CS 3331 Numerical Methods

Lecture 7: Interpolation

Cherung Lee

Outline

- Polynomial interpolation
  - Lagrange form
  - Newton form (divided-difference)
- Piecewise polynomial interpolation (spline method)
  - Linear and quadratic spline
  - Cubic spline
- Hermite interpolation
- Bezier curve

Polynomial Interpolation

Basis of polynomial interpolation

- Given $n+1$ points (x_i, y_i), there exists a polynomial p of degree n such that $p(x_i) = y_i$ for $i = 1, \cdots, n+1$.
- Assume $p(x) = a_n x^n + \cdots + a_1 x + a_0$. The goal is to find $a_n, \cdots, a_1, a_0$, which is equivalent to solving the linear system

$$\begin{pmatrix} x_1^n & \cdots & x_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n^n & \cdots & x_n & 1 \\ x_{n+1}^n & \cdots & x_{n+1} & 1 \end{pmatrix} \begin{pmatrix} a_n \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \end{pmatrix}$$

  - The matrix is called *Vandermonde matrix*.
  - This linear system can be solved in $O(n^2)$.

Lagrange form LVF pp.290-295

- A line goes through (x_1, y_1), (x_2, y_2) is

$$p(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$$

- A curve goes through (x_1, y_1), (x_2, y_2), and (x_3, y_3) is

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} y_3$$

- General form that goes through $(x_1, y_1), \ldots, (x_n, y_n)$

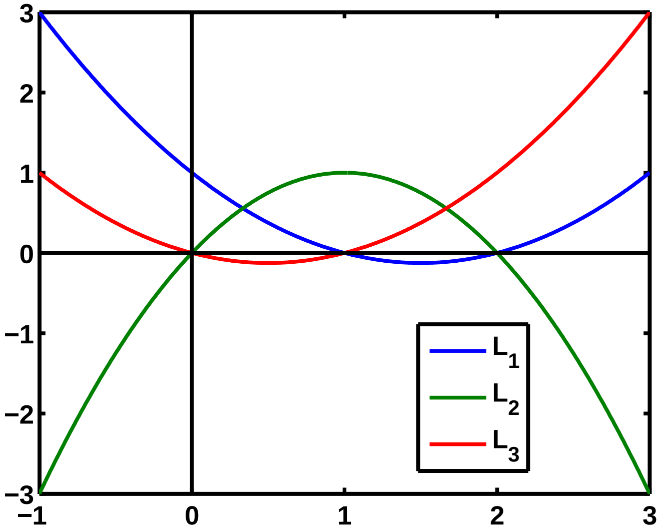
$$p(x) = \sum_{i=1}^{n} \frac{(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} y_i$$

Basis polynomials LVF pp.295

- $p(x)$ is a linear combination of basis polynomials

$$p(x) = L_1 y_1 + L_2 y_2 + \cdots + L_n y_n$$

where $L_i = \dfrac{(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$



- Basis polynomials for $x = 0, 1, 2.$

- $L_i(x) = \begin{cases} 0, & x = x_j,\ j \neq i; \\ 1, & x = x_i. \end{cases}$

Newton form LVF pp.296

- A line goes through (x_1, y_1), (x_2, y_2) is $p(x) = a_1 + a_2(x - x_1)$
  - $p(x_1) = y_1 \Rightarrow a_1 = y_1$
  - $p(x_2) = y_2 \Rightarrow a_2 = \dfrac{y_2 - y_1}{x_2 - x_1}$
- A curve goes through (x_1, y_1), (x_2, y_2), and (x_3, y_3) is

$$p(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2)$$

  - a_1 and a_2 are the same as above.
  - $p(x_3) = a_1 + a_2(x_3 - x_1) + a_3(x_3 - x_1)(x_3 - x_2) = y_3$

$$a_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1}$$

Divided differences LVF pp.296,300-301

- *Divided differences* is defined as
  - $Dy_i = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}$
  - $D^2 y_i = \dfrac{Dy_{i+1} - Dy_i}{x_{i+2} - x_i}$
  - $D^3 y_i = \dfrac{D^2 y_{i+1} - D^2 y_i}{x_{i+3} - x_i}$

