Double nodes (I)

- Given 2 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1))$$

- Interpolating polynomial of degree ≤ 1

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \end{aligned}$$

Where

$$a_0 = f(x_0), \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Double nodes (I)

- Given 2 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1))$$

- Interpolating polynomial of degree ≤ 1

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) \\ \text{with} \quad P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \end{aligned}$$

Where

$$a_0 = f(x_0), \quad a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Now let $x_1 \to x_0$, we obtain (Taylor expansion)

$$P(x) = a_0 + a_1(x - x_0),$$

where $a_0 = f(x_0)$, $a_1 = f'(x_0) \stackrel{def}{=} f[x_0, x_0]$.

$$P(x_0) = f(x_0), \quad P'(x_0) = f'(x_0).$$

Double nodes (II)

- Given 3 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)),$$

- Interpolating polynomial of degree ≤ 2

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \\ \text{with } P(x_0) &= f(x_0), \quad P(x_1) = f(x_1), \quad P(x_2) = f(x_2), \end{aligned}$$

where we have $a_0 = f[x_0], a_1 = f[x_0, x_1]$,

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Double nodes (II)

- Given 3 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)),$$

- Interpolating polynomial of degree ≤ 2

$$\begin{aligned} P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \\ \text{with} \;\; P(x_0) &= f(x_0), \;\; P(x_1) = f(x_1), \;\; P(x_2) = f(x_2), \end{aligned}$$

where we have $a_0 = f[x_0], a_1 = f[x_0, x_1]$,

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

Now let $x_2 = x_1$, we obtain (mixed interpolation)

$$\begin{aligned} f[x_1, x_2] &= f'(x_1) = f[x_1, x_1], \\ a_2 &= \frac{f[x_1, x_1] - f[x_0, x_1]}{x_1 - x_0} \overset{def}{=} f[x_0, x_1, x_1] \\ P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ P(x_0) &= f(x_0), \;\; P(x_1) = f(x_1) \;\; P'(x_1) = f'(x_1). \end{aligned}$$

Double nodes (III)

- Given 4 distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)),$$

- Interpolating polynomial of degree ≤ 3

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2).$$

where $a_0 = f[x_0], a_1 = f[x_0, x_1]$, It follows

$$\begin{aligned} a_2 &= f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ a_3 &= f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}. \end{aligned}$$

Double nodes (IV)

Let $x_1 = x_0$, and $x_3 = x_2$. It follows that

$$
\begin{aligned}
a_1 &= f[x_0, x_1] = f[x_0, x_0] \\
a_2 &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{f[x_0, x_2] - f[x_0, x_0]}{x_2 - x_0} \\
a_3 &= f[x_0, x_1, x_2, x_3] = \frac{f[x_0, x_2, x_2] - f[x_0, x_0, x_2]}{x_2 - x_0}.
\end{aligned}
$$

Hermite Interpolation:

$$
\begin{aligned}
P(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^2(x - x_2) \\
P(x_0) &= f(x_0), \;\; P'(x_0) = f'(x_0), \;\; P(x_2) = f(x_2), \;\; P'(x_2) = f'(x_2).
\end{aligned}
$$

Newton Divided Differences

- Given $m+1$ distinct points

$$(z_0, f(z_0)), (z_1, f(z_1)), \cdots, (z_m, f(z_m)),$$

- Interpolating polynomial of degree $\leq m$

$$\begin{aligned} P(x) &= a_0 + a_1(x - z_0) + a_2(x - z_0)(x - z_1) + \\ &\quad \cdots + a_m(x - z_0)(x - z_1)\cdots(x - z_{m-1}), \\ \text{with } P(z_0) &= f(z_0), \;\; P(z_1) = f(z_1), \;\; \cdots, \;\; P(z_m) = f(z_m). \end{aligned}$$

- where

$$a_j = f[z_0, z_1, \cdots, z_j], \quad \text{for} \quad j = 0, 1, \cdots, m.$$

Newton Divided Di erence Table

z_i	$f[z_i]$	$f[z_{i-1}, z_i]$	$f[z_{i-2}, z_{i-1}, z_i]$	$\cdots$	$f[z_0, z_1, \cdots, z_m]$
z_0	$\mathbf{f[z_0] = a_0}$				
		$\mathbf{f[z_0, z_1] = a_1}$			
z_1	$f[z_1]$		$\mathbf{f[z_0, z_1, z_2] = a_2}$		
		$f[z_1, z_2]$			
z_2	$f[z_2]$		$f[z_1, z_2, z_3]$		
		$f[z_2, z_3]$			
z_3	$f[z_3]$		$f[z_2, z_3, z_4]$		
		$f[z_3, z_4]$			
z_4	$f[z_4]$		$f[z_3, z_4, z_5]$	$\cdots$	$\mathbf{f[z_0, z_1, \cdots, z_m] = a_m}$
		$f[z_4, z_5]$			
z_5	$f[z_5]$		$f[z_4, z_5, z_6]$		
		$f[z_5, z_6]$			
z_6	$f[z_6]$		$f[z_5, z_6, z_7]$		
		$f[z_6, z_7]$			
z_7	$f[z_7]$		$f[z_6, z_7, z_8]$		
		$f[z_7, z_8]$			
z_8	$f[z_8]$				

Double nodes, $m = 2n + 1$, $x_j = z_{\lfloor j/2 \rfloor}$; $f[x_j; x_j] = f'(x_j)$

x_i	$f[x_i]$	$f[z_{i-1}, z_i]$	$f[z_{i-2}, z_{i-1}, z_i]$	$\cdots$	$f[z_0, z_1, \cdots, z_m]$
x_0	$\mathbf{f[x_0] = a_0}$				
		$\underline{\mathbf{f[x_0, x_0] = a_1}}$			
x_0	$f[x_0]$		$\mathbf{f[x_0, x_0, x_1] = a_2}$		
		$f[x_0, x_1]$			
x_1	$f[x_1]$		$f[x_0, x_1, x_1]$		
		$\underline{f[x_1, x_1]}$			
x_1	$f[x_1]$		$f[x_1, x_1, x_2]$		
		$f[x_1, x_2]$			
x_2	$f[x_2]$		$f[x_1, x_2, x_2]$	$\cdots$	$\mathbf{f[x_0, x_0, \cdots, x_n, x_n]}$
		$\underline{f[x_2, x_2]}$			
x_2	$f[x_2]$		$f[x_2, x_2, x_3]$		
		$f[x_2, x_3]$			
x_3	$f[x_3]$		$f[x_2, x_3, x_3]$		
		$\underline{f[x_3, x_3]}$			
x_3	$f[x_3]$				

Hermite Interpolation

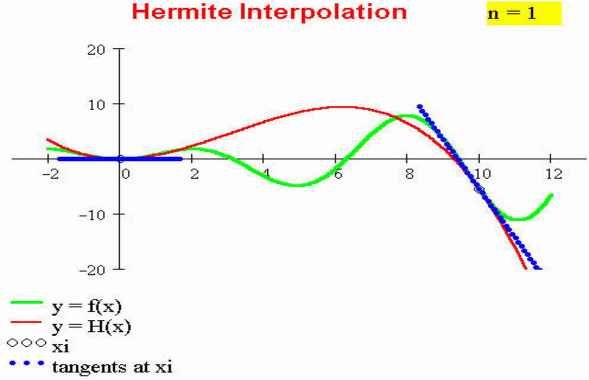
- Given $n+1$ distinct points

$$(x_0, f(x_0), f'(x_0)), (x_1, f(x_1), f'(x_1)), \cdots, (x_n, f(x_n), f'(x_n)),$$

- Interpolating polynomial $H(x)$ of degree $\leq 2n+1$ with

$$\begin{aligned} H(x_0) &= f(x_0), \quad H'(x_0) = f'(x_0), \\ H(x_1) &= f(x_1), \quad H'(x_1) = f'(x_1), \\ &\vdots \\ H(x_n) &= f(x_n), \quad H'(x_n) = f'(x_n). \end{aligned}$$

- $2n+2$ conditions, $2n+2$ coefficients in $H(x)$.


