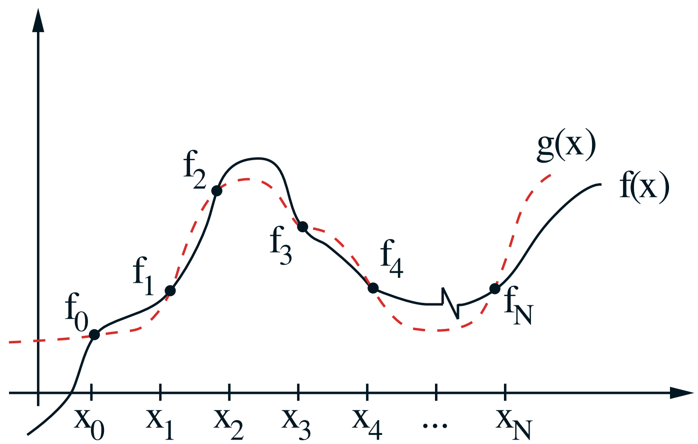
LECTURE 3

LAGRANGE INTERPOLATION

- ***Fit $N+1$ points with an N^{th} degree polynomial***



- $f(x)$ = exact function of which only $N+1$ discrete values are known and used to establish an interpolating or approximating function $g(x)$
- $g(x)$ = approximating or interpolating function. This function will pass through all specified $N+1$ ***interpolation points*** (also referred to as ***data points*** or ***nodes***).

- The ***interpolation points*** or ***nodes*** are given as:

$$\begin{array}{ll} x_o & f(x_o) \equiv f_o \\ x_1 & f(x_1) \equiv f_1 \\ x_2 & f(x_2) \equiv f_2 \\ \vdots & \\ x_N & f(x_N) \equiv f_N \end{array}$$

- There exists only one N^{th} degree polynomial that passes through a given set of $N+1$ points. It's form is (expressed as a power series):

$$g(x) = a_o + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_N x^N$$

where a_i = unknown coefficients, $i = 0, N$ ($N+1$ coefficients).

- No matter how we derive the N^{th} degree polynomial,
  - Fitting power series
  - Lagrange interpolating functions
  - Newton forward or backward interpolation

The resulting polynomial will always be the same!

Power Series Fitting to Define Lagrange Interpolation

- $g(x)$ must match $f(x)$ at the selected data points $\Rightarrow$

$$g(x_o) = f_o \quad \Rightarrow \quad a_o + a_1 x_o + a_2 x_o^2 + \ldots + a_N x_o^N = f_o$$

$$g(x_1) = f_1 \quad \Rightarrow \quad a_o + a_1 x_1 + a_2 x_1^2 + \ldots + a_N x_1^N = f_1$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$g(x_N) = f_N \quad \Rightarrow \quad a_o + a_1 x_N + a_2 x_N^2 + \ldots + a_N x_N^N = f_N$$

- Solve set of simultaneous equations

$$\begin{bmatrix} 1 & x_o & x_o^2 & \ldots & x_o^N \\ 1 & x_1 & x_1^2 & \ldots & x_1^N \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_N & x_N^2 & \ldots & x_N^N \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} f_o \\ f_1 \\ \vdots \\ f_N \end{bmatrix}$$

- It is relatively computationally costly to solve the coefficients of the interpolating function $g(x)$ (i.e. you need to program a solution to these equations).

Lagrange Interpolation Using Basis Functions

- We note that in general $g(x_i) = f_i$
- Let

$$g(x) = \sum_{i=0}^{N} f_i \, V_i(x)$$

where $V_i(x)$ ***= polynomial of degree N associated with each node i such that***

$$V_i(x_j) \equiv \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

- For example if we have 5 interpolation points (or nodes)

$$g(x_3) = f_o V_o(x_3) + f_1 V_1(x_3) + f_2 V_2(x_3) + f_3 V_3(x_3) + f_4 V_4(x_3)$$

Using the definition for $V_i(x_j)$: $V_0(x_3) = 0$; $V_1(x_3) = 0$; $V_2(x_3) = 0$; $V_3(x_3) = 1$; $V_4(x_3) = 0$,we have:

$$g(x_3) = f_3$$

- How do we construct $V_i(x)$?
  - Degree N
  - Roots at $x_o, x_1, x_2, \ldots x_{i-1}, x_{i+1}, \ldots, x_N$ (at all nodes except x_i)
  - $V_i(x_i) = 1$

