$L_k(x_j) = 0 \text{ if } j \neq k..$$

$$L_k(x_k) = 1$$



Example : 2

Compute 4D-value of $\ln 9.2$ from three data points, $\ln 9.0 = 2.1972, \ln 9.5 = 2.2513$ and $\ln 11.0 = 2.3979$ using a second-degree Lagrange polynomial interpolation function

Solution.

$$L_0(x) = \frac{(x-9.5)(x-11.0)}{(9.0-9.5)(9.0-11.0)} = x^2 - 20.5x + 104.5, \qquad L_0(9.2) = 0.5400,$$

$$L_1(x) = \frac{(x-9.0)(x-11.0)}{(9.5-9.0)(9.5-11.0)} = -\frac{1}{0.75}(x^2 - 20x + 99), \qquad L_1(9.2) = 0.4800,$$

$$L_2(x) = \frac{(x-9.0)(x-9.5)}{(11.0-9.0)(11.0-9.5)} = \frac{1}{3}(x^2 - 18.5x + 85.5), \qquad L_2(9.2) = -0.0200,$$

$$\ln 9.2 \approx p_2(9.2) = 0.5400 \cdot 2.1972 + 0.4800 \cdot 2.2513 - 0.0200 \cdot 2.3979 = 2.2192.$$

Here are some of the key differences between the different types of polynomial interpolation based on their degree:

- Linear interpolation: Linear interpolation involves fitting a straight line to two data points and is therefore a first-degree polynomial. It is the simplest form of polynomial interpolation and is often used in simple applications where the data is relatively smooth and linear. However, it can be inaccurate for data that exhibits more complex behavior.
- Quadratic interpolation: Quadratic interpolation involves fitting a parabolic curve to three data points and is therefore a second-degree polynomial. It is more accurate than linear interpolation and can capture more complex behavior in the data. However, it can still lead to overfitting or oscillations, especially if the data is noisy or has large gaps.

- Cubic interpolation: Cubic interpolation involves fitting a cubic curve to four data points and is therefore a third-degree polynomial. It is more accurate than quadratic interpolation and can capture even more complex behavior in the data. It is also less likely to lead to overfitting or oscillations but can still be sensitive to noise or gaps in the data.

- Higher-degree polynomial interpolation: Higher-degree polynomial interpolation involves fitting higher-degree polynomials to more data points and can potentially give very accurate approximations of the underlying function. However, it is also more prone to overfitting and oscillations, and can be computationally expensive for large datasets.

2. Spline Interpolation

- When there are a small number of points in the data, the degree of the interpolating polynomial will also be small, and the interpolated values are generally accurate. However, when a high-degree polynomial is used to interpolate a large number of points, large errors in interpolation are possible, as shown in Figure. The main contributing factor is the large number of peaks and valleys that accompany a high-degree polynomial.