Hermite Interpolation
n = 1
20
10
-10
-20
-2
0
2
4
6
8
10
12
y = f(x)
y = H(x)
xi
tangents at xi

Hermite Interpolation: Newton Divided Differences

- Given $n+1$ distinct points

$$(x_0, f(x_0), f'(x_0)), (x_1, f(x_1), f'(x_1)), \cdots, (x_n, f(x_n), f'(x_n)),$$

- Interpolating polynomial of degree $\leq 2n+1$

$$\begin{aligned} H(x) &= a_0 + a_1(x-x_0) + a_2(x-x_0)^2 + \\ &\quad \cdots + a_{2n+1}(x-x_0)^2(x-x_1)^2\cdots(x-x_{n-1})^2(x-x_n). \\ a_j &= f[z_0, z_1, \cdots, z_j], \quad \text{for} \quad j = 0, 1, \cdots, 2n+1, \end{aligned}$$

with $z_{2i} = z_{2i+1} = x_i$ for $i = 0, 1, \cdots, n$.

Hermite Interpolation: $m = 2n + 1$, $f[x_j, x_j] = f'(x_j)$

x_i	$f[x_i]$	$f[z_{i-1}, z_i]$	$f[z_{i-2}, z_{i-1}, z_i]$	$\cdots$	$f[z_0, z_1, \cdots, z_m]$
x_0	$\mathbf{f[x_0] = a_0}$				
		$\underline{\mathbf{f[x_0, x_0] = a_1}}$			
x_0	$f[x_0]$		$\mathbf{f[x_0, x_0, x_1] = a_2}$		
		$f[x_0, x_1]$			
x_1	$f[x_1]$		$f[x_0, x_1, x_1]$		
		$\underline{f[x_1, x_1]}$			
x_1	$f[x_1]$		$f[x_1, x_1, x_2]$		
		$f[x_1, x_2]$			
x_2	$f[x_2]$		$f[x_1, x_2, x_2]$	$\cdots$	$\mathbf{f[x_0, x_0, \cdots, x_n, x_n]}$
		$\underline{f[x_2, x_2]}$			
x_2	$f[x_2]$		$f[x_2, x_2, x_3]$		
		$f[x_2, x_3]$			
x_3	$f[x_3]$		$f[x_2, x_3, x_3]$		
		$\underline{f[x_3, x_3]}$			
x_3	$f[x_3]$				

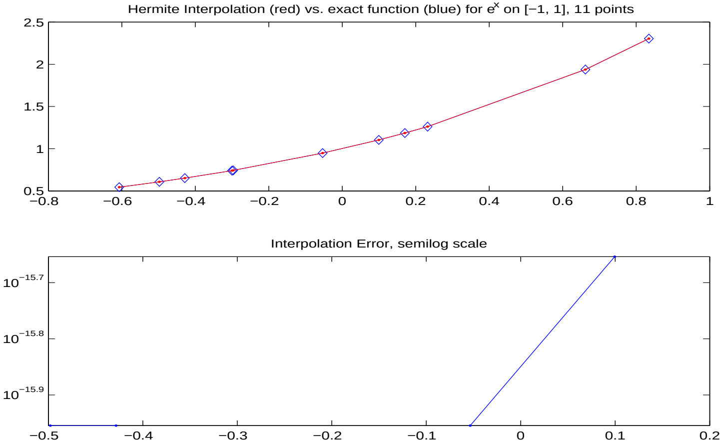
Newton Divided Differences for Hermite Interpolation

```
function F = NDD2(x,f,df)
%
% This function implements Newton's Divided Difference Algorithm
% for Hermite Interpolation. f is the vector of function values
% and df vector of derivatives.
%
% Updated by Ming Gu for Math 128A, Spring 2015
%
N = length(x);
x = x(:);
xx = reshape(repmat(x',2,1),2*N,1);
f = f(:);
df= df(:);
F  = reshape(repmat(f',2,1),2*N,1);
NN = N * 2;
F(2*(1:N))      = df;
F(1+2*(1:N-1)) = (f(2:N)-f(1:N-1))./(x(2:N)-x(1:N-1));
for k=3:2*N
    for j = 2*N:-1:k
       F(j) = (F(j)-F(j-1))/(xx(j)-xx(j-k+1));
    end
end
```

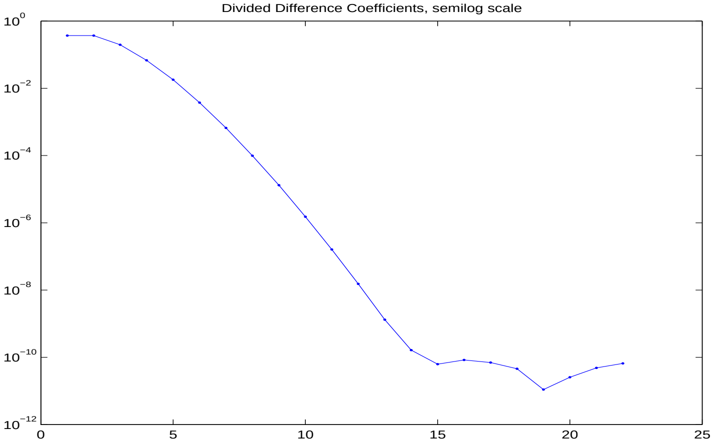
Evaluate Hermite Polynomial given coefficients

```
function f = EvaluateNDD2(xnew,x,F)
%
% This function evaluates the Hermite interpolating polynomial given
% point xnew, nodes x, and NDF coefficients F
%
n = length(x);
m = length(xnew);
f = F(2*n)*ones(m,1);
z = kron(x(:),ones(2,1));
for k=2*n-1:-1:1
    f = F(k) + f .* (xnew-z(k));
end
```

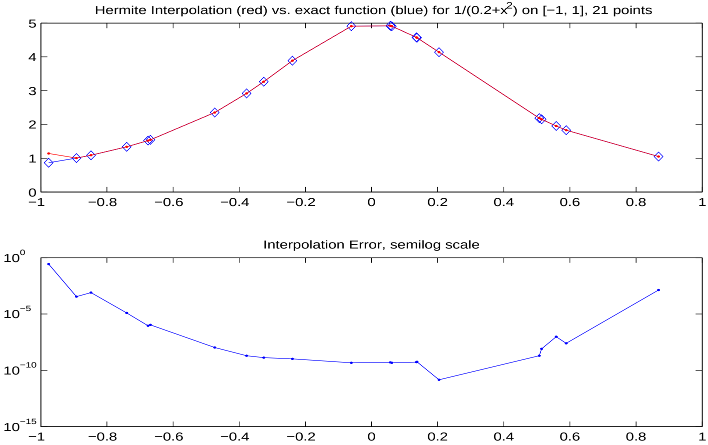
Hermite Interpolation on e^x on $[-1; 1]$



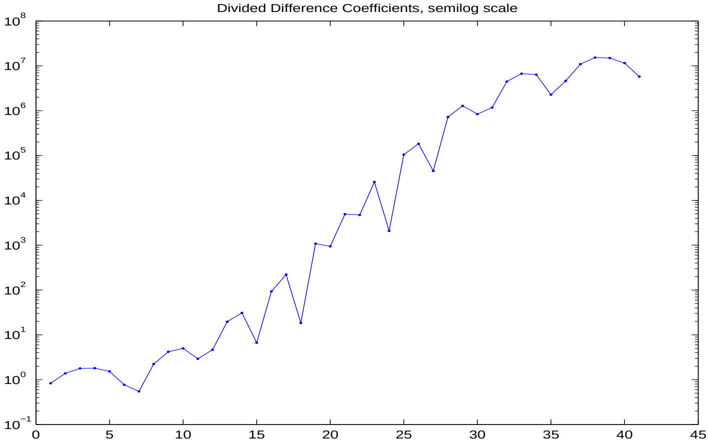
Coefficients a_j for Hermite Interpolation



Hermite Interpolation on $\frac{1}{0.2+x^2}$ on $[-1;1]$



Coefficients a_j for Hermite Interpolation



Hermite Interpolation

- Given $n+1$ distinct points

$$(x_0, f(x_0), f'(x_0)), (x_1, f(x_1), f'(x_1)), \cdots, (x_n, f(x_n), f'(x_n)),$$

- Interpolating polynomial $H(x)$ of degree $\leq 2n+1$ with

$$\begin{aligned} H(x_0) &= f(x_0), \quad H'(x_0) = f'(x_0), \\ H(x_1) &= f(x_1), \quad H'(x_1) = f'(x_1), \\ &\vdots \\ H(x_n) &= f(x_n), \quad H'(x_n) = f'(x_n). \end{aligned}$$

- $2n+2$ conditions, $2n+2$ coefficients in $H(x)$.