- Let $W_i(x) = (x-x_o)(x-x_1)(x-x_2)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_N)$
  - The function W_i is such that we ***do*** have the required roots, i.e. it equals zero at nodes $x_o, x_1, x_2, \ldots, x_N$ except at node x_i
  - Degree of $W_i(x)$ is N
  - However $W_i(x)$ in the form presented will not equal to unity at x_i

- We normalize $W_i(x)$ and define the Lagrange basis functions $V_i(x)$

$$V_i(x) = \frac{(x-x_o)(x-x_1)(x-x_2)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_N)}{(x_i-x_o)(x_i-x_1)(x_i-x_2)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_N)}$$

- Now we have $V_i(x)$ such that $V_i(x_i)$ equals:

$$V_i(x_i) = \frac{(x_i - x_o)(x_i - x_1)(x_i - x_2)\ldots(x_i - x_{i-1})(1)(x_i - x_{i+1})\ldots(x_i - x_N)}{(x_i - x_o)(x_i - x_1)(x_i - x_2)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_N)} \quad \Rightarrow$$

$$V_i(x_i) = 1$$

- We also satisfy $V_i(x_j) = 0$ for $i \neq j$

$$\text{e.g.} \quad V_1(x_2) = \frac{(x_2 - x_o)(1)(x_2 - x_2) \cdot (x_2 - x_3)\ldots(x_2 - x_N)}{(x_1 - x_o)(1)(x_1 - x_2)(x_1 - x_3)\ldots(x_1 - x_N)} = 0$$

- The general form of the interpolating function $g(x)$ with the specified form of $V_i(x)$ is:

$$\boldsymbol{g(x) = \sum_{i=0}^{N} f_i V_i(x)}$$

  - The sum of polynomials of degree N is also polynomial of degree N
  - $g(x)$ is equivalent to fitting the power series and computing coefficients $a_o, \ldots, a_N$.

Lagrange Linear Interpolation Using Basis Functions

- Linear Lagrange $(N = 1)$ is the simplest form of Lagrange Interpolation

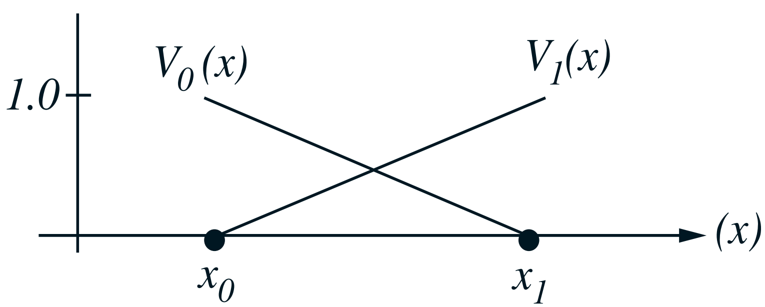
$$g(x) = \sum_{i=0}^{1} f_i V_i(x)$$

$$\Rightarrow$$

$$g(x) = f_o V_o(x) + f_1 V_1(x)$$

where

$$V_o(x) = \frac{(x - x_1)}{(x_o - x_1)} = \frac{(x_1 - x)}{(x_1 - x_o)} \qquad \text{and} \qquad V_1(x) = \frac{(x - x_o)}{(x_1 - x_o)}$$



Example

- Given the following data:

