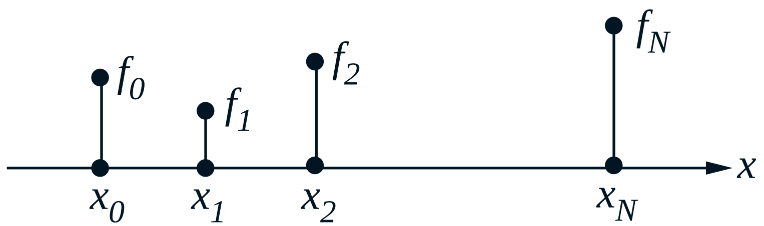
LECTURE 9

HERMITE INTERPOLATING POLYNOMIALS

- So far we have considered ***Lagrange Interpolation*** schemes which fit an N^{th} degree polynomial to $N+1$ data or interpolation points



- All these Lagrange Interpolation methods discussed had the general form:

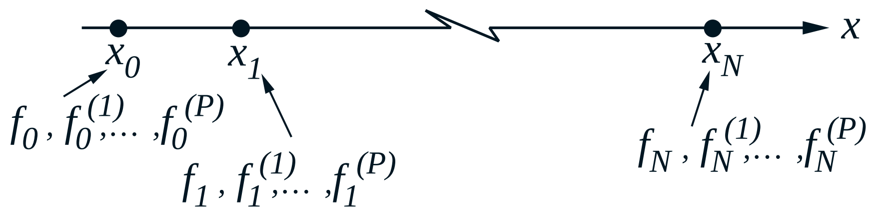
$$g(x) = \sum_{i=0}^{N} a_i x^i \quad \Rightarrow$$

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

- Fitting the data points meant requiring the interpolating polynomial to be equal to the functional values at the data points:

$$g(x_i) = f_i, \qquad i = 0, N$$

- ***Hermite Interpolation: Develop an interpolating polynomial which equals the function and its derivatives up to p^{th} order at $N+1$ data points.***



- Therefore we require that

$$g(x_i) = f_i \qquad i = 0, N \qquad (N+1) \text{ constraints}$$

$$g^{(1)}(x_i) = f_i^{(1)} \qquad i = 0, N \qquad (N+1) \text{ constraints}$$

$$\vdots$$

$$g^{(p)}(x_i) = f_i^{(p)} \qquad i = 0, N \qquad (N+1) \text{ constraints}$$

- We have a total of $(p+1)\ (N+1)$ constraints

- **We need to set up a general polynomial which is of degree $(p+1)(N+1)-1$ (number of constraints must equal the number of unknowns in the interpolating polynomial).**

- Setting up a polynomial with a total of $(p+1)\ (N+1)$ unknowns:

$$g(x) = \sum_{i=0}^{(p+1)(N+1)-1} a_i x^i$$

- Procedure to develop Hermite interpolation:
  - Set up the interpolating polynomial
  - Implement constraints
  - Solve for unknown coefficients, $a_i, \quad i = 0, (p+1)(N+1)-1$

- Note that Lagrange interpolation is a special case of Hermite interpolation ($p = 0$, i.e. no derivatives are matched).

- It is also possible to set up specialized Hermite interpolation functions which do not include all functional and/or derivative values at all nodes
  - There may be some missing functional or derivative values at certain nodes
  - This lowers the degree of the interpolating function.

Cubic Hermite Interpolation

- Develop a two data point Hermite interpolation function which passes through the function and its first derivative for the interval $[0, 1]$.

	x	f	$f^{(1)}$
x_o	0	f_o	$f_o^{(1)}$
x_1	+1	f_1	$f_1^{(1)}$

- Therefore $p = 1$ and $N + 1 = 2$.
- We must impose $(1 + 1)(2) = 4$ constraint equations (match function and its derivative at two data points).
- Therefore we require a 3rd degree polynomial.

$$g(x) = a_o + a_1 x + a_2 x^2 + a_3 x^3$$

$$g^{(1)}(x) = a_1 + 2a_2 x + 3a_3 x^2$$

- Application of constraints

$$g(0) = f_o \quad \Rightarrow \quad a_o = f_o$$

$$g(1) = f_1 \quad \Rightarrow \quad a_o + a_1 + a_2 + a_3 = f_1$$

$$g^{(1)}(0) = f_o^{(1)} \quad \Rightarrow \quad a_1 = f_o^{(1)}$$

$$g^{(1)}(1) = f_1^{(1)} \quad \Rightarrow \quad a_1 + 2a_2 + 3a_3 = f_1^{(1)}$$

- Constraint equations may be written in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f_o \\ f_1 \\ f_o^{(1)} \\ f_1^{(1)} \end{bmatrix}$$