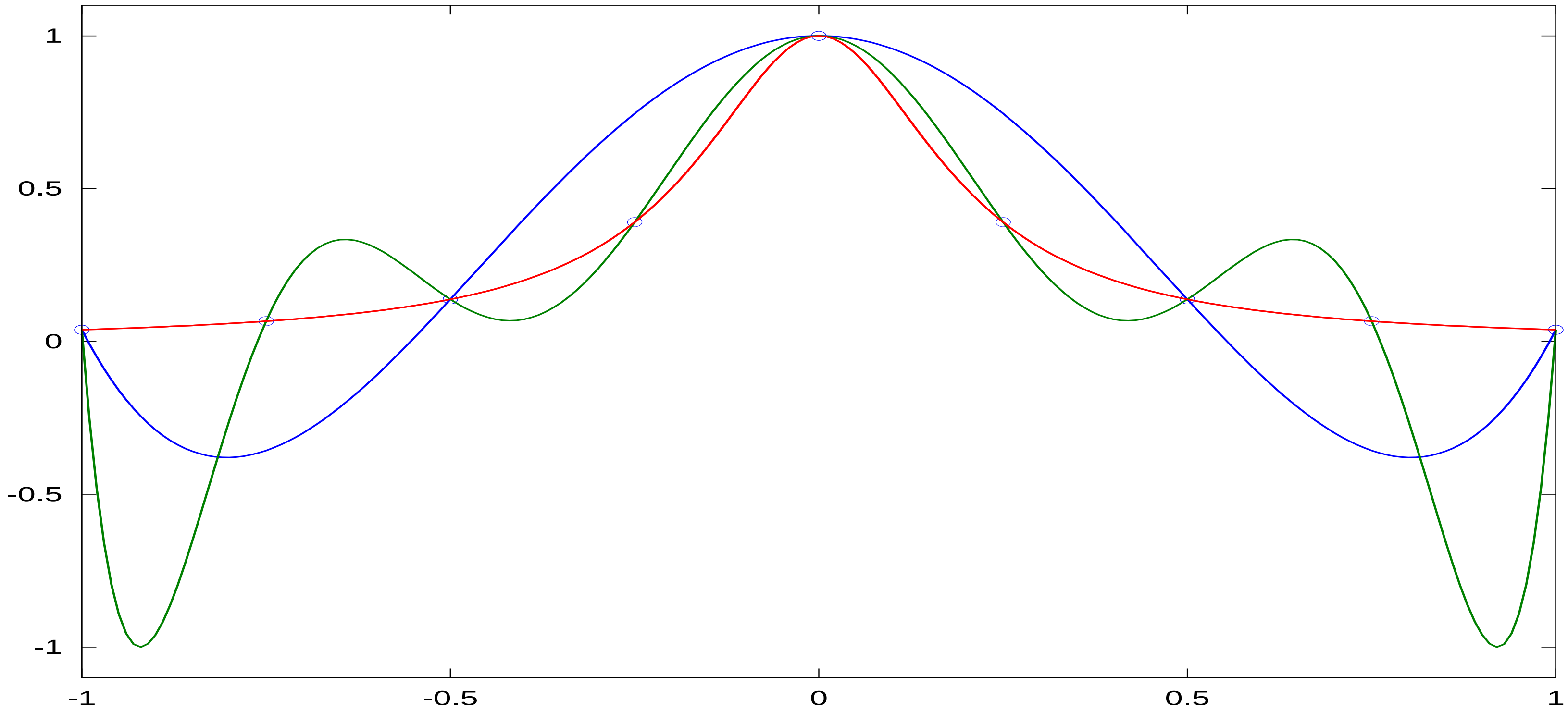
$$\begin{array}{lllll} x_1 & & & & \\ & Dy_1 & & & \\ x_2 & & D^2y_1 & & \\ & Dy_2 & & D^3y_1 & \\ x_3 & & D^2y_2 & & D^4y \\ & Dy_3 & & D^3y_2 & \\ x_4 & & D^2y_3 & & \\ & Dy_4 & & & \\ x_5 & & & & \end{array}$$

- Notation

$$\begin{array}{ll} x_1 & f[x_1] \\ x_2 & f[x_2] \\ x_3 & f[x_3] \end{array} \quad \begin{array}{l} f[x_1;x_2] = \frac{f[x_2]-f[x_1]}{x_2-x_1} \\ f[x_2;x_3] = \frac{f[x_3]-f[x_2]}{x_3-x_2} \end{array} \quad f[x_1;x_2;x_3] = \frac{f[x_2;x_3] - f[x_1;x_2]}{x_3 - x_1}$$

Problems of polynomial interpolation LVF pp.304

- Runge function: $f(x) = \frac{1}{1+25x^2}$ in the interval $[-1, 1]$.