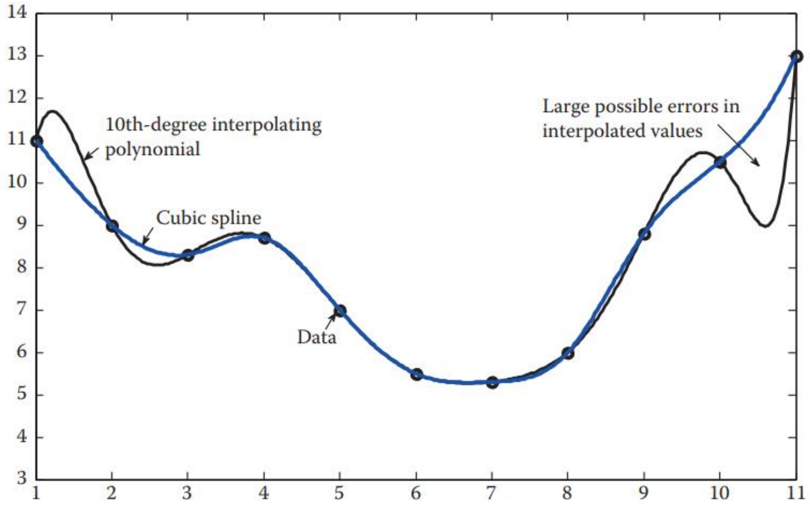
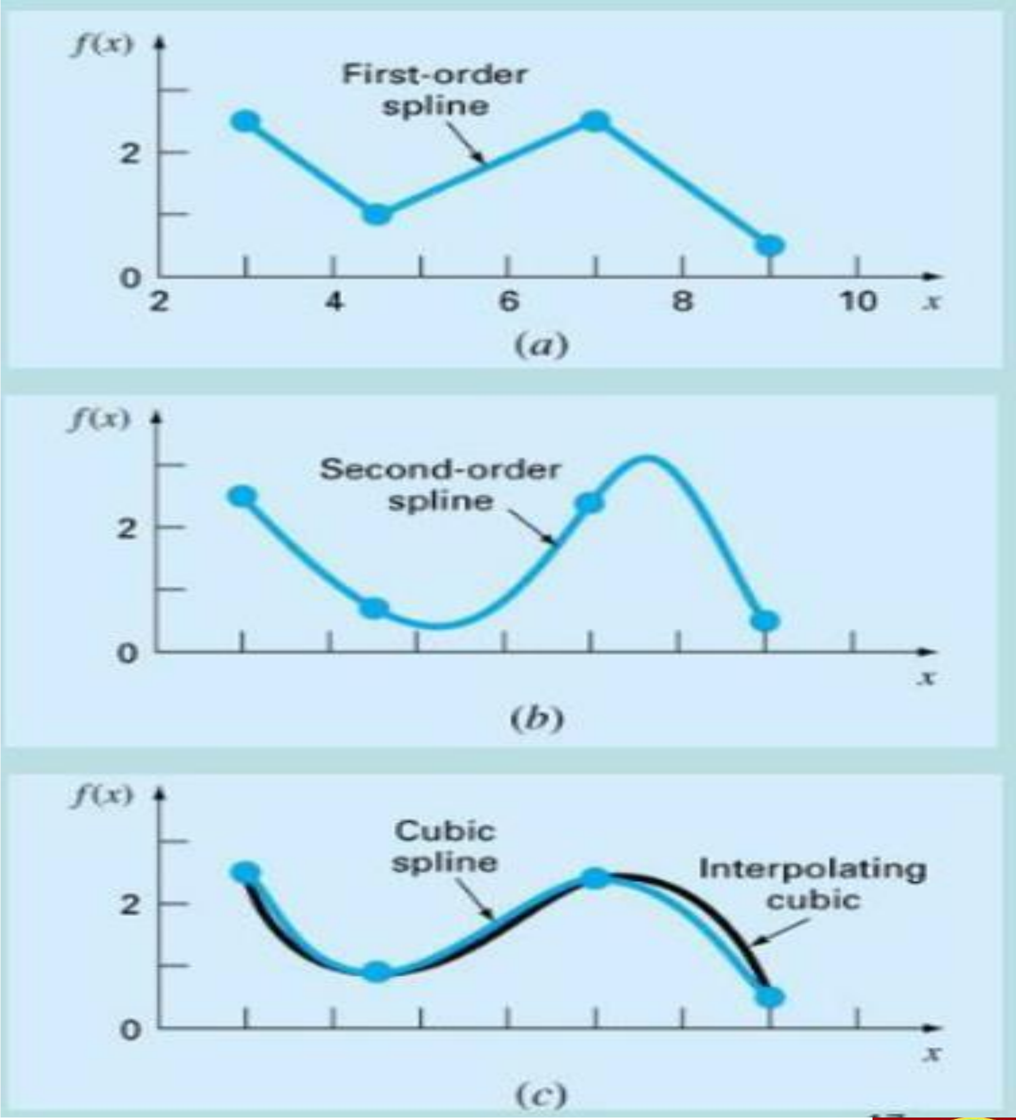


FIGURE 5.18
A 10th-degree interpolating polynomial and cubic splines for a set of 11 data points.

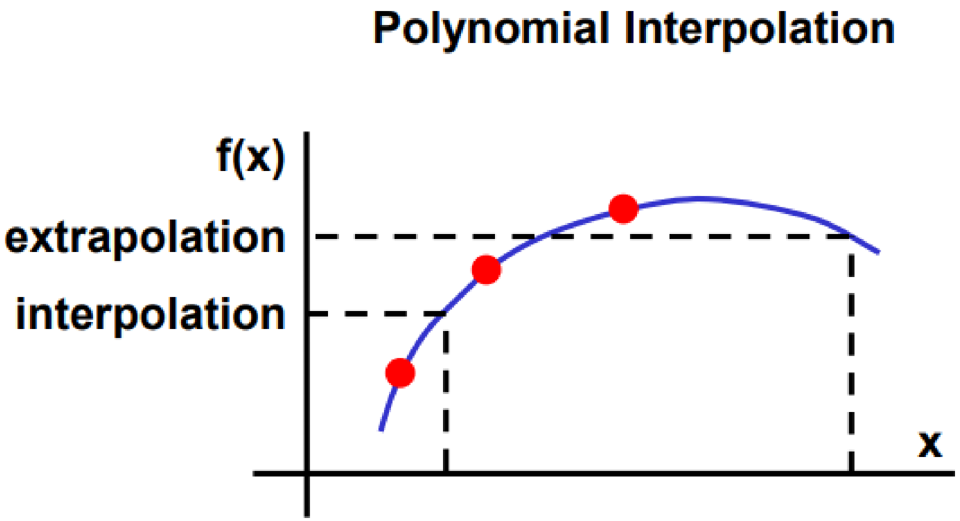
2. Spline Interpolation

- These situations may be avoided by using several low-degree polynomials, each of which is valid in one interval between one or more data points. The low degree of each polynomial in turn limits the number of peaks and valleys to a low number, hence reducing the possibility of large deviations from the main theme of the data. These low degree polynomials are known as **Splines**. The data points at which two splines meet are called knots.