- Solve for a_o, a_1, a_2, a_3

$$a_o = f_o$$

$$a_1 = f_o^{(1)}$$

$$a_2 = 3f_1 - 3f_o - f_1^{(1)} - 2f_o^{(1)}$$

$$a_3 = -2f_1 + 2f_o + f_o^{(1)} + f_1^{(1)}$$

- Therefore

$$\mathbf{g(x) = f_o + f_o^{(1)}x + (3f_1 - 3f_o - f_1^{(1)} - 2f_o^{(1)})x^2 + (-2f_1 + 2f_o + f_o^{(1)} + f_1^{(1)})x^3}$$

- Checking $g(x)$ to ensure that the constraints are satisfied:

$$g(0) = f_o$$

$$g(1) = f_1$$

$$g^{(1)}(0) = f_o^{(1)}$$

$$g^{(1)}(1) = f_1^{(1)}$$

- We note that $g(x)$ can be re-written such that the functional and derivative values are factored out:

$$g(x) = f_o(2x^3 - 3x^2 + 1) + f_1(-2x^3 + 3x^2) + f_o^{(1)}(x^3 - 2x^2 + x) + f_1^{(1)}(x^3 - x^2)$$

- $g(x)$ can be expressed in generic form as:

$$g(x) = f_o\alpha_o(x) + f_1\alpha_1(x) + f_o^{(1)}\beta_o(x) + f_1^{(1)}\beta_1(x)$$

- Each basis function is a ***third*** degree polynomial

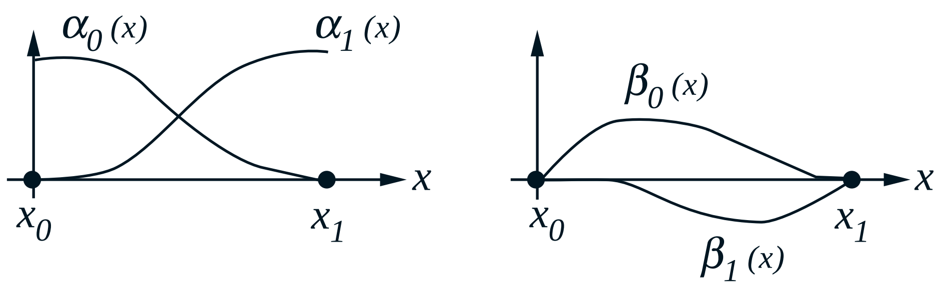
$\alpha_o(x) \equiv 2x^3 - 3x^2 + 1$ associated with the function at data point x_o

$\alpha_1(x) \equiv -2x^3 + 3x^2$ associated with the function at data point x_1

$\beta_o(x) \equiv x^3 - 2x^2 + x$ associated with the first derivative at data point x_o

$\beta_1(x) \equiv x^3 - x^2$ associated with the first derivative at data point x_1

- The cubic Hermite basis functions vary with x as:



- Therefore we can define 2 separate functions associated with each data point. Each is a ***third*** degree polynomial.

- *NOW WE NEED 2 NODES × 2 FUNCTIONS PER NODE × 4 DEGREES OF FREEDOM PER FUNCTION = 16 CONSTRAINTS.*

- Each of these functions satisfies the following constraints

- 8 constraints on the functions themselves for $g(x)$ to match the specified functional values

$$g(x) = f_o\alpha_o(x) + f_1\alpha_1(x) + f_o^{(1)}\beta_o(x) + f_1^{(1)}\beta_1(x)$$

x	$\alpha_o(x)$	$\alpha_1(x)$	$\beta_o(x)$	$\beta_1(x)$
$x_o = 0$	1	0	0	0
$x_1 = 1$	0	1	0	0

- 8 constraints on the derivatives of the functions for $g^{(1)}(x)$ to match the specified derivative values

$$g^{(1)}(x) = f_o\alpha_o^{(1)}(x) + f_1\alpha_1^{(1)}(x) + f_o^{(1)}\beta_o^{(1)}(x) + f_1^{(1)}\beta_1^{(1)}(x)$$

x	$\alpha_o^{(1)}(x)$	$\alpha_1^{(1)}(x)$	$\beta_o^{(1)}(x)$	$\beta_1^{(1)}(x)$
$x_o = 0$	0	0	1	0
$x_1 = 1$	0	0	0	1

- Mathematically these 16 constraints can be expressed as

$$\alpha_i(x_j) = \delta_{ij} \qquad \beta_i(x_j) = 0 \qquad i, j = 0, 1$$

$$\alpha_i^{(1)}(x_j) = 0 \qquad \beta_i^{(1)}(x_j) = \delta_{ij} \qquad i, j = 0, 1$$

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \text{ Kronecker Delta}$$

- ***Therefore an alternative method for setting up $g(x)$ is to associate a basis function with each functional value and the various derivative values at each data point.***
  - Each of these basis functions is a polynomial of degree $(p+1)(N+1)-1$.
  - There will be $(p+1)(N+1)$ basis functions.
  - We must set up $[(p+1)(N+1)]^2$ constraints.