$$x_o = 2 \qquad f_o = 1.5$$

$$x_1 = 5 \qquad f_1 = 4.0$$

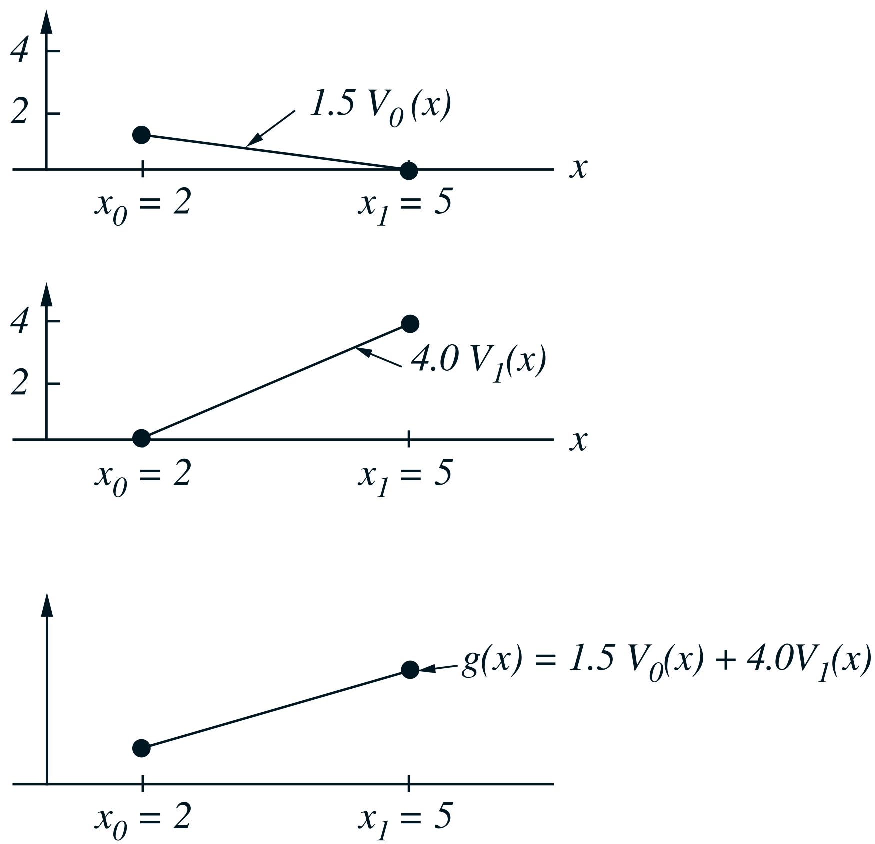
  Find the linear interpolating function $g(x)$

- Lagrange basis functions are:

$$V_o(x) = \frac{5-x}{3} \quad \text{and} \quad V_1(x) = \frac{x-2}{3}$$

- Interpolating function $g(x)$ is:

$$g(x) = 1.5V_o(x) + 4.0V_1(x)$$


4
2
$1.5\ V_0(x)$
x
$x_0 = 2$
$x_1 = 5$
4
2
$4.0\ V_1(x)$
x
$x_0 = 2$
$x_1 = 5$
$g(x) = 1.5\ V_0(x) + 4.0 V_1(x)$
$x_0 = 2$
$x_1 = 5$

Lagrange Quadratic Interpolation Using Basis Functions

- For quadratic Lagrange interpolation, N=2

$$g(x) = \sum_{i=0}^{2} f_i \; V_i(x)$$

$$\Rightarrow$$

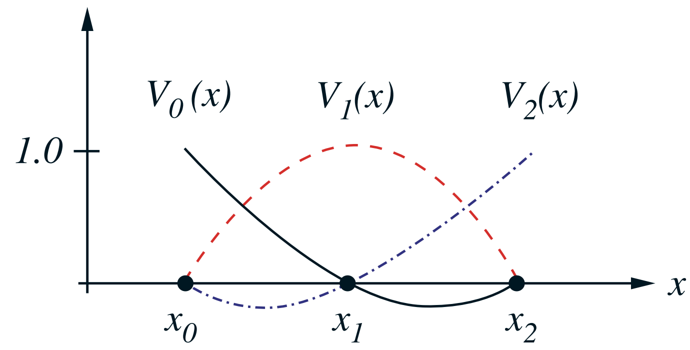
$$g(x) = f_o V_o(x) + f_1 V_1(x) + f_2 V_2(x)$$

where

$$V_o(x) = \frac{(x - x_1)(x - x_2)}{(x_o - x_1)(x_o - x_2)}$$

$$V_1(x) = \frac{(x - x_o)(x - x_2)}{(x_1 - x_o)(x_1 - x_2)}$$

$$V_2(x) = \frac{(x - x_o)(x - x_1)}{(x_2 - x_o)(x_2 - x_1)}$$



- Note that the location of the roots of $V_0(x)$, $V_1(x)$ and $V_2(x)$ are defined such that the basic premise of interpolation is satisfied, namely that $g(x_i) = f_i$. Thus:

$$g(x_o) = V_o(x_o)f_o + V_1(x_o)f_1 + V_2(x_o)f_2 = f_0$$

$$g(x_1) = V_o(x_1)f_o + V_1(x_1)f_1 + V_2(x_1)f_2 = f_1$$

$$g(x_2) = V_o(x_2)f_o + V_1(x_2)f_1 + V_2(x_2)f_2 = f_2$$

Example

- Given the following data:

$$x_o = 3 \qquad f_o = 1$$

$$x_1 = 4 \qquad f_1 = 2$$

$$x_2 = 5 \qquad f_2 = 4$$

Find the quadratic interpolating function $g(x)$

- Lagrange basis functions are

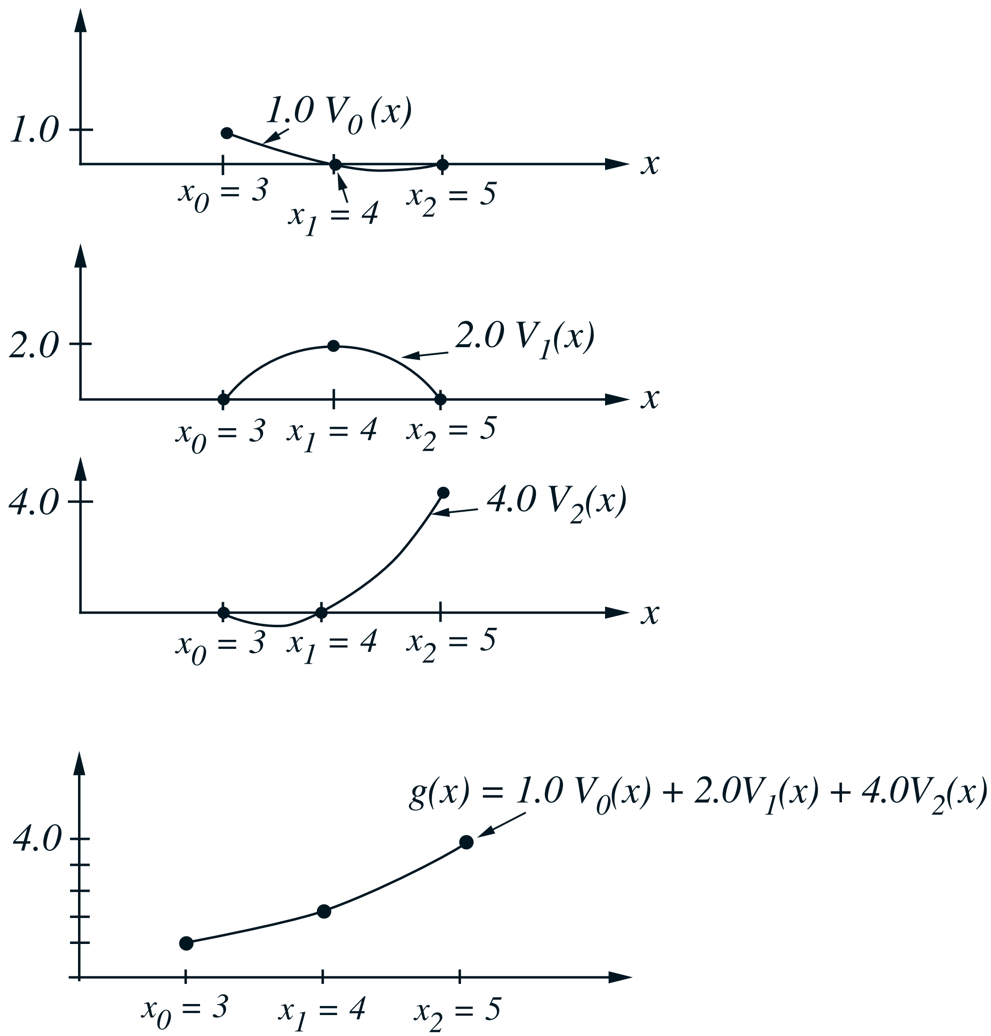
$$V_o(x) = \frac{(x-4)(x-5)}{(3-4)(3-5)}$$

$$V_1(x) = \frac{(x-3)(x-5)}{(4-3)(4-5)}$$

$$V_2(x) = \frac{(x-3)(x-4)}{(5-3)(5-4)}$$

- Interpolating function *g(x)* is:

$$g(x) = 1.0V_o(x) + 2.0V_1(x) + 4.0V_2(x)$$


1.0
$1.0\ V_0(x)$
x
$x_0 = 3$
$x_1 = 4$
$x_2 = 5$
2.0
$2.0\ V_1(x)$
x
$x_0 = 3$
$x_1 = 4$
$x_2 = 5$
4.0
$4.0\ V_2(x)$
x
$x_0 = 3$
$x_1 = 4$
$x_2 = 5$
4.0
$g(x) = 1.0\ V_0(x) + 2.0V_1(x) + 4.0V_2(x)$
$x_0 = 3$
$x_1 = 4$
$x_2 = 5$

