CMPT-439

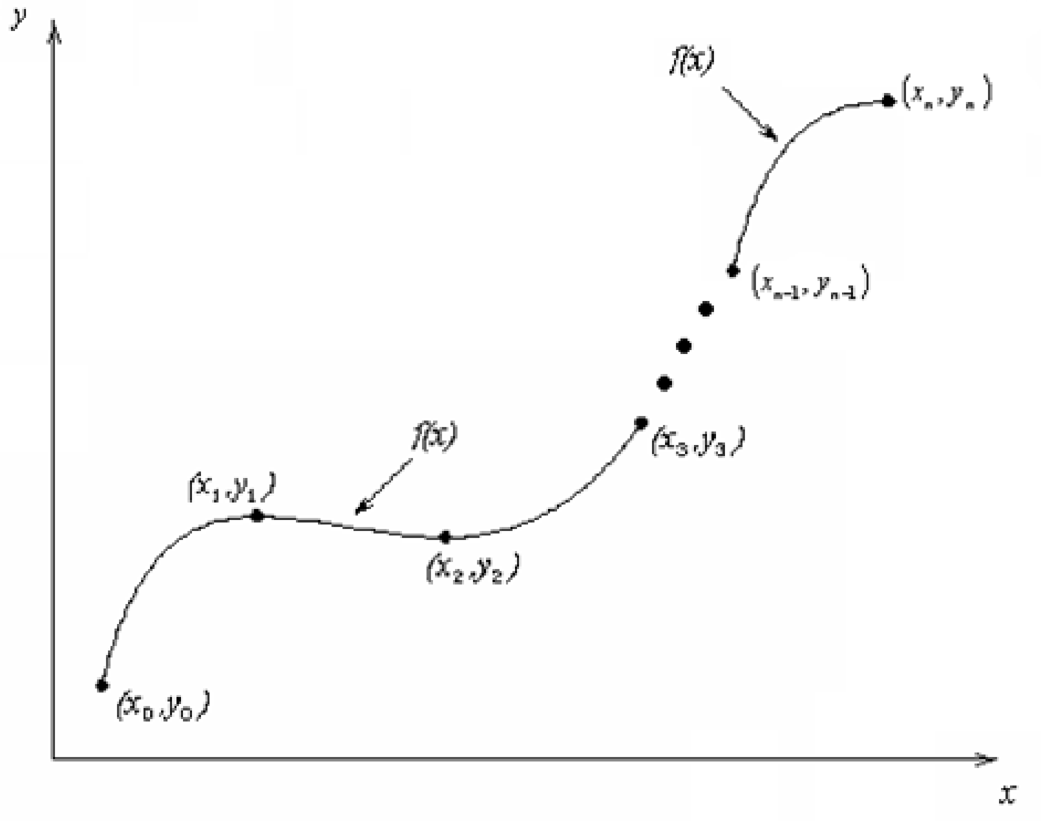
Numerical Computation

Fall 2020

Interpolation
Interpolanting Polymonials

What is Interpolation ?

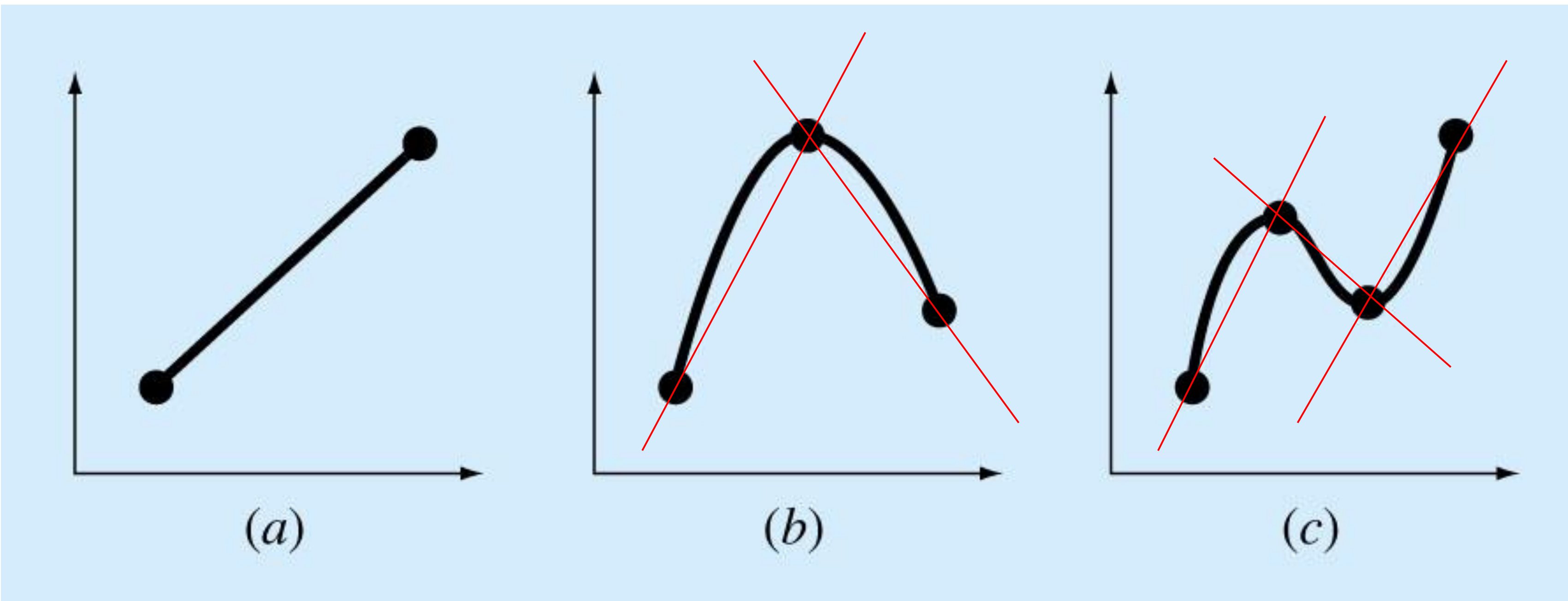
Given (x_0,y_0), (x_1,y_1), …… (x_n,y_n), find the value of y at a value of x that is not given.



Interpolation

- Interpolation is the estimation of intermediate values between precise data points by approaching them using interpolanting functions (interpolants) whose values in precise data points coincide with the ones of a function to be interpolated

Linear Interpolation



Interpolants

Polynomials are the most common choice of interpolants because they are easy to:

- Evaluate
- Differentiate
- Integrate

Interpolation

- Thus interpolation is the estimation of intermediate values between precise data points. The most common method is:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

- Although **there is one and only one n^{th}-order polynomial that fits n+1 points**, there are various mathematical formats in which this polynomial can be expressed. The most popular are:
  - The Newton polynomial
  - The Lagrange polynomial

Direct Polynomial Interpolation

Given n+1 data points (x_0,y_0), (x_1,y_1), …… (x_n,y_n), pass a polynomial of order n through the data as given below:

$$y(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

where a_0, a_1,………………. a_n are real constants.

- Set up n+1 equations to find n+1 constants.
- To find the value y at a given value of x, simply substitute the value of x in the above polynomial.

Coefficients of an Interpolating Polynomial

- Since n+1 data points are required to determine n+1 coefficients, simultaneous system of linear algebraic equations can be used to calculate "a"s:

$$
\begin{aligned}
f(x_0) &= a_0 + a_1 x_0 + a_2 x_0^2 \cdots + a_n x_0^n \\
f(x_1) &= a_0 + a_1 x_1 + a_2 x_1^2 \cdots + a_n x_1^n \\
&\vdots \\
f(x_n) &= a_0 + a_1 x_n + a_2 x_n^2 \cdots + a_n x_n^n
\end{aligned}
$$

- **where 'x's are the knowns** while **'a's are the unknowns**

Example 1

The upward velocity of a rocket is given as a function of time in Table 1.

Find the velocity at $t=16$ seconds using the direct method for linear interpolation.

Table 1 Velocity as a function of time.