Piecewise Polynomial Interpolation

Basis of piecewise interpolation LVF pp.313

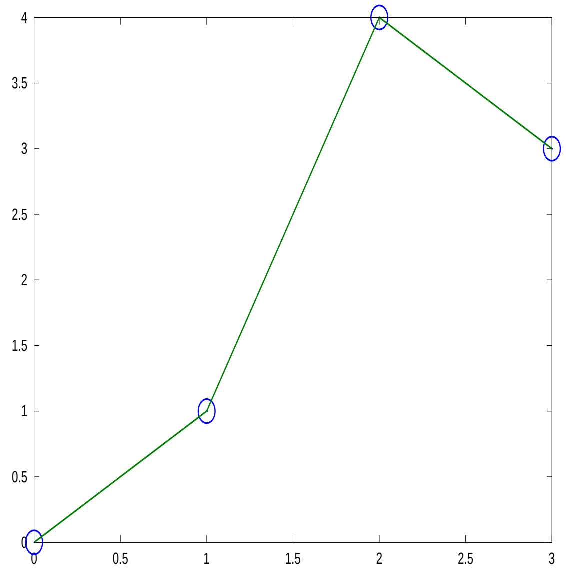
- Stitch consecutive low degree polynomials to interpolate $(x_1, y_1), \cdots, (x_n, y_n), (x_{n+1}, y_{n+1})$. (Assume $x_1 < \cdots < x_{n+1}$.)

$$\begin{array}{rclr} S_1(x) & = & a_{1,k}x^k + \cdots + a_{1,1}x + a_{1,0}, & x_1 \le x < x_2; \\ & \vdots & & \vdots \\ S_{n-1}(x) & = & a_{n-1,k}x^k + \cdots + a_{n-1,1}x + a_{n-1,0}, & x_{n-1} \le x < x_n; \\ S_n(x) & = & a_{n,k}x^k + \cdots + a_{n,1}x + a_{n,0}, & x_n \le x < x_{n+1}. \end{array}$$

- Number of unknowns for degree $k-1$ polynomial: kn
  - Continuous: $s_i(x_i) = y_i; s_i(x_{i+1}) = y_{i+1} \Rightarrow 2n$ eqns.
  - 1st differentiable: $s_i' = s_{i+1}'$ at some points.
  - 2nd differentiable: $s_i'' = s_{i+1}''$ at some points.

Piecewise linear interpolation LVF pp.312

- The piecewise linear interpolation of $(0,0),(1,1),(2,4),(3,3)$



$$p(x) = \begin{cases} y_1 + \dfrac{y_2 - y_1}{x_2 - x_1}(x - x_1), & x_1 \le x < x_2; \\[2ex] y_2 + \dfrac{y_3 - y_2}{x_3 - x_2}(x - x_2), & x_2 \le x < x_3; \\[2ex] y_3 + \dfrac{y_4 - y_3}{x_4 - x_3}(x - x_3), & x_3 \le x < x_4. \end{cases}$$

LVF has a different formula based on Lagrange form.

Piecewise quadratic interpolation (I) LVF pp.313

- For $i = 1, \ldots, n$, in each $[x_i, x_{i+1}]$, define

$$S_i(x) = y_i + a_i(x - x_i) + \frac{a_{i+1} - a_i}{2(x_{i+1} - x_i)}(x - x_i)^2$$

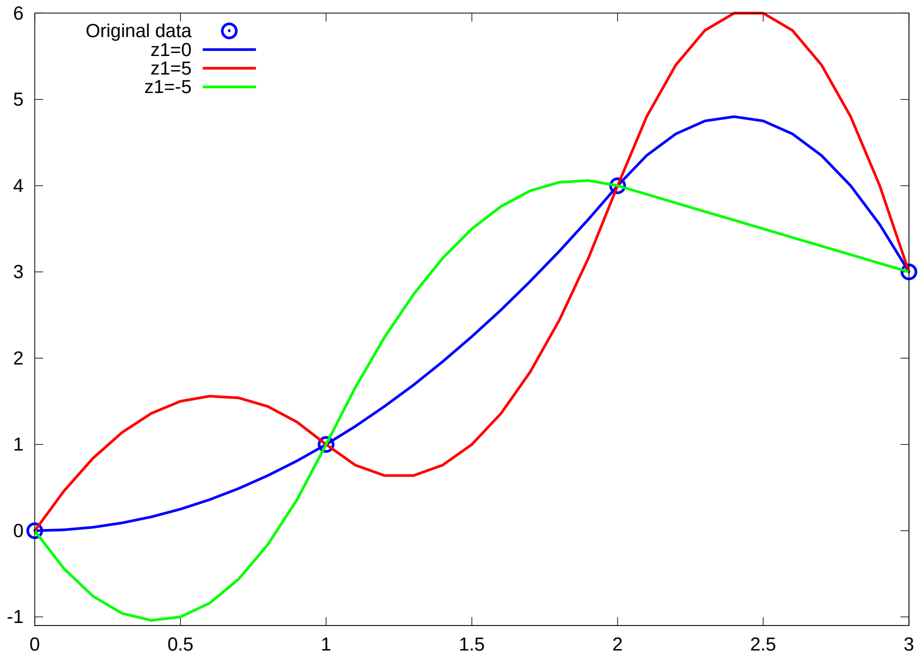
  where a_i are unknowns to be determined. ($n + 1$ unknowns)