(Equivalent) Hermite Interpolation Solution

- Given $n+1$ distinct points

$$(x_0, f(x_0), f'(x_0)), (x_1, f(x_1), f'(x_1)), \cdots, (x_n, f(x_n), f'(x_n)),$$

- Interpolating polynomial $H(x)$ is

$$H(x) = \sum_{j=0}^{n} f(x_j)\, H_j(x) + \sum_{j=0}^{n} f'(x_j)\, \widehat{H}_j(x),$$

where

$$H_j(x) = \left(1 - 2(x - x_j)L_j'(x_j)\right) L_j^2(x), \quad \widehat{H}_j(x) = (x - x_j)L_j^2(x),$$

with $L_j(x) = \prod_{i \neq j} \dfrac{x - x_i}{x_j - x_i}$.

(Equivalent) Hermite Interpolation Solution

- Given $n+1$ distinct points

$$(x_0, f(x_0), f'(x_0)), (x_1, f(x_1), f'(x_1)), \cdots, (x_n, f(x_n), f'(x_n)),$$

- Interpolating polynomial $H(x)$ is

$$H(x) = \sum_{j=0}^{n} f(x_j)\, H_j(x) + \sum_{j=0}^{n} f'(x_j)\, \widehat{H}_j(x),$$

where

$$H_j(x) = \left(1 - 2(x - x_j)L_j'(x_j)\right) L_j^2(x), \quad \widehat{H}_j(x) = (x - x_j)L_j^2(x),$$

with $L_j(x) = \prod_{i \neq j} \dfrac{x - x_i}{x_j - x_i}$.

(Equivalent) Hermite Interpolation Solution

- Interpolating polynomial $H(x)$ is

$$H(x) = \sum_{j=0}^{n} f(x_j)\, H_j(x) + \sum_{j=0}^{n} f'(x_j)\, \widehat{H}_j(x),$$

where

$$H_j(x) = \left(1 - 2(x - x_j)L_j'(x_j)\right) L_j^2(x), \quad \widehat{H}_j(x) = (x - x_j)L_j^2(x).$$

- For $i \neq j$,

$$H_j(x_i) = 0, \;\; H_j'(x_i) = 0, \widehat{H}_j(x_i) = 0, \;\; \widehat{H}_j'(x_i) = 0.$$

- For $i = j$,

$$H_i(x_i) = 1, \;\; H_i'(x_i) = 0, \widehat{H}_i(x_i) = 0, \;\; \widehat{H}_i'(x_i) = 1.$$

Hermite Interpolation Error

Theorem: Suppose $x_0, x_1, \cdots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{2n+2}[a, b]$. Then, for each $x \in [a, b]$, a number $\xi(x)$ between $, x_0, x_1, \cdots, x_n$ (hence $\in (a, b)$) exists with

$$f(x) = H(x) + \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}(x-x_0)^2(x-x_1)^2\cdots(x-x_n)^2,$$

where $H(x)$ is the interpolating polynomial.

Hermite Interpolation Error: Proof

If $x = x_0, x_1, \cdots, \ x_n$, then error $= 0$ and theorem is true. Now let x be not equal to any node. Define function g for $t \in [a, b]$

$$g(t) \stackrel{def}{=} (f(t) - H(t)) - (f(x) - H(x)) \frac{(t - x_0)^2 (t - x_1)^2 \cdots (t - x_n)^2}{(x - x_0)^2 (x - x_1)^2 \cdots (x - x_n)^2}$$

$$= (f(t) - H(t)) - (f(x) - H(x)) \prod_{j=0}^{n} \frac{(t - x_j)^2}{(x - x_j)^2} \in C^{2n+2}[a, b].$$

Then $g(t)$ vanishes at $n + 2$ distinct points:

$$g(x) = 0, \quad g(x_k) = 0, \quad , \text{for} \quad k = 0, 1, \cdots, n.$$

and $g'(t)$ vanishes at $n + 1$ distinct points:

$$g'(x_k) = 0, \quad , \text{for} \quad k = 0, 1, \cdots, n.$$

There must be a ξ between x and nodal points such that

$$g^{(2n+2)}(\xi) = 0.$$

Hermite Interpolation Error: Proof

Since

$$g^{(2n+2)}(\xi) = (f(t) - H(t))^{(2n+2)}\,|_{t=\xi} - (f(x) - H(x))\left(\prod_{j=0}^{n} \frac{(t - x_j)^2}{(x - x_j)^2}\right)^{(2n+2)}\Big|_{\xi}$$

$$= f^{(2n+2)}(\xi) - (f(x) - H(x))\,\frac{(2n+2)!}{\prod_{j=0}^{n}(x - x_j)^2}$$

$$= 0$$

Therefore

$$f(x) = H(x) + \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\,(x - x_0)^2(x - x_1)^2 \cdots (x - x_n)^2,$$

The Splines Idea

- Given $n + 1$ distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$

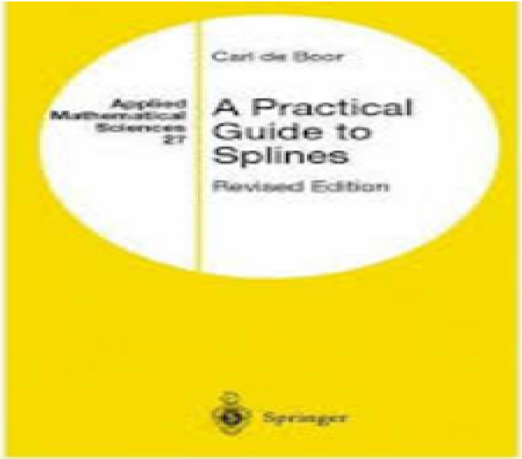
- Find *cubic spline interpolant* $S(x)$,
  - for $x \in [x_j, x_{j+1}]$, $j = 0, 1, \cdots, n-1$,

$$S(x) = S_j(x) \stackrel{def}{=} a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3.$$

    ($S(x)$ piece-wise low-degree polynomial)

  - $S(x_j) = f(x_j)$ $j = 0, 1, \cdots, n$.
    ($S(x)$ matches $f(x)$ at all nodal points)
  - $S(x) \in C^2[x_0, x_n]$.
    ($S(x)$ remains smooth-enough)

Carl de Boor: The book about Splines



Carl de Boor

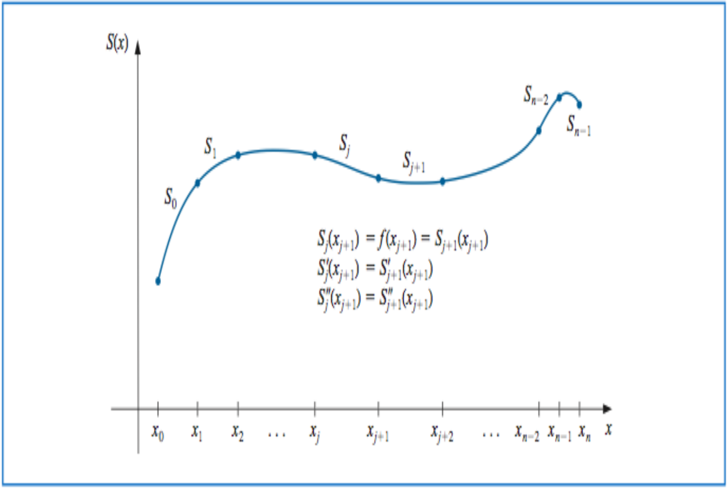
Carl-Wilhelm Reinhold de Boor

Born	3 December 1937 (age 79) Stolp, Germany (present-day Słupsk, Poland)
Fields	Mathematics (Numerical analysis)
Institutions	Purdue University University of Wisconsin–Madison University of Washington
Alma mater	University of Michigan
Notable awards	John von Neumann Prize (1996) National Medal of Science (2003)