$t, (\text{s})$	$v(t), (\text{m/s})$
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

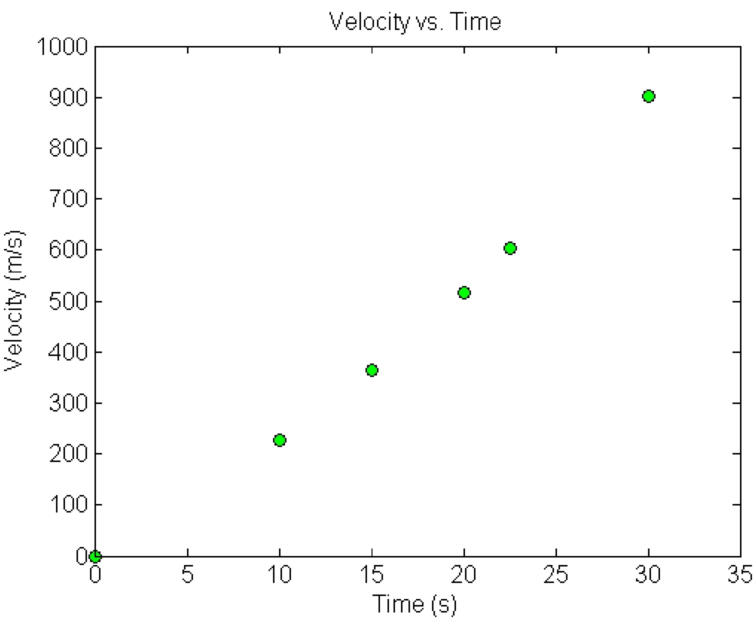


Figure 2 Velocity vs. time data for the rocket example

Example 1: Solution

$$v(t) = a_0 + a_1 t$$

$$v(15) = a_0 + a_1(15) = 362.78$$

$$v(20) = a_0 + a_1(20) = 517.35$$

Solving the above two equations gives,

$$a_0 = -100.93 \quad a_1 = 30.914$$

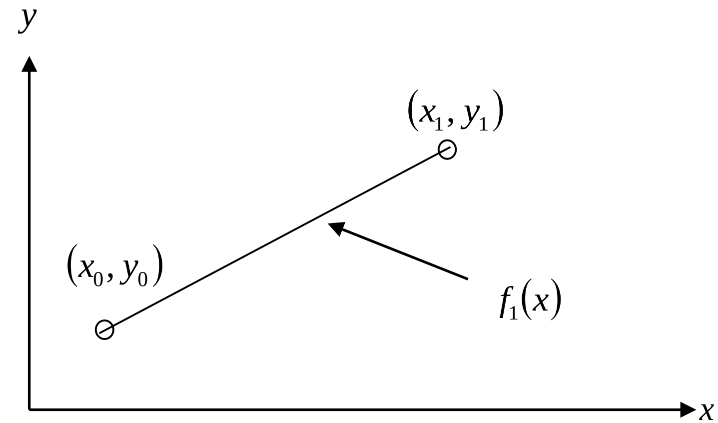


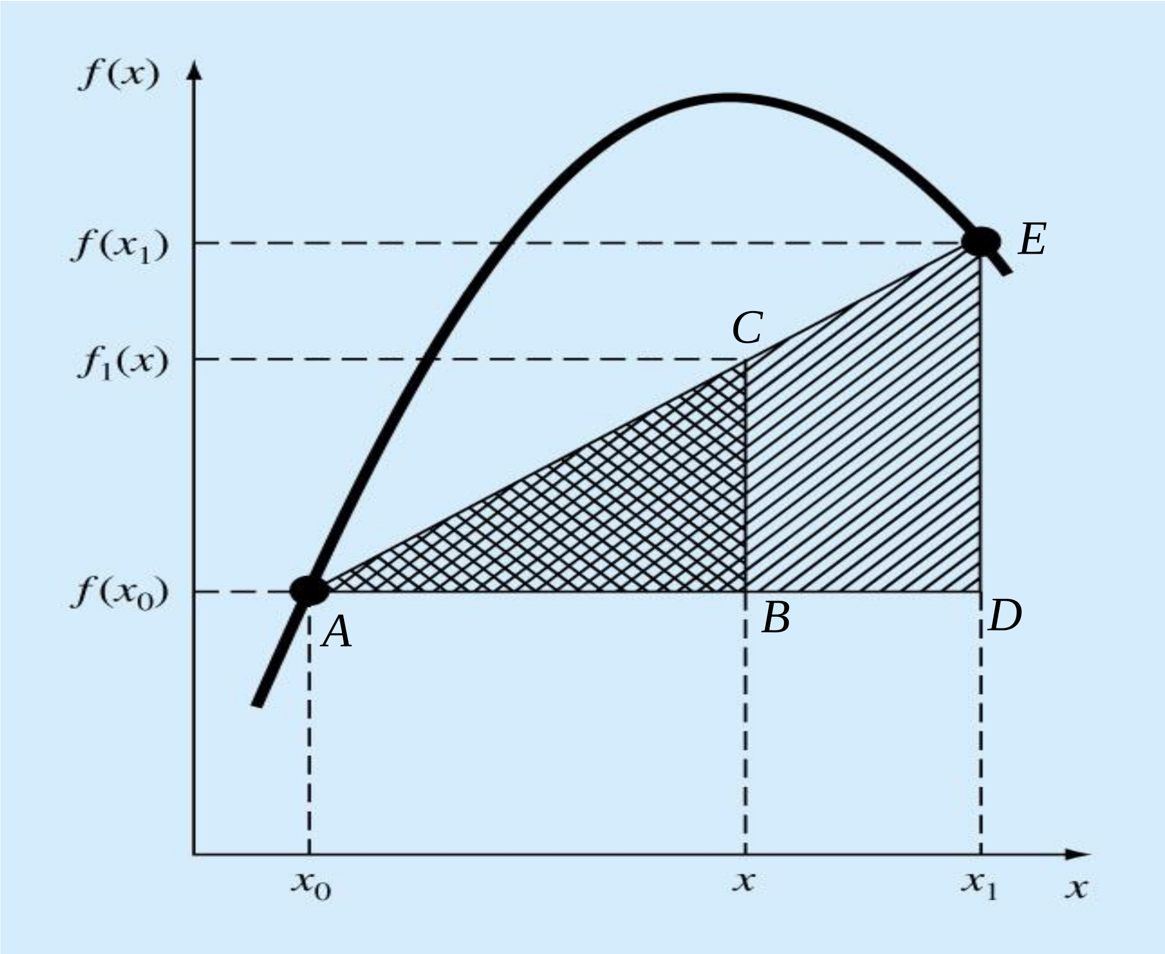
Figure 3 Linear interpolation.

Hence

$$v(t) = -100.93 + 30.914t, \ 15 \le t \le 20.$$

$$v(16) = -100.93 + 30.914(16) = 393.7 \text{ m/s}$$

Newton's Divided-Difference Interpolating Polynomials



$$ABC \sim ADE \rightarrow \frac{BC}{AB} = \frac{DE}{AD}$$

$$BC = f_1(x) - f(x_0)$$

$$AB = x - x_0$$

$$DE = f(x_1) - f(x_0)$$

$$AD = x_1 - x_0$$

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Newton's Divided-Difference Interpolating Polynomials

Linear Interpolation

- Is the simplest form of interpolation, connecting two data points with a straight line.