  - The first derivation is $S_i'(x) = a_i + \frac{a_{i+1} - a_i}{x_{i+1} - x_i}(x - x_i)$.
    * $S_i'(x_i) = a_i$ and $S_i'(x_{i+1}) = a_{i+1}$.
  - Equations $S_i(x_i) = y_i$ are enforced. Another n equations

$$S_i(x_{i+1}) = y_{i+1} = y_i + a_i(x_{i+1} - x_i) + \frac{a_{i+1} - a_i}{2(x_{i+1} - x_i)}(x_{i+1} - x_i)^2$$

$$\Rightarrow a_{i+1} = 2\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - a_i$$

- Choose a_1, then other a_i can be computed sequentially.
  - Problem: the curve changes hugely with different a_1.



- Original data points $(0,0),(1,1),(2,4),(3,3)$

- Red line: $a_1 = 5$
- Blue line: $a_1 = 0$
- Green line: $a_1 = -5$

Piecewise quadratic interpolation (II) LVF pp.314

- Knots: the x-values that divide the interval.
  - In method (I), knots= data points.
- Method (II) defines knots= $\{z_i | i = 1, \ldots, n+1\}$ for n points.

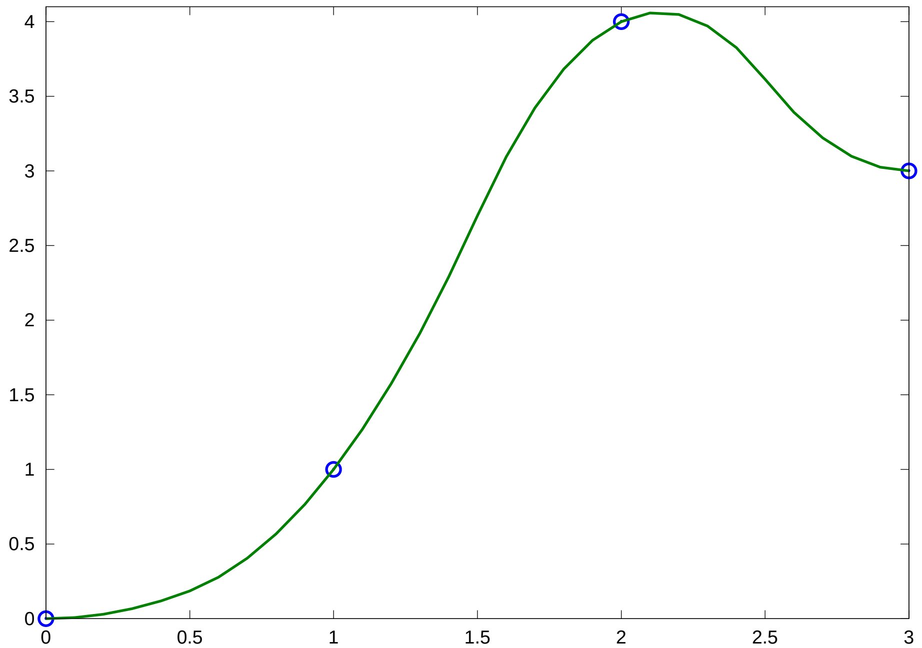
$$\begin{cases} z_1 & = & x_1, \\ z_i & = & (x_i + x_{i+1})/2, \quad \text{for } i = 1, n; \\ z_{n+1} & = & x_n, \end{cases}$$

  - For $n+1$ knots, one can define n quadratic polynomials.

$$S_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + y_i$$

  - $2n$ unknowns.

- $S_i(z_{i+1}) = S_{i+1}(z_{i+1})$ for $i = 1 : n - 1$ gives $n - 1$ eqns.
- $S'_i(z_{i+1}) = S'_{i+1}(z_{i+1})$ for $i = 1 : n - 1$ gives $n - 1$ eqns.
- Another 2 equations can be obtained by setting $S'_1(x_1) = 0$ and $S'_n(x_n) = 0$.