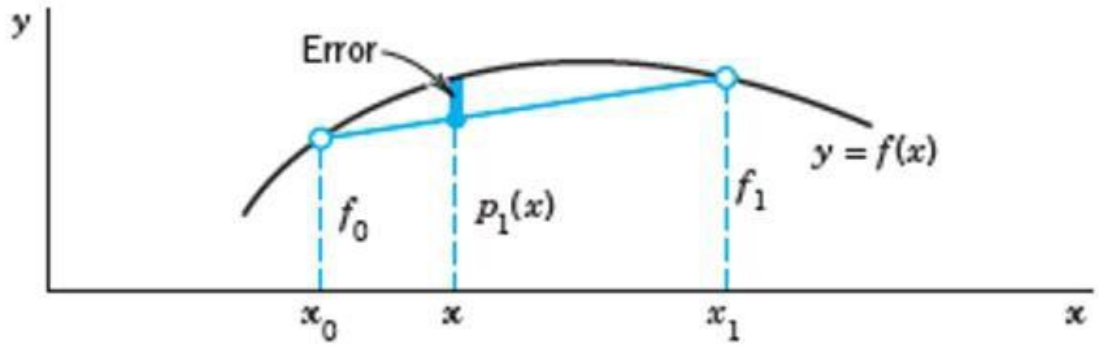


Extrapolation

Extrapolation is the process of estimating or predicting a value beyond the range of a given set of data points or observations. In other words, it involves extending a function or a curve outside the domain of the data used to define it.



Error Estimate



If f is itself a polynomial of degree n (or less), it must coincide with p_n because the $n+1$ data $(x_0, f_0), (x_1, f_1), \dots \dots \dots \ (x_n, f_n)$ determine a polynomial uniquely, so the error is zero. Now the special f has its $(n+1)^{\text{st}}$ derivative identically zero. This makes it plausible that for a general f its $(n+1)^{\text{st}}$ derivative $f^{(n+1)}$ should measure the error.

Error of Lagrange polynomial approximations

It is important to understand the nature of the error term when the Lagrange polynomial is used to approximate a continuous function $f(x)$. It is similar to the error term for the Taylor polynomial, except that the factor $(x - x_0)^{N+1}$ is replaced with the product $(x - x_0)(x - x_1)\cdots(x - x_N)$. This is expected because interpolation is exact at each of the $N + 1$ nodes x_k, where we have $E_N(x_k) = f(x_k) - P_N(x_k) = y_k - y_k = 0$ for $k = 0, 1, 2, \ldots, N$.

Theorem 4.3 (Lagrange Polynomial Approximation). Assume that $f \in C^{N+1}[a, b]$ and that $x_0, x_1, \ldots, x_N \in [a, b]$ are $N + 1$ nodes. If $x \in [a, b]$, then

$$f(x) = P_N(x) + E_N(x)$$

Where $P_N(x)$ is a polynomial that can be used to approximate $f(x)$:

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x)$$

The error term $E_N(x)$ has the form

$$E_N(x) = \frac{(x - x_0)(x - x_1)\ldots.(x - x_N) f^{(N+1)}(c)}{(N + 1)!}$$

For some value $c = c(x)$ that lies in the interval $[a, b]$.

Example : 3

Estimate the error in the computation of 4D-value of $\ln 9.2$ from data points, $\ln 9.0 = 2.1972, \ln 9.5 = 2.2513$ using a first-degree Lagrange polynomial interpolation function

$$E_N(x) = \frac{(x - x_0)(x - x_1)\ldots.(x - x_N)f^{(N+1)}(c)}{(N+1)!}$$

Formula gives the error for ***any*** *polynomial interpolation method if $f(x)$ has a continuous $(n+1)$st derivative.*

EXAMPLE **Error Estimate of Linear Interpolation.**

Solution. We have $n = 1$, $f(t) = \ln t$, $f'(t) = 1/t$, $f''(t) = -1/t^2$. Hence

$$\epsilon_1(x) = (x - 9.0)(x - 9.5)\frac{(-1)}{2t^2}, \qquad \text{thus} \qquad \epsilon_1(9.2) = \frac{0.03}{t^2}.$$

$t = 9$ gives the maximum $0.03/9^2 = 0.00037$ and $t = 9.5$ gives the minimum $0.03/9.5^2 = 0.00033$, so that we get $0.00033 \leqq \epsilon_1(9.2) \leqq 0.00037$, or better, 0.00038 because $0.3/81 = 0.003703\cdots$.

Thank You