Slope and a finite divided difference approximation to 1st derivative

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$\boxed{f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)}$$

Linear-interpolation formula

- $f_1(x)$ designates that this is a first-order interpolating polynomial

Newton's Divided-Difference Interpolating Polynomials

Quadratic Interpolation

- If three data points are available, the estimate is improved by introducing some curvature into the line connecting the points: $f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1)$
- A simple procedure can be used to determine the values of the coefficients:

$$x = x_0 \qquad b_0 = f\left(x_0\right)$$

$$x = x_1 \qquad b_1 = \frac{f\left(x_1\right) - f\left(x_0\right)}{x_1 - x_0}$$

$$x = x_2 \qquad b_2 = \frac{\dfrac{f\left(x_2\right) - f\left(x_1\right)}{x_2 - x_1} - \dfrac{f\left(x_1\right) - f\left(x_0\right)}{x_1 - x_0}}{x_2 - x_0}$$

General Form of Newton's Interpolating Polynomials

$$f_n(x) = f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0]$$
$$+ \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})f[x_n, x_{n-1}, \cdots, x_0]$$

$$b_0 = f(x_0)$$
$$b_1 = f[x_1, x_0]$$
$$b_2 = f[x_2, x_1, x_0]$$
$$\vdots$$
$$b_n = f[x_n, x_{n-1}, \cdots, x_1, x_0]$$

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j}$$

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k}$$

$$\vdots$$

$$f[x_n, x_{n-1}, \ldots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \ldots, x_1] - f[x_{n-1}, x_{n-2}, \ldots, x_0]}{x_n - x_0}$$

Bracketed function evaluations are finite divided differences

Lagrange Interpolating Polynomials

- The Lagrange interpolating polynomial is simply a reformulation of the Newton's polynomial that **avoids** the computation of divided differences:

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Lagrange Interpolating Polynomials

Lagrangian interpolating polynomial is given by

$$f_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

where 'n' in $f_n(x)$ stands for the n^{th} order polynomial that approximates the function $y = f(x)$ given at $(n+1)$ data points as $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1}), (x_n, y_n)$, and

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

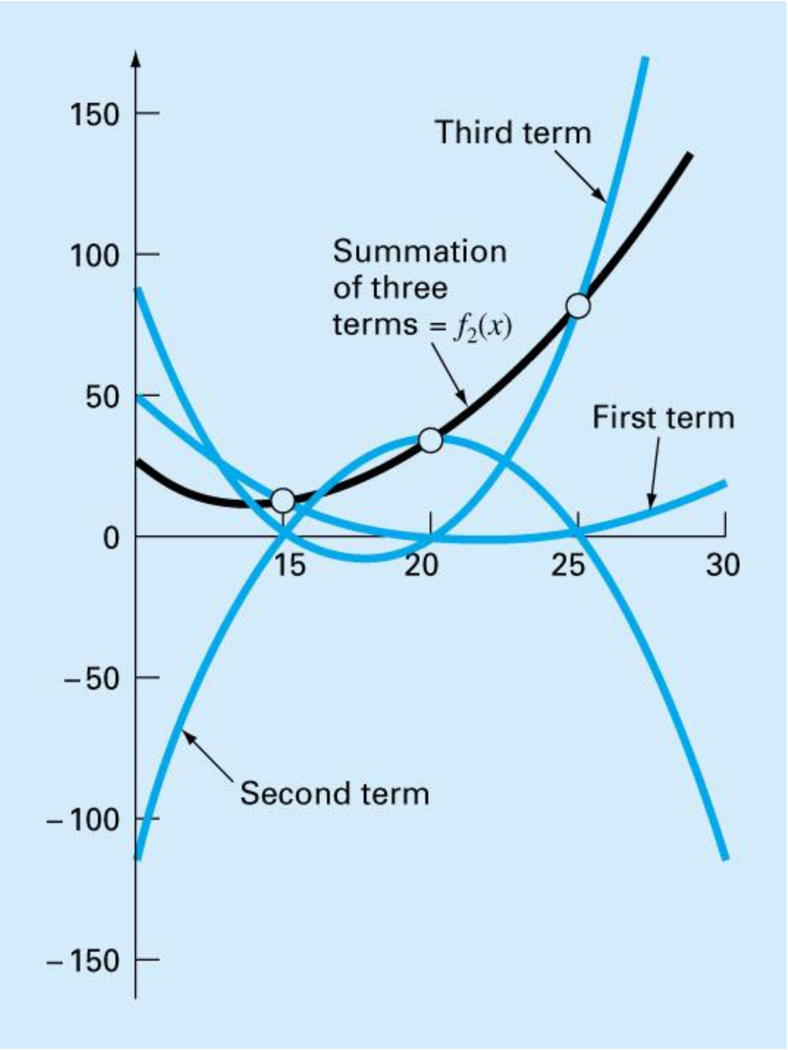
$L_i(x)$ is a weighting function that includes a product of $(n-1)$ terms with terms of $j = i$ omitted.

Lagrange Interpolating Polynomials

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

$$f_2(x) = \underbrace{\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}}_{L_0} f(x_0) + \underbrace{\frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}}_{L_1} f(x_1) + \underbrace{\frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}}_{L_2} f(x_2)$$

Lagrange Interpolating Polynomials



Error of Interpolation

- For an n^{th}-order interpolating polynomial, the interpolation error is:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\cdots(x-x_n)$$

- ξ is located in some interval containing the data points: $\left[x, x_0, x_1, \ldots, x_n\right]$

Example

The upward velocity of a rocket is given as a function of time in Table 1. Find the velocity at t=16 seconds using the Lagrangian method for linear interpolation.

Table Velocity as a function of time

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

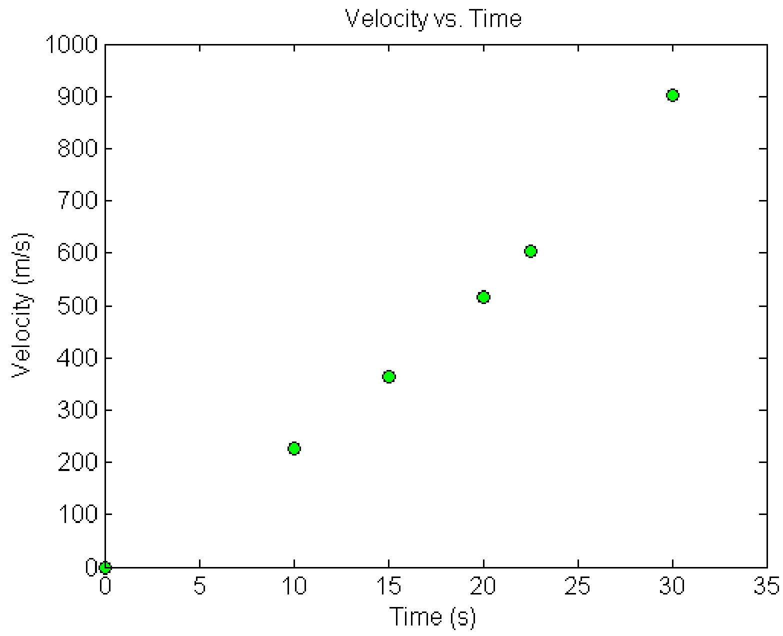


Figure. Velocity vs. time data for the rocket example

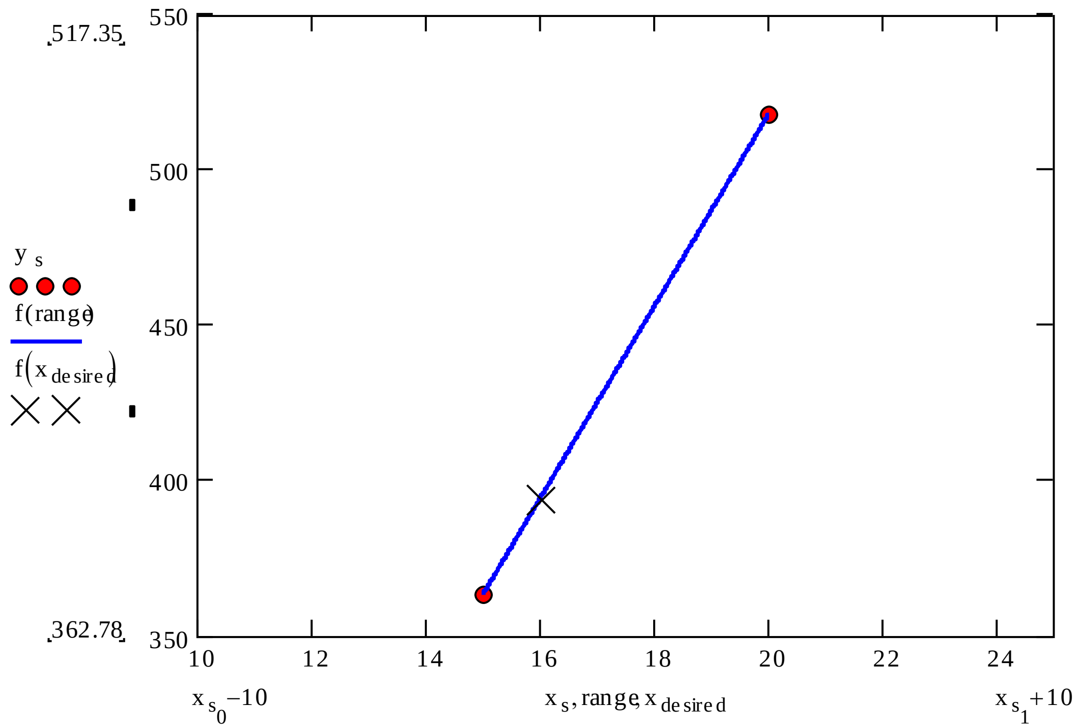
Linear Interpolation

$$v(t) = \sum_{i=0}^{1} L_i(t)v(t_i)$$

$$= L_0(t)v(t_0) + L_1(t)v(t_1)$$

$$t_0 = 15, v(t_0) = 362.78$$

$$t_1 = 20, v(t_1) = 517.35$$



Linear Interpolation (contd)