The Splines Idea

- Spine Interpolant $S(x)$ for given $n+1$ distinct points

$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$



The Splines Equations (I)

For $x \in [x_j, x_{j+1}]$, $j = 0, 1, \cdots, n-1$,

$$S(x) = S_j(x) \stackrel{def}{=} a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3.$$

- for $j = 0, 1, \cdots, n-1$,

$$a_j = S_j(x_j) = f(x_j).$$

- let $h_j = x_{j+1} - x_j$, for $j = 0, 1, \cdots, n-1$,

$$a_{j+1} = S_{j+1}(x_{j+1}) = S_j(x_{j+1}) = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$$

$$b_j + c_j h_j + d_j h_j^2 = \frac{a_{j+1} - a_j}{h_j}, \quad j = 0, 1, \cdots, n-1.$$

This ensures $S(x) \in C[x_0, x_n]$.

The Splines Equations (II)

For $x \in [x_j, x_{j+1}], \;\; j = 0, 1, \cdots, n-1,$

$$\begin{aligned} S_j(x) &= a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \\ S_j'(x) &= b_j + 2c_j(x - x_j) + 3d_j(x - x_j)^2 \end{aligned}$$

- Define $b_n = S_{n-1}'(x_n)$; for $j = 0, \cdots, n-1$,

$$b_{j+1} = S_{j+1}'(x_{j+1}) = S_j'(x_{j+1}) = b_j + 2c_j h_j + 3d_j h_j^2.$$

This ensures $S'(x) \in C[x_0, x_n]$.

The Splines Equations (III)

For $x \in [x_j, x_{j+1}], \;\; j = 0, 1, \cdots, n-1,$

$$\begin{aligned} S_j(x) &= a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3 \\ S_j''(x) &= 2c_j + 6d_j(x - x_j) \end{aligned}$$

- Define $c_n = S_{n-1}''(x_n)/2$; for $j = 0, \cdots, n-1$,

$$2c_{j+1} = S_{j+1}''(x_{j+1}) = S_j''(x_{j+1}) = 2c_j + 6d_j h_j.$$

This ensures $S''(x) \in C[x_0, x_n]$.

The Splines Equations (IV)

For $j = 0, \cdots, n-1$,

$$2c_{j+1} = 2c_j + 6d_j h_j, \quad \text{so} \quad d_j = \frac{c_{j+1} - c_j}{3h_j}$$

$$b_j + c_j h_j + d_j h_j^2 = \frac{a_{j+1} - a_j}{h_j}, \quad \text{so} \quad b_j = -\frac{h_j}{3}(2c_j + c_{j+1}) + \frac{a_{j+1} - a_j}{h_j}.$$

$$b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2, \quad \text{so} \quad b_{j+1} = b_j + h_j(c_j + c_{j+1}).$$

Last two equations do not involve d_j. Next get rid of b_j.

The Splines Equations (V)

For $j = 0, \cdots, n-2$,

$$-\frac{h_{j+1}}{3}(2c_{j+1} + c_{j+2}) + \frac{a_{j+2} - a_{j+1}}{h_{j+1}}$$
$$= -\frac{h_j}{3}(2c_j + c_{j+1}) + \frac{a_{j+1} - a_j}{h_j} + h_j(c_j + c_{j+1}),$$

which simplifies to (for $j = 1, \cdots, n-1$)

$$h_{j-1}\, c_{j-1} + 2(h_{j-1} + h_j)\, c_j + h_j c_{j+1} = 3\left(\frac{a_{j+1} - a_j}{h_j} - \frac{a_j - a_{j-1}}{h_{j-1}}\right).$$

$n-1$ equations with $n+1$ unknowns.

Natural Splines: $S_0''(x_0) = S_{n-1}''(x_n) = 0$

- $c_0 = S_0''(x_0)/2 = 0$, $c_n = S_{n-1}''(x_n)/2 = 0$.
- Equations for $\{c_j\}_{j=1}^{n-1}$,

$$2(h_0 + h_1)\, c_1 + h_1 c_2 = 3\left(\frac{a_2 - a_1}{h_1} - \frac{a_1 - a_0}{h_0}\right),$$

$$h_{j-1}\, c_{j-1} + 2(h_{j-1} + h_j)\, c_j + h_j c_{j+1} = 3\left(\frac{a_{j+1} - a_j}{h_j} - \frac{a_j - a_{j-1}}{h_{j-1}}\right), 2 \le j \le n-2,$$

$$h_{n-2}\, c_{n-2} + 2(h_{n-2} + h_{n-1})\, c_{n-1} = 3\left(\frac{a_n - a_{n-1}}{h_{n-1}} - \frac{a_{n-1} - a_{n-2}}{h_{n-2}}\right)$$

Natural Splines: equations in matrix form

- Equations for $\{c_j\}_{j=1}^{n-1}$,

$$\begin{pmatrix} 2(h_0+h_1) & h_1 & & & \\ h_1 & 2(h_1+h_2) & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & 2(h_{n-3}+h_{n-2}) & h_{n-2} \\ & & & h_{n-2} & 2(h_{n-2}+h_{n-1}) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} = 3 \begin{pmatrix} \frac{a_2-a_1}{h_1} - \frac{a_1-a_0}{h_0} \\ \vdots \\ \frac{a_n-a_{n-1}}{h_{n-1}} - \frac{a_{n-1}-a_{n-2}}{h_{n-2}} \end{pmatrix}.$$

- Equations for $\{d_j\}_{j=0}^{n}, \{b_j\}_{j=0}^{n}$,

$$d_j = \frac{c_{j+1}-c_j}{3h_j}, \quad \text{and} \quad b_j = -\frac{h_j}{3}(2c_j + c_{j+1}) + \frac{a_{j+1}-a_j}{h_j}.$$

Clamped Splines: $S_0'(x_0) = f'(x_0)$; $S_{n-1}'(x_n) = f'(x_n)$

- Equation for c_0, c_1:

$$\begin{aligned} f'(x_0) &= S_0'(x_0) = b_0 = -\frac{h_0}{3}(2c_0 + c_1) + \frac{a_1 - a_0}{h_0}. \\ 2h_0c_0 + h_0c_1 &= 3\left(\frac{a_1 - a_0}{h_0} - f'(x_0)\right). \end{aligned}$$

- Equation for c_{n-1}, c_n:

$$\begin{aligned} f'(x_n) &= S_{n-1}'(x_n) = b_n = b_{n-1} + h_{n-1}(c_{n-1} + c_n) \\ &= -\frac{h_{n-1}}{3}(2c_{n-1} + c_n) + \frac{a_n - a_{n-1}}{h_{n-1}} + h_{n-1}(c_{n-1} + c_n), \end{aligned}$$

which leads to

$$h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3\left(f'(x_n) - \frac{a_n - a_{n-1}}{h_{n-1}}\right).$$

Clamped Splines: equations in matrix form

- Equations for $\{c_j\}_{j=0}^{n}$,

$$\begin{pmatrix} 2h_0 & h_0 & & & \\ h_0 & 2(h_0+h_1) & h_1 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ & & & h_{n-1} & 2h_{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = 3 \begin{pmatrix} \frac{a_1-a_0}{h_0} - f'(x_0) \\ \frac{a_2-a_1}{h_1} - \frac{a_1-a_0}{h_0} \\ \vdots \\ \frac{a_n-a_{n-1}}{h_{n-1}} - \frac{a_{n-1}-a_{n-2}}{h_{n-2}} \\ f'(x_n) - \frac{a_n-a_{n-1}}{h_{n-1}} \end{pmatrix}.$$