Lagrange Cubic Interpolation Using Basis Functions

- For Cubic Lagrange interpolation, N=3

Example

- Consider the following table of functional values (generated with $f(x) = \ln x$)

i	x_i	f_i
0	0.40	-0.916291
1	0.50	-0.693147
2	0.70	-0.356675
3	0.80	-0.223144

- Find $g(0.60)$ as:

$$g(x) = f_o\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_o-x_1)(x_o-x_2)(x_o-x_3)} + f_1\frac{(x-x_o)(x-x_2)(x-x_3)}{(x_1-x_o)(x_1-x_2)(x_1-x_3)}$$

$$+ f_2\frac{(x-x_o)(x-x_1)(x-x_3)}{(x_2-x_o)(x_2-x_1)(x_2-x_3)} + f_3\frac{(x-x_o)(x-x_1)(x-x_2)}{(x_3-x_o)(x_3-x_1)(x_3-x_2)}$$

$$g(0.60) = -0.916291 \cdot \frac{(0.60-0.50)(0.60-0.70)(0.60-0.80)}{(0.40-0.50)(0.40-0.70)(0.40-0.80)}$$

$$-0.693147 \cdot \frac{(0.60-0.40)(0.60-0.70)(0.60-0.80)}{(0.50-0.40)(0.50-0.70)(0.50-0.80)}$$

$$-0.356675 \cdot \frac{(0.60-0.40)(0.60-0.50)(0.60-0.80)}{(0.70-0.40)(0.70-0.50)(0.70-0.80)}$$

$$-0.223144 \cdot \frac{(0.60-0.40)(0.60-0.50)(0.60-0.70)}{(0.80-0.40)(0.80-0.50)(0.80-0.70)}$$

$$\Rightarrow$$

$$\boldsymbol{g(0.60) = -0.509976}$$

Errors Associated with Lagrange Interpolation

- Using Taylor series analysis, the error can be shown to be given by:

$$e(x) = f(x) - g(x)$$

$$\boldsymbol{e(x) = L(x) f^{(N+1)}(\xi) \qquad x_o \le \xi \le x_N}$$

where

$$f^{N+1}(\xi) = N+1^{th} \text{ derivative of } f \text{ w.r.t. } x \text{ evaluated at } \xi$$

$$L(x) = \frac{(x-x_o)(x-x_1)\dots(x-x_N)}{(N+1)!} = \text{an } N+1^{th} \text{ degree polynomial}$$

- Notes
  - If $f(x)$ = polynomial of degree M where $M \le N$, then

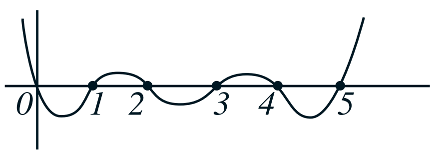
$$f^{(N+1)}(x) = 0 \quad \Rightarrow \quad e(x) = 0 \text{ } \textit{for all } x$$

Therefore $g(x)$ will be an exact representation of $f(x)$

- Since in general ξ is not known, if the interval $[x_o, x_N]$ is small and if $f^{(N+1)}(x)$ does not change rapidly in the interval

$$e(x) \approx L(x) f^{(N+1)}(x_m) \quad \text{where } x_m = \frac{x_o + x_N}{2}.$$

- $f^{(N+1)}$ can be estimated by using Finite Difference (F.D.) formulae
- $L(x)$ will significantly effect the distribution of the error
- $L(x)$ is a minimum at the center of $[x_o, x_N]$ and a maximum near the edges
  - e.g. using 6 point interpolation $L(x)$ looks like:



  - $L(x) = 0$ at all data points
  - $L(x)$ largest $0 \le x \le 1$ $4 \le x \le 5$. $L(x)$ becomes very large outside of the interval.

- As the size of the interpolating domain increases, so does the maximum error within the interval

$$D = x_N - x_o \uparrow \;\Rightarrow\; L_{max}\Big|_{x_0 \le x \le x_N} \uparrow \;\Rightarrow\; e_{max}\Big|_{x_0 \le x \le x_N} \uparrow$$

- As N increases from a small value, $L_{max}\Big|_{x_0 \le x \le x_N} \downarrow \;\Rightarrow\; e_{max}\Big|_{x_0 \le x \le x_N} \downarrow$

- However as $N > N_{CRIT} \Rightarrow L_{max}\Big|_{x_0 \le x \le x_N} \uparrow$ for a given $[x_o, x_N]$ and thus $e_{max}\Big|_{x_0 \le x \le x_N} \uparrow$
    - Therefore convergence as $N \uparrow$ does not necessarily occur!!