$$L_0(t) = \prod_{\substack{j=0 \\ j\neq 0}}^{1} \frac{t - t_j}{t_0 - t_j} = \frac{t - t_1}{t_0 - t_1}$$

$$L_1(t) = \prod_{\substack{j=0 \\ j\neq 1}}^{1} \frac{t - t_j}{t_1 - t_j} = \frac{t - t_0}{t_1 - t_0}$$

$$v(t) = \frac{t - t_1}{t_0 - t_1} v(t_0) + \frac{t - t_0}{t_1 - t_0} v(t_1) = \frac{t - 20}{15 - 20}(362.78) + \frac{t - 15}{20 - 15}(517.35)$$

$$v(16) = \frac{16 - 20}{15 - 20}(362.78) + \frac{16 - 15}{20 - 15}(517.35)$$

$$= 0.8(362.78) + 0.2(517.35)$$

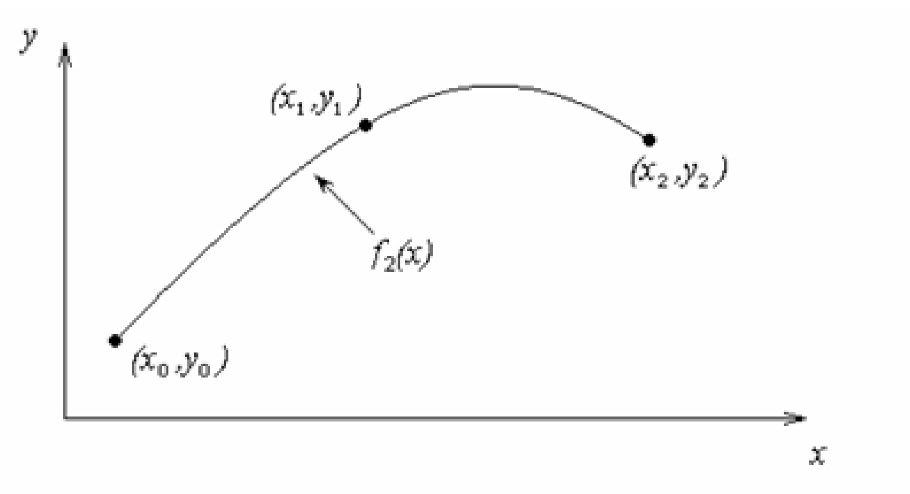
$$= 393.7 \text{ m/s.}$$

Quadratic Interpolation

For the second order polynomial interpolation (also called quadratic interpolation), we choose the velocity given by

$$v(t) = \sum_{i=0}^{2} L_i(t) v(t_i)$$

$$= L_0(t)v(t_0) + L_1(t)v(t_1) + L_2(t)v(t_2)$$



Example

The upward velocity of a rocket is given as a function of time in Table 1. Find the velocity at $t=16$ seconds using the Lagrangian method for quadratic interpolation.

Table Velocity as a function of time

t (s)	$v(t)$ (m/s)
0	0
10	227.04
15	362.78
20	517.35
22.5	602.97
30	901.67

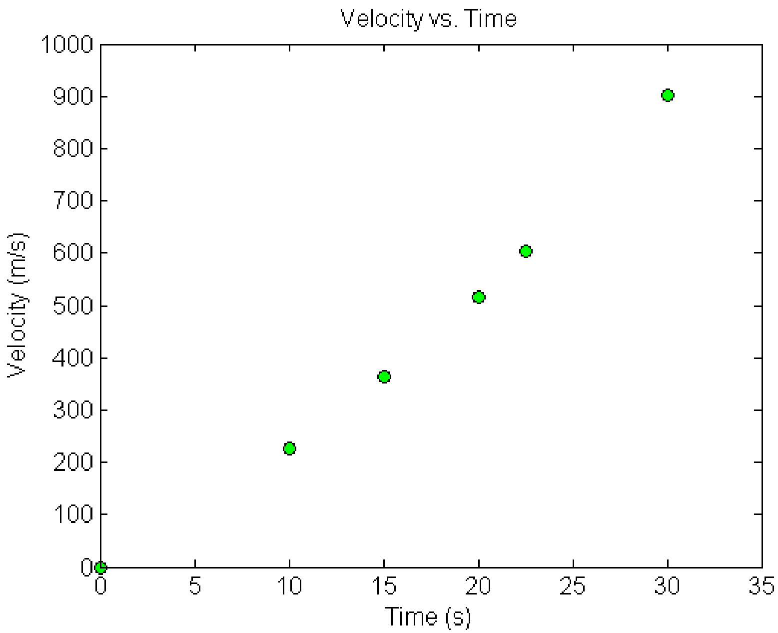


Figure. Velocity vs. time data for the rocket example

Quadratic Interpolation

$$t_0 = 10,\ v(t_0) = 227.04$$

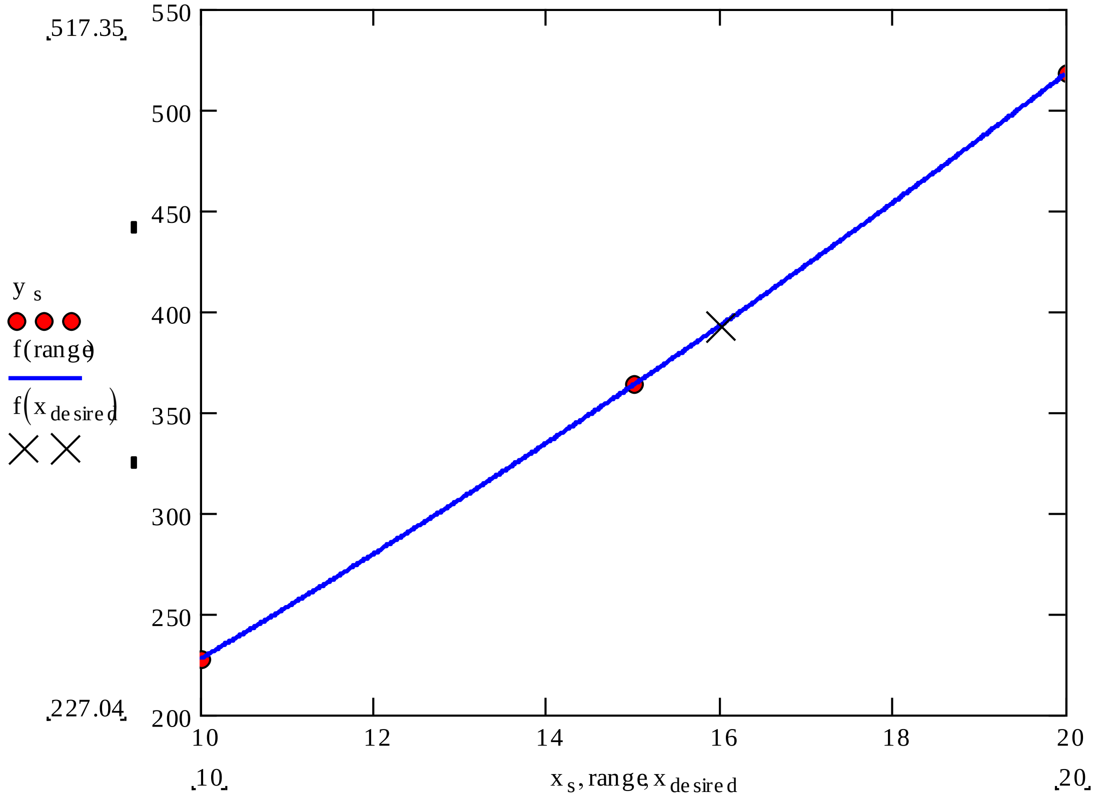
$$t_1 = 15,\ v(t_1) = 362.78$$

$$t_2 = 20,\ v(t_2) = 517.35$$

$$L_0(t) = \prod_{\substack{j=0 \\ j\neq 0}}^{2} \frac{t - t_j}{t_0 - t_j} = \left(\frac{t - t_1}{t_0 - t_1}\right)\left(\frac{t - t_2}{t_0 - t_2}\right)$$