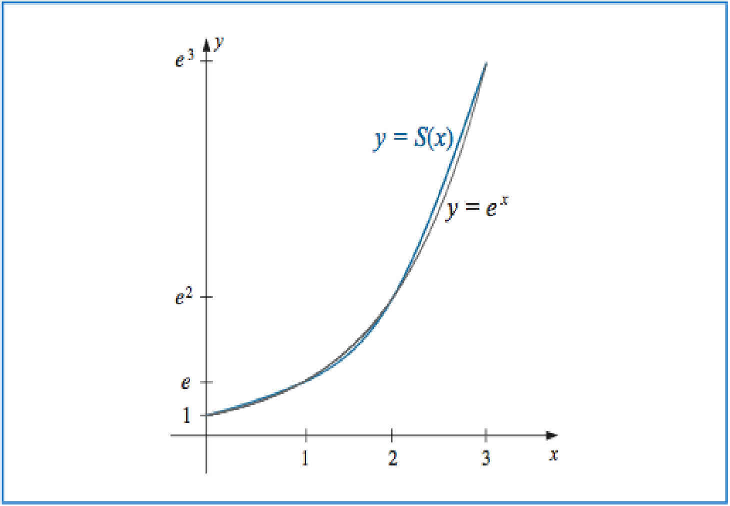
- Equations for $\{d_j\}_{j=0}^{n}, \{b_j\}_{j=0}^{n}$,

$$d_j = \frac{c_{j+1}-c_j}{3h_j}, \quad \text{and} \quad b_j = -\frac{h_j}{3}(2c_j + c_{j+1}) + \frac{a_{j+1}-a_j}{h_j}.$$

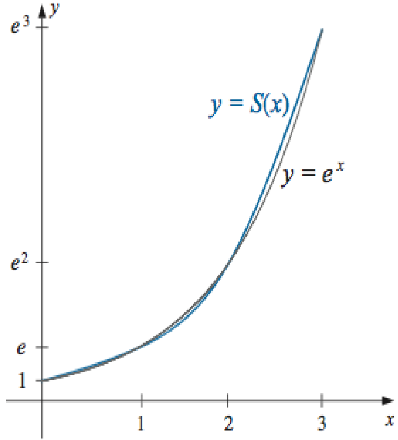
Clamped Splines

```
function Splines = ClampedSplines(x,f,df)
%
% This code implements the clamped splines
%
% Written by Ming Gu for Math 128A, Fall 2008
% Updated by Ming Gu for Math 128A, Spring 2015
%
n = length(x);
h = diff(x(:));
rhs = 3 * diff([df(1);diff(f(:))./h;df(2)]);

A = diag(h,1)+diag(h,-1)+2*diag([[0;h]+[h;0]]);

%
% compute the c coefficients. This is a simple
% but very slow way to do it.
%
c         = A \ rhs;
d         = (diff(c)./h)/3;
b         = diff(f(:))./h-(h/3).*(2*c(1:n-1)+c(2:n));
Splines.a = f(:);
Splines.b = b;
Splines.c = c;
Splines.d = d;
```

Natural Splines, $f(x) = e^x$, $x_0 = 0; x_1 = 1; x_2 = 2; x_3 = 3$



Clamped Splines, $f(x) = e^x$, $x_0 = 0; x_1 = 1; x_2 = 2; x_3 = 3; f'(0) = 1; f'(3) = e^3$



Splines

- Given $n+1$ distinct points

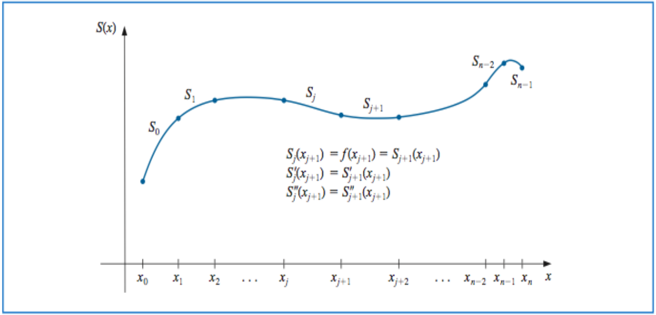
$$(x_0, f(x_0)), (x_1, f(x_1)), \cdots, (x_n, f(x_n)),$$

- Find *cubic spline interpolant* $S(x) \in C^2[x_0, x_n]$,

$$S(x) = S_j(x) \stackrel{def}{=} a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3,$$

for $x \in [x_j, x_{j+1}]$, $0 \le j \le n-1$.

- $S(x_j) = f(x_j)$, $0 \le j \le n$.

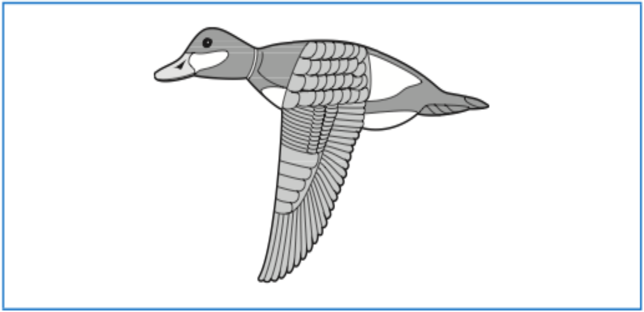


Two flavors of Splines

- **Natural Splines**: $S_0''(x_0) = S_{n-1}''(x_n) = 0$.
- **Clamped Splines**: $S_0'(x_0) = f'(x_0),\ S_{n-1}'(x_n) = f'(x_n)$.

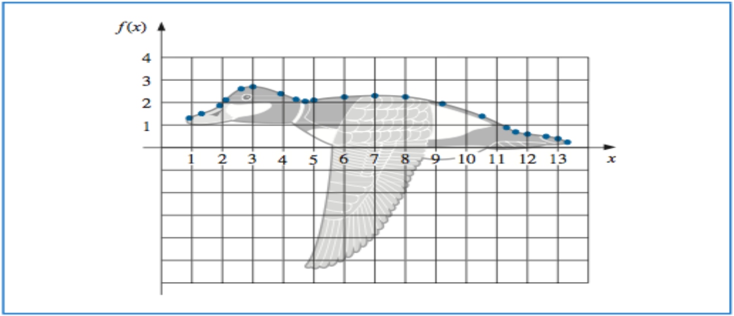
A duck in Splines

- A duck in flight



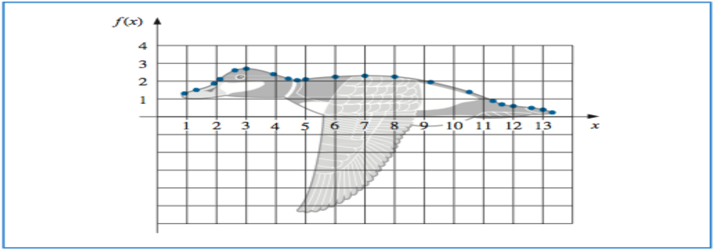
- Goal: to approximate the top profile.