- Original points $(0, 0), (1, 1), (2, 4), (3, 3)$
- Problem: the linear equation has no structure: $O(n^3)$ algorithm

Piecewise cubic interpolation LVF pp.317,322-323

- For $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, define $h_i = x_{i+1} - x_i$. On each interval $[x_i, x_{i+1}]$, the cubic interpolation has the form

$$P_i(x) = a_i \frac{(x_{i+1} - x)^3}{h_i} + a_{i+1} \frac{(x - x_i)^3}{h_i} + b_i(x_{i+1} - x) + c_i(x - x_i).$$

- Its derivatives are

$$P_i'(x) = -3a_i \frac{(x_{i+1} - x)^2}{h_i} + 3a_{i+1} \frac{(x - x_i)^2}{h_i} - b_i + c_i,$$

$$P_i''(x) = 6a_i \frac{(x_{i+1} - x)}{h_i} + 6a_{i+1} \frac{(x - x_i)}{h_i}.$$

  - $P_i''(x_{i+1}) = 6a_{i+1} = P_{i+1}''(x_{i+1})$.

- There are $3n-2$ unknowns $(a_1, \ldots, a_n, b_1, \ldots, b_{n-1}, c_1, \ldots, c_{n-1})$.
- The continuity gives $2(n-1)$ equations:

$$\begin{cases} P_i(x_i) = y_i & \Rightarrow \quad a_i h_i^2 + b_i h_i = y_i & \Rightarrow \quad b_i = \frac{y_i}{h_i} - a_i h_i \\ P_i(x_{i+1}) = y_{i+1} & \Rightarrow \quad a_{i+1} h_i^2 + c_i h_i = y_{i+1} & \Rightarrow \quad c_i = \frac{y_{i+1}}{h_i} - a_{i+1} h_i \end{cases}$$

- The first differentiable gives $(n-2)$ eqns. $P_i'(x_{i+1}) = P_{i+1}'(x_{i+1})$

$$3h_i a_{i+1} - b_i + c_i = -3h_{i+1} a_{i+1} - b_{i+1} + c_{i+1}.$$

- Substitute b_i and c_i

$$h_i a_i + 2h_i a_{i+1} + \frac{y_{i+1} - y_i}{h_i} = -2h_{i+1} a_{i+1} - h_{i+1} a_{i+2} + \frac{y_{i+2} - y_{i+1}}{h_{i+1}}.$$

$$h_i a_i + 2(h_i + h_{i+1}) a_{i+1} + h_{i+1} a_{i+2} = \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}$$

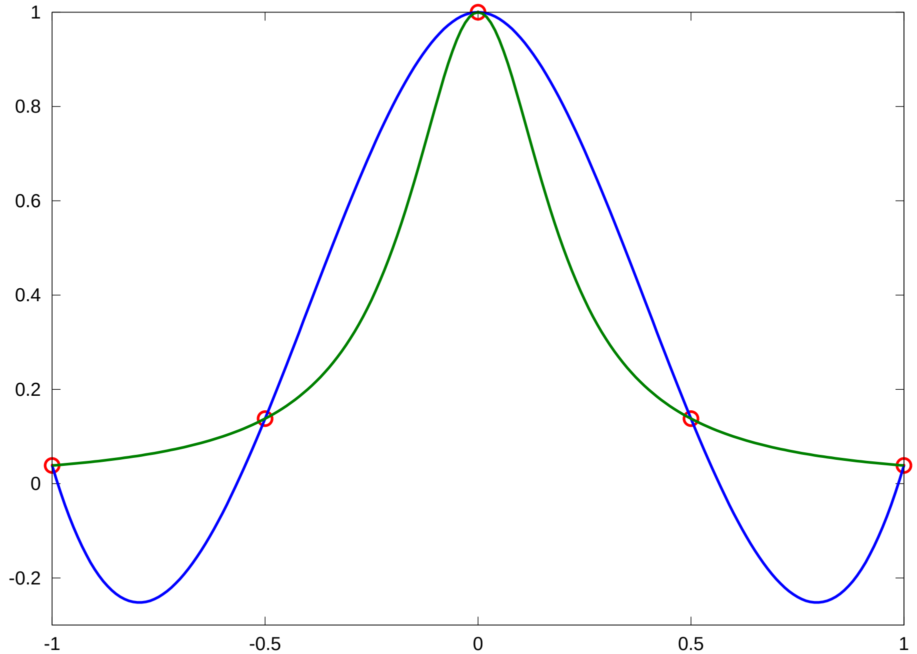
Natural cubic spline

- Cubic spline with additional two conditions: $a_1 = 0$, $a_n = 0$.
- The a_is can be obtained by solving an $(n-2) \times (n-2)$ tridiagonal linear system $\Rightarrow O(n)$.