- Properties of $f^{(N+1)}(\xi)$ will also influence error as D and N vary

Example

- Estimate the error made in the previous example knowing that $f(x) = \ln(x)$ (usually we do not have this information).

$$e(x) \approx L(x) f^{(N+1)}(x_m)$$

$$\Rightarrow$$

$$e(x) \approx \frac{(x - x_o)(x - x_1)(x - x_2)(x - x_3)}{(3+1)!} \; f^{(3+1)}(x_m)$$

$$\Rightarrow$$

$$e(0.60) \approx \frac{(0.60 - 0.40)(0.60 - 0.50)(0.60 - 0.70)(0.60 - 0.80)}{(3+1)!} f^{(3+1)}(0.6)$$

$$\Rightarrow$$

$$e(0.60) = 0.000017 \;\; f^{(4)}(0.6)$$

- We estimate the fourth derivative of $f(x)$ using the analytical function itself

$$f(x) = \ln x \quad \Rightarrow$$

$$f^{(1)}(x) = x^{-1} \quad \Rightarrow$$

$$f^{(2)}(x) = -x^{-2} \quad \Rightarrow$$

$$f^{(3)}(x) = 2x^{-3} \quad \Rightarrow$$

$$f^{(4)}(x) = -6x^{-4} \quad \Rightarrow$$

$$f^{(4)}(0.6) = -46.29$$

- Therefore

$$\mathbf{e(0.60) = -0.00079}$$

- Exact error is computed as:

$$\mathbf{E(x) = \ln(0.60) - g(0.60) = -0.00085}$$

Therefore error estimate is excellent

- Typically we would ***also*** have to estimate $f^{(N+1)}(x_m)$ using a Finite Difference (F.D.) approximation (a discrete differentiation formula).

SUMMARY OF LECTURES 2 AND 3

- Linear interpolation passes a straight line through 2 data points.

- Power series $\rightarrow$ $N+1$ data points $\rightarrow$ N^{th} degree polynomial $\rightarrow$ find coefficients by solving a matrix

- Lagrange Interpolation passes an N^{th} degree polynomial through $N+1$ data points $\rightarrow$ Use specialized nodal functions to make finding $g(x)$ easier.

$$g(x) = \sum_{i=0}^{N} f_i V_i(x)$$

where

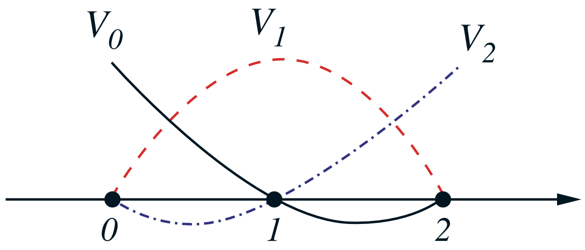
$g(x)$ = the interpolating function approximating *f(x)*

f_i = the value of the function at the data (or interpolation) point *i*

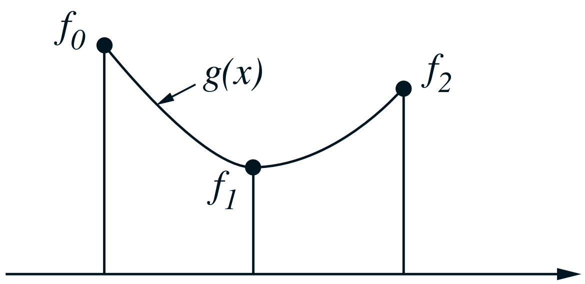
$V_i(x)$ = the Lagrange basis function

- Each Lagrange polynomial or basis function is set up such that it equals unity at the data point with which it is associated, zero at all other data points and nonzero in-between.

- For example when $N = 2 \rightarrow$ 3 data points



$$g(x) = f_o V_o(x) + f_1 V_1(x) + f_2 V_2(x)$$



- Linear interpolation is the same as Lagrange Interpolation with $N = 1$

- Error estimates can be derived but depend on knowing $f^{(N+1)}(x_m)$ (or at some point in the interval).

$$\boldsymbol{e(x) = L(x)f^{(N+1)}(\xi)} \qquad \boldsymbol{x_o \le \xi \le x_N}$$

where

$f^{N+1}(\xi) = N+1^{th}$ derivative of f w.r.t. x evaluated at ξ

$$L(x) = \frac{(x-x_o)(x-x_1)\dots(x-x_N)}{(N+1)!} = \text{an } N+1^{th} \text{ degree polynomial}$$