$$L_1(t) = \prod_{\substack{j=0 \\ j\neq 1}}^{2} \frac{t - t_j}{t_1 - t_j} = \left(\frac{t - t_0}{t_1 - t_0}\right)\left(\frac{t - t_2}{t_1 - t_2}\right)$$

$$L_2(t) = \prod_{\substack{j=0 \\ j\neq 2}}^{2} \frac{t - t_j}{t_2 - t_j} = \left(\frac{t - t_0}{t_2 - t_0}\right)\left(\frac{t - t_1}{t_2 - t_1}\right)$$



Quadratic Interpolation

$$v(t) = \left(\frac{t-t_1}{t_0-t_1}\right)\left(\frac{t-t_2}{t_0-t_2}\right)v(t_0) + \left(\frac{t-t_0}{t_1-t_0}\right)\left(\frac{t-t_2}{t_1-t_2}\right)v(t_1) + \left(\frac{t-t_0}{t_2-t_0}\right)\left(\frac{t-t_1}{t_2-t_1}\right)v(t_2)$$

$$\begin{aligned} v(16) &= \left(\frac{16-15}{10-15}\right)\left(\frac{16-20}{10-20}\right)(227.04) + \left(\frac{16-10}{15-10}\right)\left(\frac{16-20}{15-20}\right)(362.78) + \left(\frac{16-10}{20-10}\right)\left(\frac{16-15}{20-15}\right)(517.35) \\ &= (-0.08)(227.04) + (0.96)(362.78) + (0.12)(527.35) \\ &= 392.19 \text{ m/s} \end{aligned}$$

The absolute relative approximate error $|\in_a|$ obtained between the results from the first and second order polynomial is

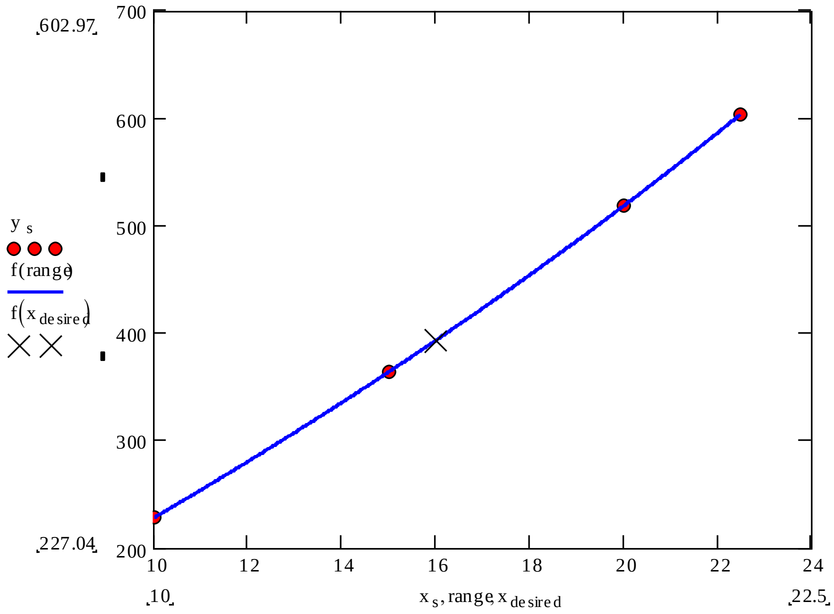
$$\begin{aligned} |\in_a| &= \left|\frac{392.19 - 393.70}{392.19}\right| \times 100 \\ &= 0.38410\% \end{aligned}$$

Cubic Interpolation

For the third order polynomial (also called cubic interpolation), we choose the velocity given by

$$v(t) = \sum_{i=0}^{3} L_i(t)v(t_i)$$

$$= L_0(t)v(t_0) + L_1(t)v(t_1) + L_2(t)v(t_2) + L_3(t)v(t_3)$$



Cubic Interpolation (contd)

$$t_o = 10, \; v(t_o) = 227.04 \qquad t_1 = 15, \; v(t_1) = 362.78$$

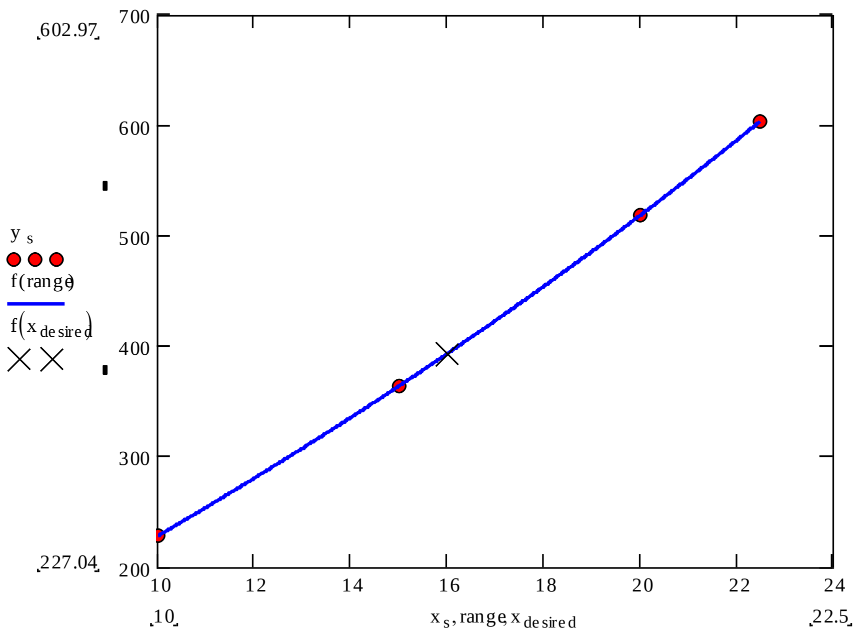
$$t_2 = 20, \; v(t_2) = 517.35 \qquad t_3 = 22.5, \; v(t_3) = 602.97$$

$$L_0(t) = \prod_{\substack{j=0 \\ j \neq 0}}^{3} \frac{t - t_j}{t_0 - t_j} = \left(\frac{t - t_1}{t_0 - t_1}\right)\left(\frac{t - t_2}{t_0 - t_2}\right)\left(\frac{t - t_3}{t_0 - t_3}\right);$$

$$L_1(t) = \prod_{\substack{j=0 \\ j \neq 1}}^{3} \frac{t - t_j}{t_1 - t_j} = \left(\frac{t - t_0}{t_1 - t_0}\right)\left(\frac{t - t_2}{t_1 - t_2}\right)\left(\frac{t - t_3}{t_1 - t_3}\right)$$

$$L_2(t) = \prod_{\substack{j=0 \\ j \neq 2}}^{3} \frac{t - t_j}{t_2 - t_j} = \left(\frac{t - t_0}{t_2 - t_0}\right)\left(\frac{t - t_1}{t_2 - t_1}\right)\left(\frac{t - t_3}{t_2 - t_3}\right);$$

$$L_3(t) = \prod_{\substack{j=0 \\ j \neq 3}}^{3} \frac{t - t_j}{t_3 - t_j} = \left(\frac{t - t_0}{t_3 - t_0}\right)\left(\frac{t - t_1}{t_3 - t_1}\right)\left(\frac{t - t_2}{t_3 - t_2}\right)$$