$$\begin{pmatrix} 2(h_1+h_2) & h_2 & & \\ h_2 & 2(h_2+h_3) & h_3 & \\ & \ddots & \ddots & \ddots \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} a_2 \\ a_3 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} \frac{y_3-y_2}{h_2} - \frac{y_2-y_1}{h_1} \\ \frac{y_4-y_3}{h_3} - \frac{y_3-y_2}{h_2} \\ \vdots \\ \frac{y_6-y_5}{h_5} - \frac{y_5-y_4}{h_4} \end{pmatrix}$$

An example LVF pp.319

$$X = [\ -1 \quad -0.5 \quad 0 \quad 0.5 \quad 1\ ]$$
$$Y = [\ 0.0385 \quad 0.1379 \quad 1 \quad 0.1379 \quad 0.0385\ ]$$



Matlab command:

```
xi = -1:0.01:1;
yi = spline(X,Y,xi);
plot(X,Y,'o',xi,yi,'-');
```

Hermite Interpolation

Interpolation of derivatives

- Suppose the polynomial does not only interpolate data points, but also their derivatives.

- Given $(x_1, y_1, y_1'), (x_2, y_2, y_2')$, find a cubic polynomial $H(x) = a_1 + a_2 x + a_3 x^2 + a_4 x^3$ such that

$$\begin{cases} H(x_1) &= a_1 + a_2 x_1 + a_3 x_1^2 + a_4 x_1^3 &= y_1 \\ H(x_2) &= a_1 + a_2 x_2 + a_3 x_2^2 + a_4 x_2^3 &= y_2 \\ H'(x_1) &= a_2 + 2a_3 x_1 + 3a_4 x_1^2 &= y_1' \\ H'(x_2) &= a_2 + 2a_3 x_2 + 3a_4 x_2^2 &= y_2' \end{cases}$$

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_1 & 3x_1^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_1' \\ y_2' \end{pmatrix}$$

Hermite interpolation LVF pp.306

- Let $H(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)^2 + a_4(x-x_1)^2(x-x_2)$.
- $H'(x) = a_2 + 2a_3(x-x_1) + a_4[2(x-x_1)(x-x_2) + (x-x_1)^2]$

$$\begin{cases} H(x_1) &= a_1 = y_1 & (1) \\ H'(x_1) &= a_2 = y_1' & (2) \\ H(x_2) &= a_1 + a_2(x_2-x_1) + a_3(x_2-x_1)^2 = y_2 & (3) \\ H'(x_2) &= a_2 + 2a_3(x_2-x_1) + a_4(x_2-x_1)^2 = y_2' & (4) \end{cases}$$

- Substituting (1) and (2) into (3), $a_3 = \frac{1}{x_2-x_1}\left(\frac{y_2-y_1}{x_2-x_1} - y'\right)$
- Substituting in (4), $a_4 = \frac{1}{(x_2-x_1)^2}\left(y_2' - y_1' - 2(\frac{y_2-y_1}{x_2-x_1} - y')\right)$

- It looks like Newton's form, except repeating points $\Rightarrow$ using divided differences

z_i	w_i	Dw_i	D^2w_i	D^3w_i
$z_1 = x_1$	$w_1 = \boxed{y_1}$			
		$\boxed{y_1'}$		
$z_2 = x_1$	$w_2 = y_1$		$\boxed{\frac{Dw_2 - Dw_1}{z_3 - z_1}}$	
		$\frac{w_3 - w_2}{z_3 - z_2}$		$\boxed{\frac{D^2w_2 - D^2w_1}{z_4 - z_1}}$
$z_3 = x_2$	$w_3 = y_2$		$\frac{Dw_3 - Dw_2}{z_4 - z_2}$	
		y_2'		
$z_4 = x_2$	$w_3 = y_2$			