General Hermite Interpolation Using Basis Functions

- In general, Hermite interpolation can be set up as:

$$g(x) = \sum_{i=0}^{N} \alpha_i(x) f_i + \sum_{i=0}^{N} \beta_i(x) f_i^{(1)} + \dots + \sum_{i=0}^{N} \theta_i(x) f_i^{(p)}$$

- We must satisfy the constraints:

$$g(x_j) = f_j \qquad j = 0, N$$

$$g^{(1)}(x_j) = f_j^{(1)} \qquad j = 0, N$$

$$\vdots$$

$$g^{(p)}(x_j) = f_j^{(p)} \qquad j = 0, N$$

- In order to satisfy

$$g(x_j) = f_j$$

$$\Rightarrow$$

$$\sum_{i=0}^{N} \alpha_i(x_j) f_i + \sum_{i=0}^{N} \beta_i(x_j) f_i^{(1)} + \ldots + \sum_{i=0}^{N} \theta_i(x_j) f_i^{(p)} = f_j$$

- We require the following constraints:

$$\alpha_i(x_j) = \delta_{ij} = \begin{cases} 0 & \mathrm{i} \neq j \\ 1 & \mathrm{i} = j \end{cases} \qquad i, j = 0, N$$

$$\beta_i(x_j) = 0 \qquad i, j = 0, N$$

$$\vdots$$

$$\theta_i(x_j) = 0 \qquad i, j = 0, N$$

- In order to satisfy:

$$g^{(1)}(x_j) = f_j^{(1)} \qquad j = 0, N$$

$$\Rightarrow$$

$$\sum_{i=0}^{N} \alpha_i^{(1)}(x_j) f_i + \sum_{i=0}^{N} \beta_i^{(1)}(x_j) f_i^{(1)} + \ldots + \sum_{i=0}^{N} \theta_i^{(1)}(x_j) f_i^{(p)} = f_j^{(1)}$$

- We require the following constraints:

$$\alpha_i^{(1)}(x_j) = 0 \qquad i, j = 0, N$$

$$\beta_i^{(1)}(x_j) = \delta_{ij} \qquad i, j = 0, N$$

$$\vdots$$

$$\theta_i^{(1)}(x_j) = 0 \qquad i, j = 0, N$$

- In order to satisfy the p^{th} derivative conditions:

$$g^{(p)}(x_j) = f_j^{(p)} \qquad j = 0, N$$

$$\Rightarrow$$

$$\sum_{i=0}^{N} \alpha_i^{(p)}(x_j) f_i + \sum_{i=0}^{N} \beta_i^{(p)}(x_j) f_i^{(1)} + \ldots + \sum_{i=0}^{N} \theta_i^{(p)}(x_j) f_i^{(p)} = f_j^{(p)}$$

- We require the following constraints:

$$\alpha_i^{(p)}(x_j) = 0 \qquad i, j = 0, N$$

$$\beta_i^{(p)}(x_j) = 0 \qquad i, j = 0, N$$

$$\vdots$$

$$\theta_i^{(p)}(x_j) = \delta_{ij} \qquad i, j = 0, N$$

- Each set of basis functions has the general form

$$\alpha_i(x) = \sum_{j=0}^{(p+1)(N+1)-1} a_{ij}x^j \qquad i = 0, N$$

$$\beta_i(x) = \sum_{j=0}^{(p+1)(N+1)-1} b_{ij}x^j \qquad i = 0, N$$

$$\vdots$$

$$\theta_i(x) = \sum_{j=0}^{(p+1)(N+1)-1} t_{ij}x^j \qquad i = 0, N$$

EXTRAPOLATION

- Use an interpolating function outside of the range within which the data points lie
- Extrapolation must always be used with caution
  - How was interpolation established?
  - What is the behavior of the function?
  - How far away from the interval is the point extrapolated?

SUMMARY OF LECTURES 8 AND 9

- Normalized Chebyshev polynomials are polynomial functions whose maximum amplitude is minimized over a given interval.

- If we select the roots of the $(N+1)^{th}$ degree Chebyshev polynomial as data (or interpolation) points for a N^{th} degree polynomial interpolation formula (e.g. Lagrange), we will have minimized the maximum error over the interval as far as we can.

- If Chebyshev roots are used as data points:

$$e^c(x) = \frac{1}{(N+1)}\psi_{N+1} f^{(N+1)}(\xi)$$

  - The polynomial terms in the error expression actually equal the $N+1^{th}$ degree Chebyshev polynomial and its maximum value on the interval is a minimum as compared to using any other set of interpolation points.

  - N^{th} degree polynomial interpolation error is also more evenly distributed when Chebyshev roots are used as data points as opposed to using evenly spaced points since in the latter case the error is increased near the ends of the interval and decreased within the interval.

- Hermite interpolation passes through the function and its first p derivatives at $N + 1$ data points. This results in a polynomial function of degree $(p + 1)(N + 1) - 1$.

- Extrapolation is the use of an interpolating formula for locations which do not lie within the interval.