Cubic Interpolation (contd)

$$v(t)=\left(\frac{t-t_1}{t_0-t_1}\right)\left(\frac{t-t_2}{t_0-t_2}\right)\left(\frac{t-t_3}{t_0-t_3}\right)v(t_1)+\left(\frac{t-t_0}{t_1-t_0}\right)\left(\frac{t-t_2}{t_1-t_2}\right)\left(\frac{t-t_3}{t_1-t_3}\right)v(t_2)$$

$$+\left(\frac{t-t_0}{t_2-t_0}\right)\left(\frac{t-t_1}{t_2-t_1}\right)\left(\frac{t-t_3}{t_2-t_3}\right)v(t_2)+\left(\frac{t-t_1}{t_3-t_1}\right)\left(\frac{t-t_1}{t_3-t_1}\right)\left(\frac{t-t_2}{t_3-t_2}\right)v(t_3)$$

$$v(16)=\left(\frac{16-15}{10-15}\right)\left(\frac{16-20}{10-20}\right)\left(\frac{16-22.5}{10-22.5}\right)(227.04)+\left(\frac{16-10}{15-10}\right)\left(\frac{16-20}{15-20}\right)\left(\frac{16-22.5}{15-22.5}\right)(362.78)$$

$$+\left(\frac{16-10}{20-10}\right)\left(\frac{16-15}{20-15}\right)\left(\frac{16-22.5}{20-22.5}\right)(517.35)+\left(\frac{16-10}{22.5-10}\right)\left(\frac{16-15}{22.5-15}\right)\left(\frac{16-20}{22.5-20}\right)(602.97)$$

$$=(-0.0416)(227.04)+(0.832)(362.78)+(0.312)(517.35)+(-0.1024)(602.97)$$

$$=392.06\ \mathrm{m/s}$$

The absolute relative approximate error $|\in_a|$ obtained between the results from the second and third order polynomial is

$$|\in_a|=\left|\frac{392.06-392.19}{392.06}\right|\times 100$$

$$=0.033269\%$$

Comparison Table

Order of Polynomial	**1**	**2**	**3**
v(t=16) m/s	393.69	392.19	392.06
Absolute Relative Approximate Error	--------	0.38410%	0.033269%

Lagrangian Interpolation: Algorithm

```
Function Lagrange(z,x,f)
% Returns an interpolated value at z
% x is an (n+1)-dimensional array of the given data points
% f is an (n+1)-dimensional array of the given function values

Interpolated_Value=0
for i=1:n+1
   Lagrangian=1
   for j=1:n+1
      if(j~=i)
      then Lagrangian=Lagrangian*(z-x(j))/(x(i)-x(j))
      end if
   end for j
   Interpolated_Value=Interpolated_Value+Lagrangian*f(i)
end for I
Return Interpolated_Value
```

Spline Interpolation

- There are cases where polynomials can lead to erroneous results because of round-off errors or significant distinction between an interpolating polynomial and an interpolated function
- Alternative approach is to apply lower-order polynomials to subsets of data points. Such connecting polynomials are called spline functions

Why Splines ?

$$f(x)=\frac{1}{1+25x^2}$$

Table : Six equidistantly spaced points in [-1, 1]

x	$y=\frac{1}{1+25x^2}$
-1.0	0.038461
-0.6	0.1
-0.2	0.5
0.2	0.5
0.6	0.1
1.0	0.038461

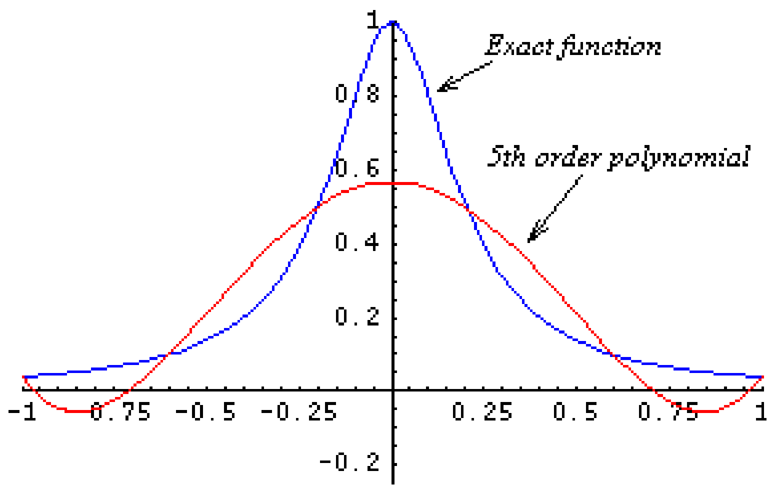


Figure : 5th order polynomial vs. exact function

Why Splines ?

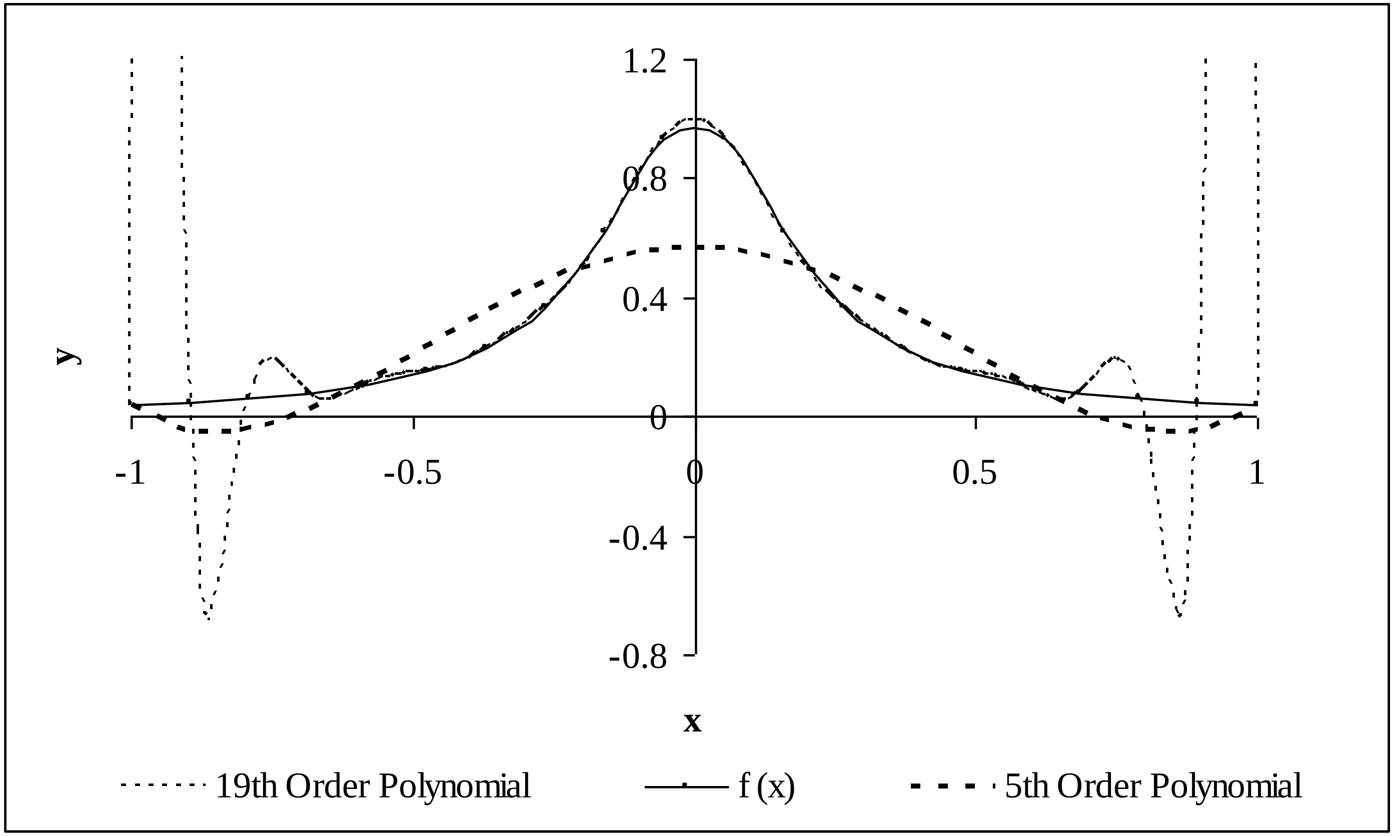
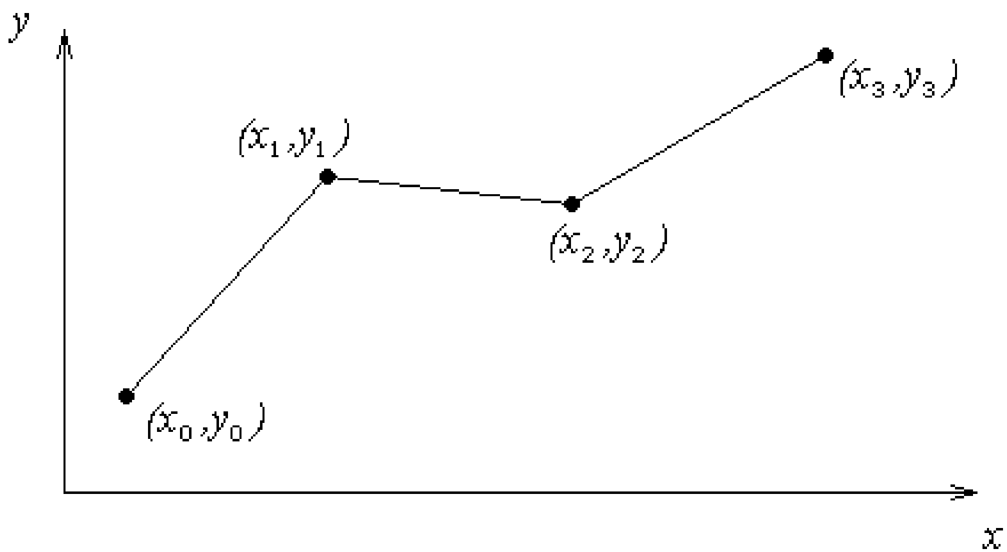