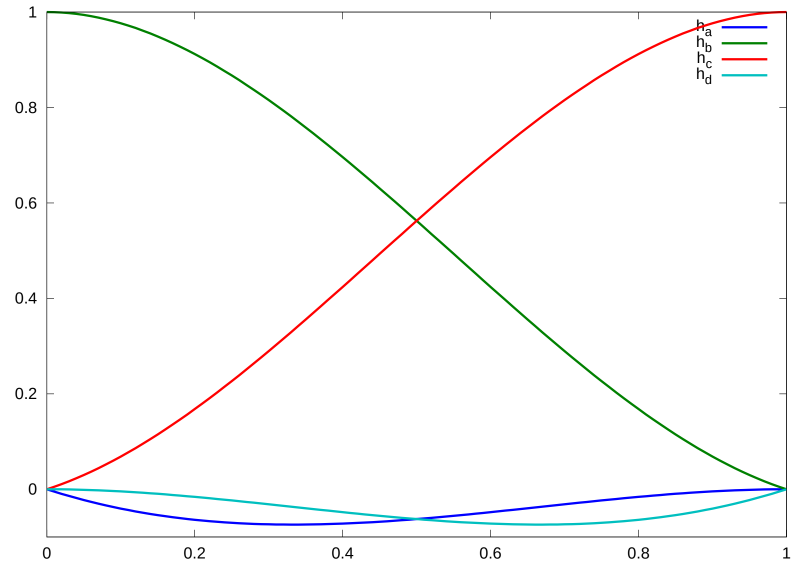
- $a_1 = y_1$, $a_2 = y_1'$, $a_3 = \frac{Dw_2 - Dw_1}{z_3 - z_1}$, $a_4 = \frac{D^2w_2 - D^2w_1}{z_4 - z_1}$.

Cubic Hermite interpolation LVF pp.308

- Cubic Hermite interpolation can be constructed by four basis functions for $x_1 = 0, x_2 = 1$,

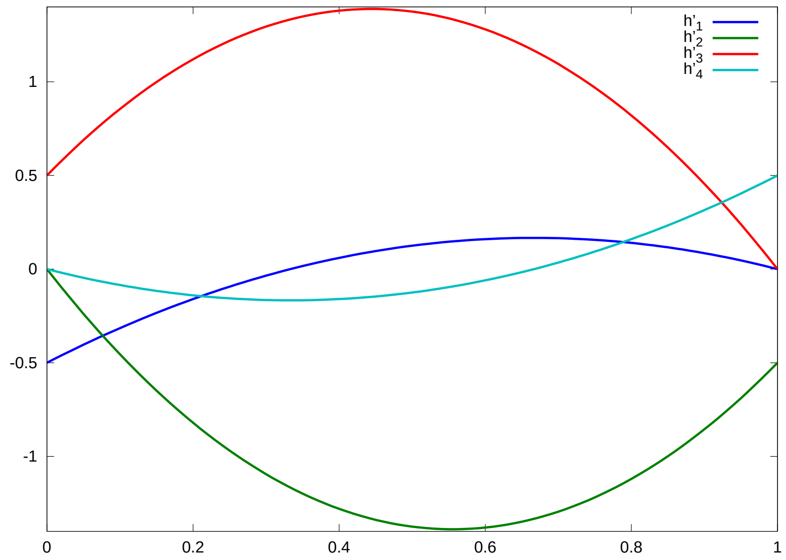
$$H(x) = a_1 h_1(x) + a_2 h_2(x) + a_3 h_3(x) + a_4 h_4(x)$$

$$\begin{aligned} h_1(x) &= -0.5x^3 + x^2 - 0.5x \\ h_2(x) &= 1.5x^3 - 2.5x^2 + 1 \\ h_3(x) &= -1.5x^3 + 2x^2 + 0.5x \\ h_4(x) &= 0.5x^3 - 0.5x^2 \end{aligned}$$



$$H'(x) = a_1 h'_1(x) + a_2 h'_2(x) + a_3 h'_3(x) + a_4 h'_4(x)$$

$$\begin{aligned} h'_1(x) &= -1.5x^2 + 2x - 0.5 \\ h'_2(x) &= 4.5x^2 - 5x \\ h'_3(x) &= -4.5x^2 + 4x + 0.5 \\ h'_4(x) &= 1.5x^2 - x \end{aligned}$$



$$\begin{cases} H(0) = y_1 & \Rightarrow \quad a_2 = y_1 \\ H(1) = y_2 & \Rightarrow \quad a_3 = y_2 \\ H'(0) = y'_1 & \Rightarrow \quad (a_3 - a_1)/2 = y'_1 \quad \Rightarrow \quad a_1 = y_2 - 2y'_1 \\ H'(1) = y'_2 & \Rightarrow \quad (a_4 - a_2)/2 = y'_2 \quad \Rightarrow \quad a_4 = y_1 + 2y'_2 \end{cases}$$

Cubic Hermite spline LVF pp.316

- Interpolate $(x_1, y_1, y_1'), (x_2, y_2, y_2'), \ldots, (x_n, y_n, y_n'), (x_{n+1}, y_{n+1}, y_{n+1}')$.
- The cubic Hermite interpolation in $[x_k, x_{k+1}]$ is