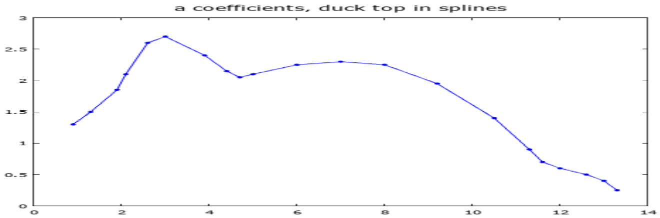
duck top profile



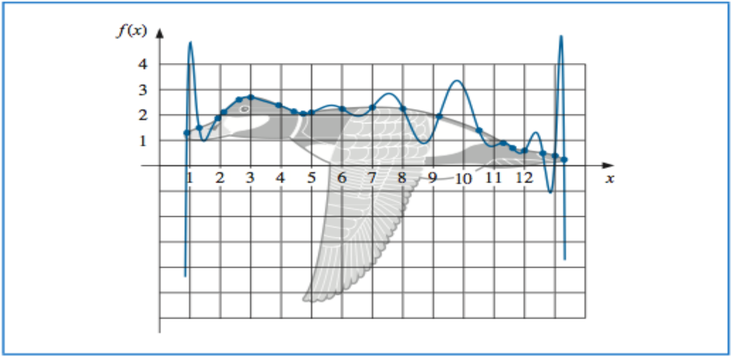
x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

Duck top profile in Natural Splines, *a* coefficients

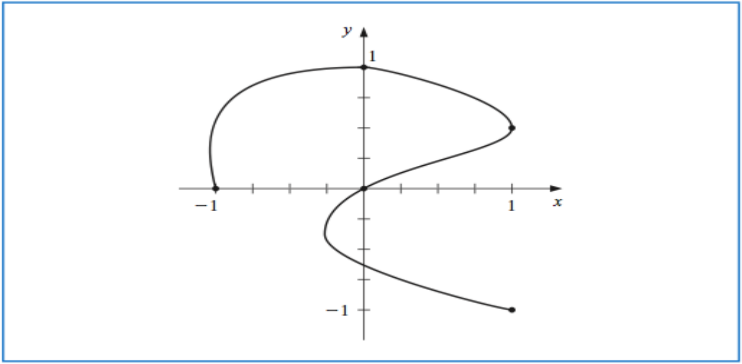




Duck top profile in 20-degree polynomial interpolation



Parametric Curves: $x = x(t)$; $y = y(t)$



Parametric Curve Approximation

- Given $n+1$ distinct points

$$(x_0, y_0), (x_1, y_1), \cdots, (x_n, y_n),$$

where

$$x_j = x(t_j), \quad y_j = y(t_j), \quad j = 0, 1, \cdots, n.$$

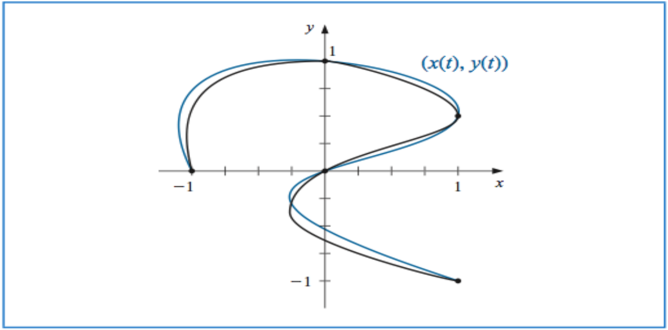
- Example

i	0	1	2	3	4
t_i	0	0.25	0.5	0.75	1
x_i	−1	0	1	0	1
y_i	0	1	0.5	0	−1

Parametric Curve Approximation

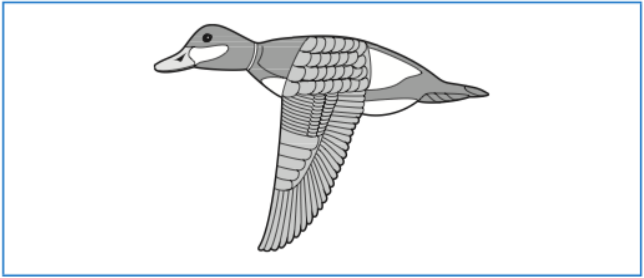
- Polynomial interpolation on $x = x(t)$, and $y = y(t)$, respectively.

$$
\begin{aligned}
x(t) &= \left(\left(\left(64t - \frac{352}{3}\right) t + 60\right) t - \frac{14}{3}\right) t - 1, \\
y(t) &= \left(\left(\left(-\frac{64}{3} t + 48\right) t - \frac{116}{3}\right) t + 11\right) t.
\end{aligned}
$$



Parametric Curves in Computer Graphics

- **Design**: Piece-wise cubic Hermite polynomial.
- **Feature**: Each cubic Hermite polynomial is completely determined by function/derivative at endpoints.
- **Consequence**:, Each portion of the curve can be changed while leaving most of the curve the same.



As duck flies, parametric curve can effectively evolve.

Parametric Curves: Piece-wise cubic Hermite polynomials

- **Given**: $n+1$ data points $(x(t_0), y(t_0)), \cdots, (x(t_n), y(t_n))$.
- **Given**: $n+1$ derivatives $\frac{dy}{dx}|_{t=t_i}$, $0 \le i \le n$.

- **Find**: Piece-wise cubic Hermite polynomial:

$$x = x_i(t), \quad y = y_i(t), \quad \text{for} \quad i \in [t_i, t_{i+1}], \quad 0 \le i \le n,$$

such that

$$\begin{aligned} x_i(t_i) &= x(t_i), \;\; y_i(t_i) = y(t_i), \;\; x_i(t_{i+1}) = x(t_{i+1}), \;\; y_i(t_{i+1}) = y(t_{i+1}), \\ \frac{dy}{dx}\Big|_{t=t_i} &= \frac{y_i'(t_i)}{x_i'(t_i)}, \quad \text{and} \quad \frac{dy}{dx}\Big|_{t=t_{i+1}} = \frac{y_i'(t_{i+1})}{x_i'(t_{i+1})} \end{aligned}$$

8 parameters in $x_i(t), y_i(t)$ with 6 given conditions for each i.

Guidepoints guide slopes (assume $[t_i; t_{i+1}] = [0; 1]$)

- The cubic Hermite polynomial $x(t)$ on $[0, 1]$ satisfies

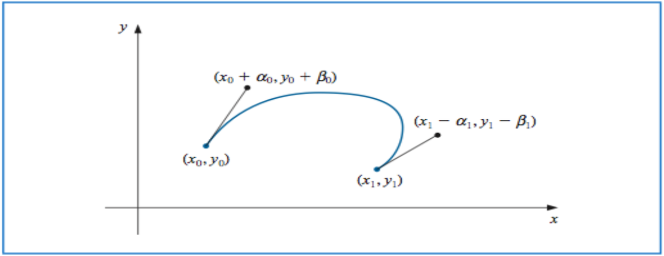
$$x(0) = x_0, \quad x(1) = x_1, x'(0) = \alpha_0, \quad \text{and} \quad x'(1) = \alpha_1.$$

- The cubic Hermite polynomial $y(t)$ on $[0, 1]$ satisfies

$$y(0) = y_0, \quad y(1) = y_1, y'(0) = \beta_0, \quad \text{and} \quad y'(1) = \beta_1.$$

- Guidepoints guide slopes

$$\frac{dy}{dx}\Big|_{t=0} = \frac{\beta_0}{\alpha_0}, \quad \text{and} \quad \frac{dy}{dx}\Big|_{t=1} = \frac{\beta_1}{\alpha_1}.$$



Guidepoints, example

- **Given** end points

$$(x_0, y_0) = (0, 0), \quad \text{and} \quad (x_1, y_1) = (1, 0).$$

Guidepoints, example

- **Given** end points

$$(x_0, y_0) = (0, 0), \quad \text{and} \quad (x_1, y_1) = (1, 0).$$

- **Given** end points

$$(x_0 + \alpha_0, y_0 + \beta_0) = (1, 1), \quad \text{and} \quad (x_1 - \alpha_1, y_1 - \beta_1) = (0, 1).$$

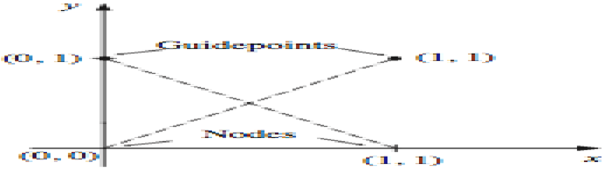
Guidepoints, example

- **Given** end points

$$(x_0, y_0) = (0, 0), \quad \text{and} \quad (x_1, y_1) = (1, 0).$$

- **Given** end points

$$(x_0 + \alpha_0, y_0 + \beta_0) = (1, 1), \quad \text{and} \quad (x_1 - \alpha_1, y_1 - \beta_1) = (0, 1).$$