Figure : Higher order polynomial interpolation is a bad idea

Linear Spline Interpolation

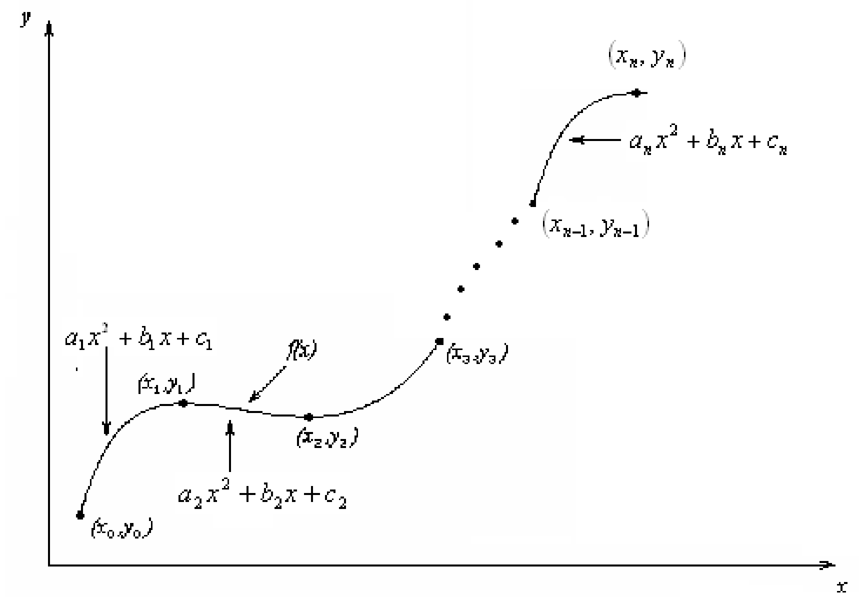
Given $(x_0, y_0), (x_1, y_1), \ldots\ldots, (x_{n-1}, y_{n-1})(x_n, y_n)$, fit linear splines to the data. This simply involves forming the consecutive data through straight lines. So if the above data is given in an ascending order, the linear splines are given by $(y_i = f(x_i))$

Figure : Linear splines



Quadratic Spline Interpolation

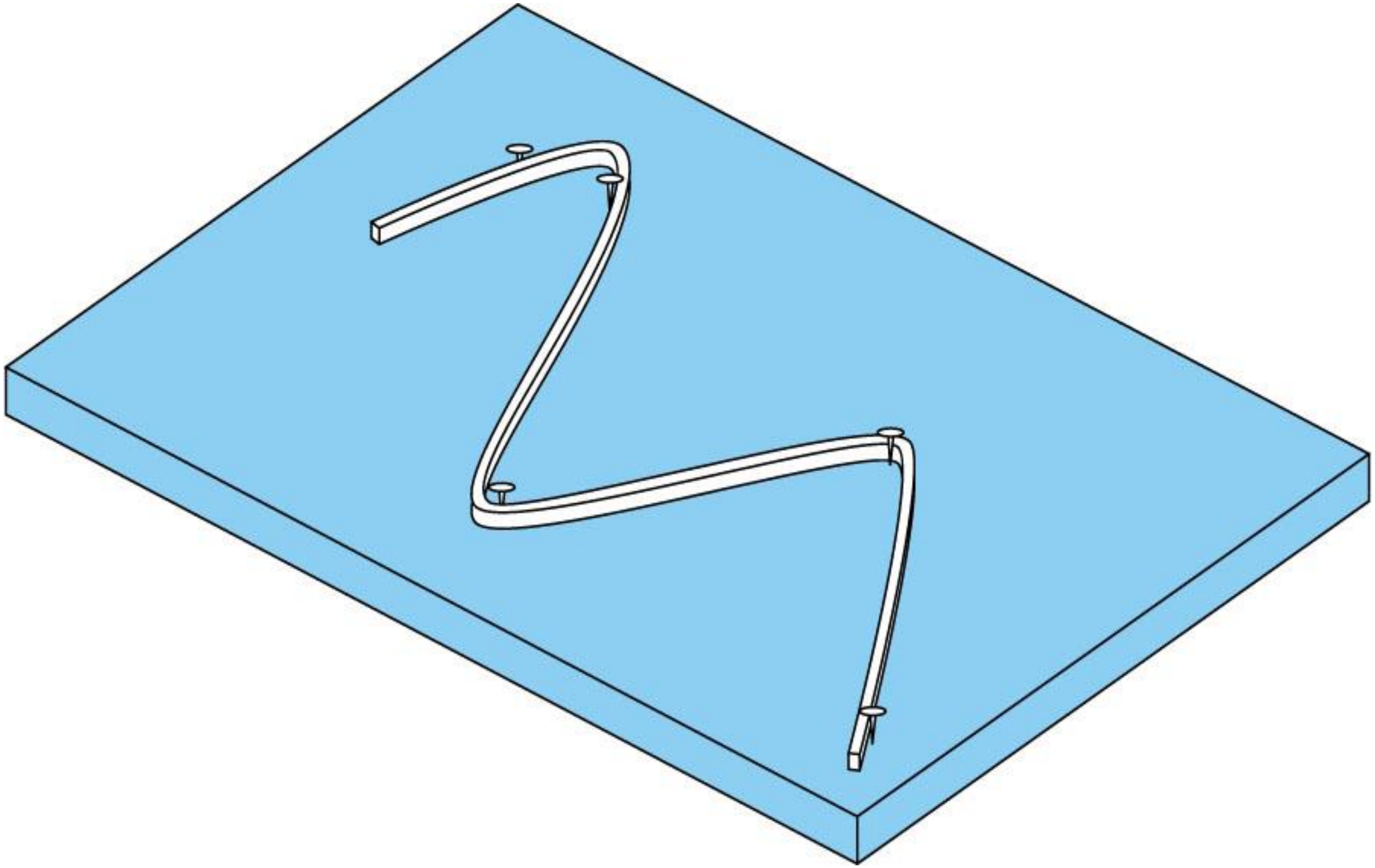
Given $(x_0, y_0),(x_1, y_1),......,(x_{n-1}, y_{n-1}),(x_n, y_n)$, fit quadratic splines through the data. The splines are given by