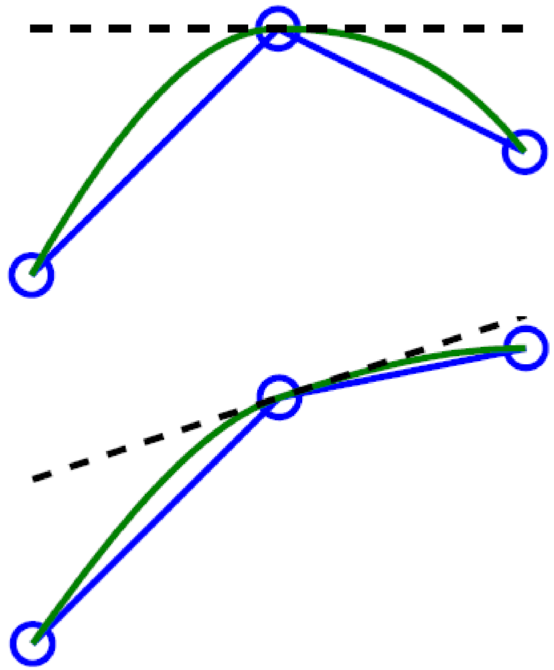
$$H(t) = a_1 h_1(t) + a_2 h_2(t) + a_3 h_3(t) + a_4 h_4(t)$$

where $t = \dfrac{x - x_k}{x_{k+1} - x_k} \left(t = \begin{cases} 0, & x = x_k; \\ 1, & x = x_{k+1}. \end{cases} \right)$

- Need to change the formula of a_1 and a_4.
  - By chain rule, $\dfrac{\partial H(t)}{\partial x} = \dfrac{\partial H(t)}{\partial t} \dfrac{\partial t}{\partial x} = \dfrac{H'(t)}{x_{k+1} - x_k}$.
  - Let $\Delta_k = (x_{k+1} - x_k)$, $\begin{cases} a_1 = y_2 - 2\Delta_k y_1' \\ a_4 = y_1 + 2\Delta_k y_2' \end{cases}$

Getting rid of the tangents

- In Matlab `pchip`, y'_k is determined by *shape-preserving method*.*



- Use the information from piecewise linear interpolation.
- Let $\rho_k = (y_{k+1} - y_k)/(x_{k+1} - x_k)$.
- If ρ_k and ρ_{k-1} have opposite signs, or one of them is zero, $y'_k = 0$.
- If ρ_k and ρ_{k-1} have same sign,

$$\frac{1}{y'_k} = \frac{1}{2}\left(\frac{1}{\rho_k} + \frac{1}{\rho_{k-1}}\right)$$

(assume $|x_{k+1} - x_k| = |x_k - x_{k-1}|$)

*Numerical Computing with MATLAB, Cleve Moler.

Bezier Curve

Bezier curves LVF pp.376

- Here we use vector form to represent points, $\mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$.
- Linear Bezier curves: given $\mathbf{p}_1, \mathbf{p}_2$, $x \in [0, 1]$

$$B^1_{\mathbf{p}_1,\mathbf{p}_2}(x) = (1-x)\mathbf{p}_1 + x\mathbf{p}_2$$

- Quadratic Bezier curves: given $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, $x \in [0, 1]$

$$\begin{aligned} B^2_{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3}(x) &= (1-x)B^1_{\mathbf{p}_1,\mathbf{p}_2}(x) + xB^1_{\mathbf{p}_2,\mathbf{p}_3}(x) \\ &= (1-x)^2\mathbf{p}_1 + 2x(1-x)\mathbf{p}_2 + x^2\mathbf{p}_3 \end{aligned}$$

- Cubic Bezier curves: given $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$, $x \in [0, 1]$

$$\begin{aligned} B^3_{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4}(x) &= (1-x)B^2_{\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3}(x) + xB^2_{\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4}(x) \\ &= (1-x)^3\mathbf{p}_1 + 3x(1-x)^2\mathbf{p}_2 + 3x^2(1-x)\mathbf{p}_3 + x^3\mathbf{p}_4 \end{aligned}$$

- The coefficient function is Bernstein polynomial.

$$B_i^n = \mathcal{C}_i^n (1-x)^{n-i} x^i,$$

where $\mathcal{C}_i^n = n!/i!(n-i)!$, binomial coefficient.

- Bezier curve of degree n: $B^n(x) = \sum_{i=0}^{n} B_i^n \mathbf{p}_i$.

- Properties
  - Passes through the first and last control points
  - Tangent to lines $\overline{\mathbf{p}_1\mathbf{p}_2}$ and $\overline{\mathbf{p}_{n-1}\mathbf{p}_n}$.
  - Lies within the convex hull of the control points.
  - Can be translated and rotated by performing these operations on the control points.