- The cubic Hermite polynomial $x(t) = t$ satisfies

$$x(0) = 0, \;\; x(1) = 1, x'(0) = 1, \;\; \text{and} \;\; x'(1) = 1.$$

- The cubic Hermite polynomial $x(t) = t$ satisfies

$$x(0) = 0, \quad x(1) = 1, x'(0) = 1, \quad \text{and} \quad x'(1) = 1.$$

- The cubic Hermite polynomial $y(t) = -t^2 + t$ satisfies

$$y(0) = 0, \quad y(1) = 0, y'(0) = 1, \quad \text{and} \quad y'(1) = -1.$$

- The cubic Hermite polynomial $x(t) = t$ satisfies

$$x(0) = 0, \;\; x(1) = 1, x'(0) = 1, \;\; \text{and} \;\; x'(1) = 1.$$

- The cubic Hermite polynomial $y(t) = -t^2 + t$ satisfies

$$y(0) = 0, \;\; y(1) = 0, y'(0) = 1, \;\; \text{and} \;\; y'(1) = -1.$$

- Slopes

$$\frac{dy}{dx}\Big|_{t=0} = \frac{1}{1} = 1, \quad \text{and} \quad \frac{dy}{dx}\Big|_{t=1} = \frac{-1}{1} = -1.$$

- The cubic Hermite polynomial $x(t) = t$ satisfies

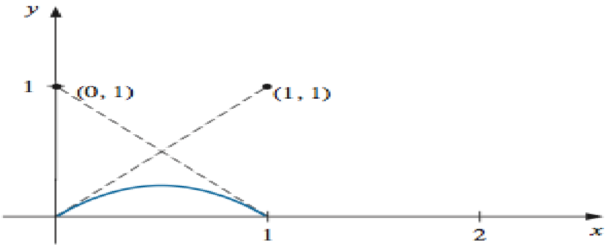
$$x(0) = 0, \;\; x(1) = 1, x'(0) = 1, \;\; \text{and} \;\; x'(1) = 1.$$

- The cubic Hermite polynomial $y(t) = -t^2 + t$ satisfies

$$y(0) = 0, \;\; y(1) = 0, y'(0) = 1, \;\; \text{and} \;\; y'(1) = -1.$$

- Slopes

$$\frac{dy}{dx}\Big|_{t=0} = \frac{1}{1} = 1, \quad \text{and} \quad \frac{dy}{dx}\Big|_{t=1} = \frac{-1}{1} = -1.$$



Numerical Differentiation

Derivative of given function $f(x)$ at x_0

$$
\begin{aligned}
f'(x_0) &= \lim_{h\to 0} \frac{f(x_0+h)-f(x_0)}{h} \\
&\approx \frac{f(x_0+h)-f(x_0)}{h} \quad \text{for small values of } h.
\end{aligned}
$$

How good is this approximation?

By Taylor expansion, there is a ξ between x_0 and x_0+h,

$$
\begin{aligned}
f(x_0+h) &= f(x_0) + h\,f'(x_0) + \frac{1}{2}h^2\,f''(\xi), \\
\text{therefore} \quad f'(x_0) &= \frac{f(x_0+h)-f(x_0)}{h} - \frac{1}{2}h\,f''(\xi).
\end{aligned}
$$

Numerical Differentiation

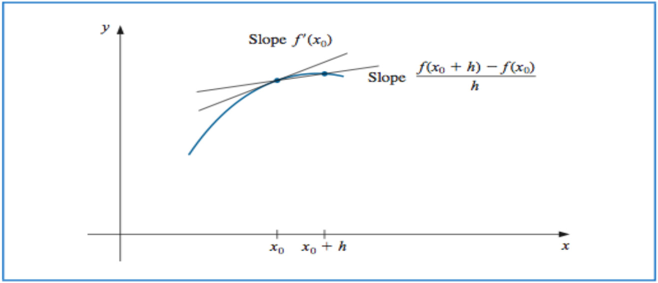
$$
\begin{aligned}
f'(x_0) &= \frac{f(x_0+h)-f(x_0)}{h} - \frac{1}{2} h\, f''(\xi) \\
&\approx \frac{f(x_0+h)-f(x_0)}{h}.
\end{aligned}
$$

- **forward-di erence formula** for $h > 0$,
- **backward-di erence formula** for $h < 0$.

Numerical Differentiation

$$
\begin{aligned}
f'(x_0) &= \frac{f(x_0+h)-f(x_0)}{h} - \frac{1}{2} h\, f''(\xi) \\
&\approx \frac{f(x_0+h)-f(x_0)}{h}.
\end{aligned}
$$

- **forward-di erence formula** for $h > 0$,
- **backward-di erence formula** for $h < 0$.



Numerical Differentiation, example

Example: Use the forward-difference formula to approximate the derivative of $f(x) = \mathbf{ln}(x)$ at $x_0 = 1.8$, using $= 0.1, 0.05, 0.01$.

Because $f''(\xi) = -1/\xi^2$ for $\xi \in [x_0, x_0 + h] \subset [1.8, 1.9]$, it follows that the approximation error

$$\frac{1}{2}\left|h\, f''(\xi)\right| \leq \frac{1}{2}\left|\frac{h}{1.8^2}\right|.$$

Numerical Differentiation, example

Example: Use the forward-difference formula to approximate the derivative of $f(x) = \textbf{ln}(x)$ at $x_0 = 1.8$, using $= 0.1, 0.05, 0.01$.

Because $f''(\xi) = -1/\xi^2$ for $\xi \in [x_0, x_0 + h] \subset [1.8, 1.9]$, it follows that the approximation error

$$\frac{1}{2}\left|h\, f''(\xi)\right| \leq \frac{1}{2}\left|\frac{h}{1.8^2}\right|.$$

h	$f(1.8+h)$	$\frac{f(1.8+h)-f(1.8)}{h}$	$\frac{\|h\|}{2(1.8)^2}$
0.1	0.64185389	0.5406722	0.0154321
0.05	0.61518564	0.5479795	0.0077160
0.01	0.59332685	0.5540180	0.0015432

Numerical Differentiation, via polynomial interpolation

Suppose that $\{x_0, x_1, \cdots, x_n\}$ are $n+1$ distinct numbers,

$$f(x) = \left(\sum_{j=0}^{n} f(x_j) L_j(x)\right) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i), \quad L_j(x) = \prod_{i \neq j} \frac{(x - x_i)}{(x_j - x_i)},$$

for some $\xi(x)$.

Numerical Differentiation, via polynomial interpolation

Suppose that $\{x_0, x_1, \cdots, x_n\}$ are $n+1$ distinct numbers,

$$f(x) = \left(\sum_{j=0}^{n} f(x_j) L_j(x)\right) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i), \quad L_j(x) = \prod_{i \neq j} \frac{(x - x_i)}{(x_j - x_i)},$$

for some $\xi(x)$. So

$$\begin{aligned} f'(x) &= \left(\sum_{j=0}^{n} f(x_j) L_j'(x)\right) + \left(\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\right) \frac{d}{dx} \left(\prod_{i=0}^{n} (x - x_i)\right) \\ &\quad + \frac{d}{dx} \left(\frac{f^{(n+1)}(\xi(x))}{(n+1)!}\right) \left(\prod_{i=0}^{n} (x - x_i)\right). \end{aligned}$$

Messy. But last term $= 0$ for $x = x_k, k = 0, 1, \cdots, n$

Numerical Differentiation, via polynomial interpolation

$$
\begin{aligned}
f'(x_k) &= \left(\sum_{j=0}^{n} f(x_j) L_j'(x_k)\right) + \left(\frac{f^{(n+1)}(\xi(x_k))}{(n+1)!}\right) \left(\prod_{i \neq k} (x_k - x_i)\right) \\
&\approx \sum_{j=0}^{n} f(x_j) L_j'(x_k).
\end{aligned}
$$

$(n+1)$**-point formula**, works for any nodal choices.