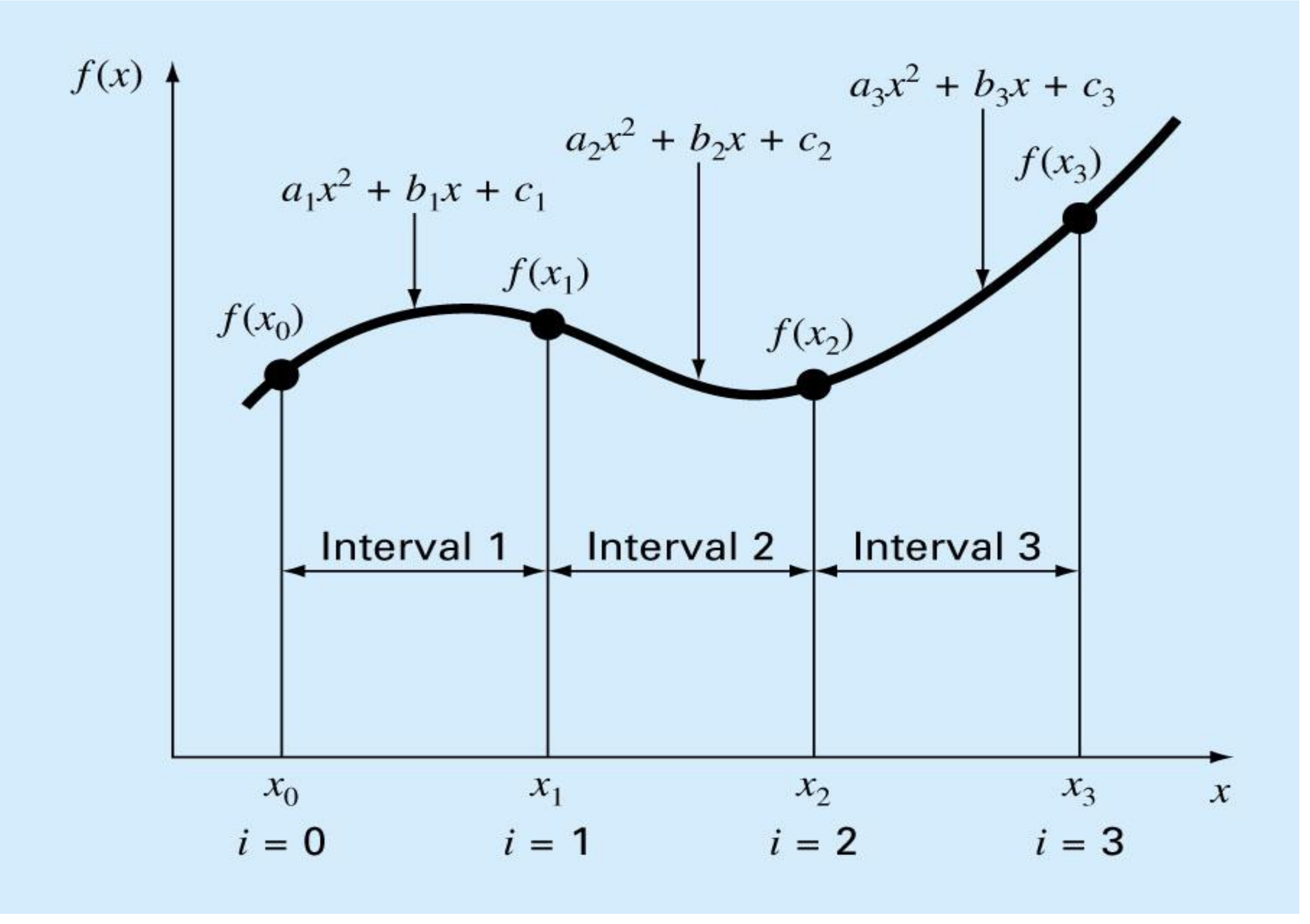
$$
\begin{aligned}
f(x) &= a_1 x^2 + b_1 x + c_1, & x_0 \le x \le x_1 \\
&= a_2 x^2 + b_2 x + c_2, & x_1 \le x \le x_2 \\
&\ \ . \\
&\ \ . \\
&\ \ . \\
&= a_n x^2 + b_n x + c_n, & x_{n-1} \le x \le x_n
\end{aligned}
$$

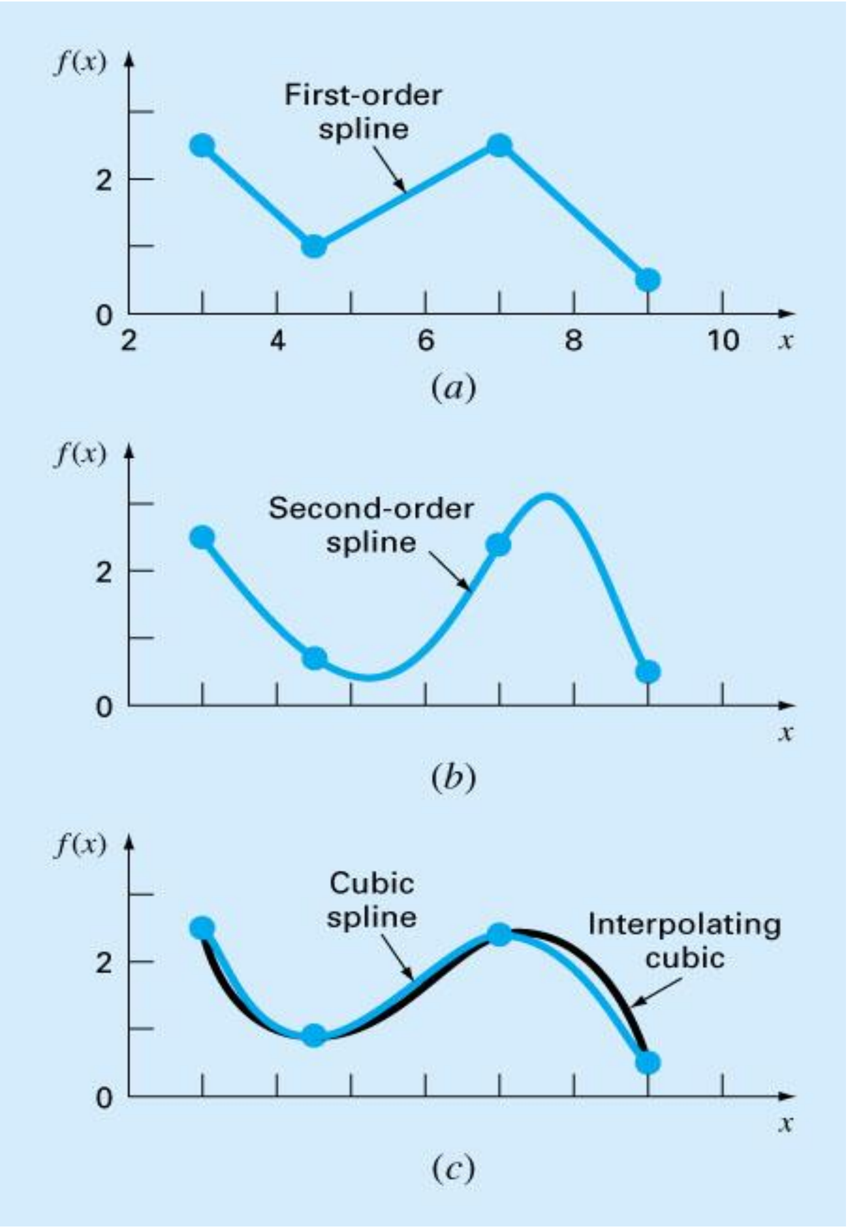
Find $a_i,\ b_i,\ c_i,\ i = 1, 2, \ldots, \text{n}$

Spline Interpolation



Spline Interpolation





f(x)
First-order
spline
2
0
2
4
6
8
10
x
(a)
f(x)
Second-order
spline
2
0
x
(b)
f(x)
Cubic
spline
Interpolating
cubic
2
0
x
(c)