3-point formulas ($n = 2$), $j = 0; 1; 2$

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad \text{we have} \quad L_0'(x) = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}.$$

3-point formulas ($n = 2$), $j = 0; 1; 2$

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad \text{we have} \quad L_0'(x) = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}.$$

$$L_1'(x) = \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \quad \text{and} \quad L_2'(x) = \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}.$$

3-point formulas ($n = 2$), $j = 0; 1; 2$

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad \text{we have} \quad L_0'(x) = \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}.$$

$$L_1'(x) = \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \quad \text{and} \quad L_2'(x) = \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}.$$

$$f'(x_j) = f(x_0)\left[\frac{2x_j - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}\right] + f(x_1)\left[\frac{2x_j - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}\right] + f(x_2)\left[\frac{2x_j - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}\right] + \frac{1}{6} f^{(3)}(\xi_j) \prod_{\substack{k=0 \\ k \neq j}}^{2} (x_j - x_k),$$

3-point formulas ($n = 2$), equi-spaced nodes

Choose $x_1 = x_0 + h, x_2 = x_0 + 2h$, with $h \neq 0$.

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0).$$

3-point formulas ($n = 2$), equi-spaced nodes

Choose $x_1 = x_0 + h, x_2 = x_0 + 2h$, with $h \neq 0$.

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0).$$

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_0 + h) - \frac{1}{2}f(x_0 + 2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_0),$$

3-point formulas ($n = 2$), equi-spaced nodes

Choose $x_1 = x_0 + h, x_2 = x_0 + 2h$, with $h \neq 0$.

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0).$$

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_0+h) - \frac{1}{2}f(x_0+2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_0),$$

$$f'(x_0+h) = \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0+2h)\right] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

3-point formulas ($n = 2$), equi-spaced nodes

Choose $x_1 = x_0 + h, x_2 = x_0 + 2h$, with $h \neq 0$.

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_1) - \frac{1}{2}f(x_2)\right] + \frac{h^2}{3}f^{(3)}(\xi_0).$$

$$f'(x_0) = \frac{1}{h}\left[-\frac{3}{2}f(x_0) + 2f(x_0+h) - \frac{1}{2}f(x_0+2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_0),$$

$$f'(x_0+h) = \frac{1}{h}\left[-\frac{1}{2}f(x_0) + \frac{1}{2}f(x_0+2h)\right] - \frac{h^2}{6}f^{(3)}(\xi_1),$$

$$f'(x_0+2h) = \frac{1}{h}\left[\frac{1}{2}f(x_0) - 2f(x_0+h) + \frac{3}{2}f(x_0+2h)\right] + \frac{h^2}{3}f^{(3)}(\xi_2).$$

5-point formulas, equi-spaced nodes

Choose $x_1 = x_0 - h, x_2 = x_0 - 2h,\ x_3 = x_0 + h, x_4 = x_0 + 2h$,with $h \neq 0$.

$$f'(x_0) = \frac{1}{12h}[f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)] + \frac{h^4}{30}f^{(5)}(\xi),$$

Second order derivatives, equi-spaced points

$$
\begin{aligned}
f(x_0 + h) &= f(x_0) + f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 + \frac{1}{6} f'''(x_0)\,h^3 \\
&\quad + \frac{1}{24} f^{(4)}(\xi_+)\,h^4, \\
f(x_0 - h) &= f(x_0) - f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 - \frac{1}{6} f'''(x_0)\,h^3 \\
&\quad + \frac{1}{24} f^{(4)}(\xi_-)\,h^4.
\end{aligned}
$$

Second order derivatives, equi-spaced points

$$
\begin{aligned}
f(x_0+h) &= f(x_0) + f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 + \frac{1}{6} f'''(x_0)\,h^3 \\
&\quad + \frac{1}{24} f^{(4)}(\xi_+)\,h^4, \\
f(x_0-h) &= f(x_0) - f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 - \frac{1}{6} f'''(x_0)\,h^3 \\
&\quad + \frac{1}{24} f^{(4)}(\xi_-)\,h^4.
\end{aligned}
$$

Adding up, terms with difference signs cancel,

$$
\begin{aligned}
f(x_0+h) + f(x_0-h) &= 2f(x_0) + f''(x_0)\,h^2 + \frac{h^4}{24}\left(f^{(4)}(\xi_+) + f^{(4)}(\xi_-)\right) \\
&= 2f(x_0) + f''(x_0)\,h^2 + \frac{2h^4}{24} f^{(4)}(\xi).
\end{aligned}
$$

Second order derivatives, equi-spaced points

$$f(x_0+h) = f(x_0) + f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 + \frac{1}{6} f'''(x_0)\,h^3 + \frac{1}{24} f^{(4)}(\xi_+)\,h^4,$$

$$f(x_0-h) = f(x_0) - f'(x_0)\,h + \frac{1}{2} f''(x_0)\,h^2 - \frac{1}{6} f'''(x_0)\,h^3 + \frac{1}{24} f^{(4)}(\xi_-)\,h^4.$$

Adding up, terms with difference signs cancel,

$$\begin{aligned} f(x_0+h) + f(x_0-h) &= 2f(x_0) + f''(x_0)\,h^2 + \frac{h^4}{24}\left(f^{(4)}(\xi_+) + f^{(4)}(\xi_-)\right) \\ &= 2f(x_0) + f''(x_0)\,h^2 + \frac{2h^4}{24} f^{(4)}(\xi). \end{aligned}$$

Therefore

$$f''(x_0) = \frac{f(x_0+h) + f(x_0-h) - 2f(x_0)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi).$$

Round-Off Error Instability

(For example) three-point midpoint formula

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} - \frac{h^2}{6} f^{(3)}(\xi).$$

- every step in the computation could incur round-off error.
- division by $2h$ could magnify round-off error.

Assume round-off error model

$$f(x_0+h) = \widehat{f}(x_0+h) + e(x_0+h) \quad \text{and} \quad f(x_0-h) = \widehat{f}(x_0-h) + e(x_0-h),$$

where $|e(x_0+h)| \le \epsilon$, $|e(x_0-h)| \le \epsilon$. No other round-off errors.

It follows

$$f'(x_0) - \frac{\widehat{f}(x_0+h) - \widehat{f}(x_0-h)}{2h} = \frac{e(x_0+h) - e(x_0-h)}{2h} - \frac{h^2}{6} f^{(3)}(\xi)$$

It follows

$$\begin{aligned} f'(x_0) - \frac{\widehat{f}(x_0+h) - \widehat{f}(x_0-h)}{2h} &= \frac{e(x_0+h) - e(x_0-h)}{2h} - \frac{h^2}{6} f^{(3)}(\xi) \\ \left| f'(x_0) - \frac{\widehat{f}(x_0+h) - \widehat{f}(x_0-h)}{2h} \right| &\leq \left| \frac{e(x_0+h) - e(x_0-h)}{2h} \right| + \frac{h^2}{6} |f^{(3)}(\xi)|. \end{aligned}$$

Assume $|e(x_0+h)| \leq \epsilon$, $|e(x_0-h)| \leq \epsilon$ and $|f^{(3)}(\xi)| \leq M$, an upper bound on $|f^{(3)}(x)|$,

$$\left| f'(x_0) - \frac{\widehat{f}(x_0+h) - \widehat{f}(x_0-h)}{2h} \right| \leq \frac{\epsilon}{h} + \frac{M\,h^2}{6} \stackrel{def}{=} e(h).$$

Round-Off Error Instability: optimal h choice

$$e(h) = \frac{\epsilon}{h} + \frac{M\,h^2}{6}$$

is smallest at

$$h_{\mathbf{min}} = \left(\frac{3\epsilon}{M}\right)^{\frac{1}{3}} = O\left(\epsilon^{\frac{1}{3}}\right),$$

with

$$e(h_{\mathbf{min}}) = \frac{1}{2}\left(\frac{9\epsilon^2}{M}\right)^{\frac{1}{3}} = O\left(\epsilon^{\frac{2}{3}}\right).$$