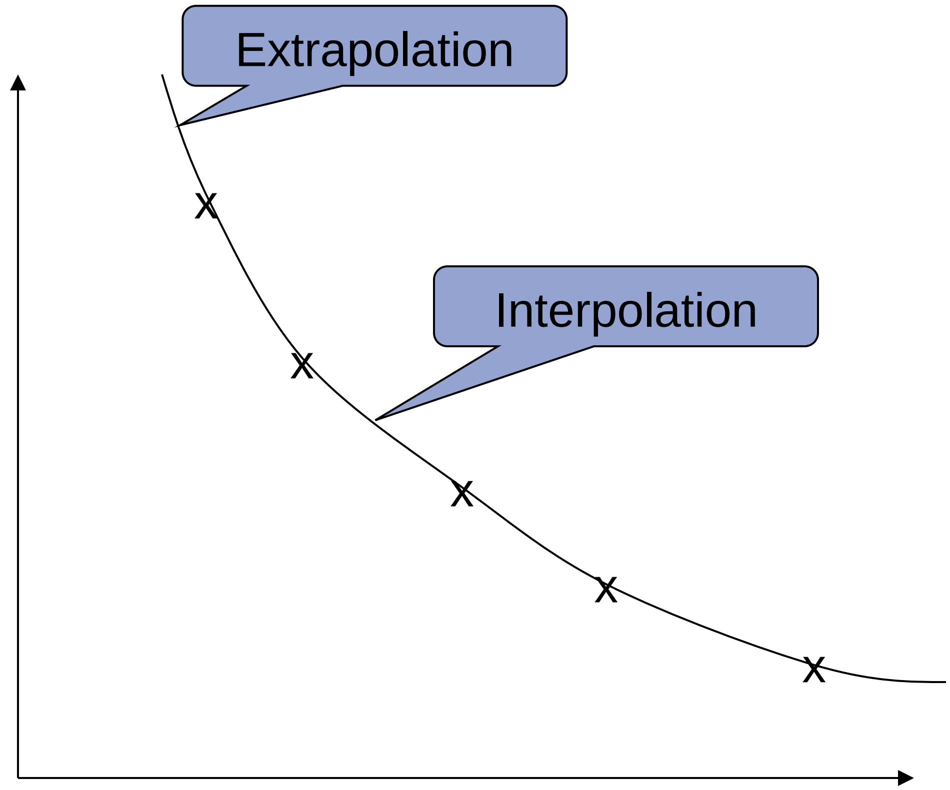
Numerical Analysis – Interpolation

Hanyang University

Jong-Il Park

Fitting

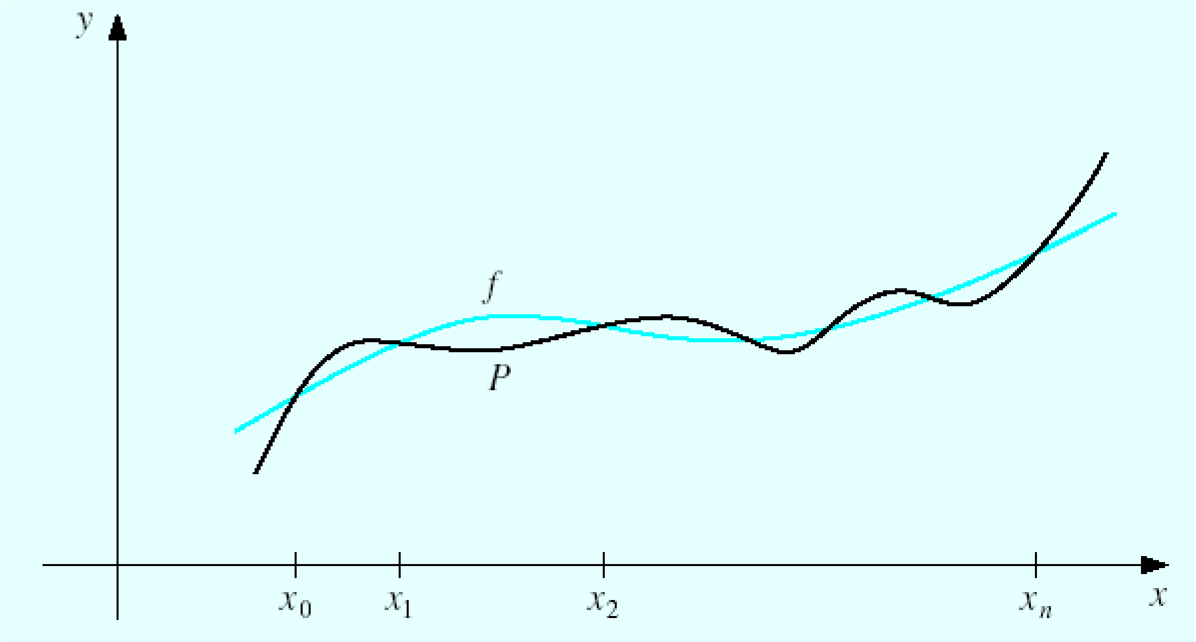
- Exact fit
  - Interpolation
  - Extrapolation
- Approximate fit



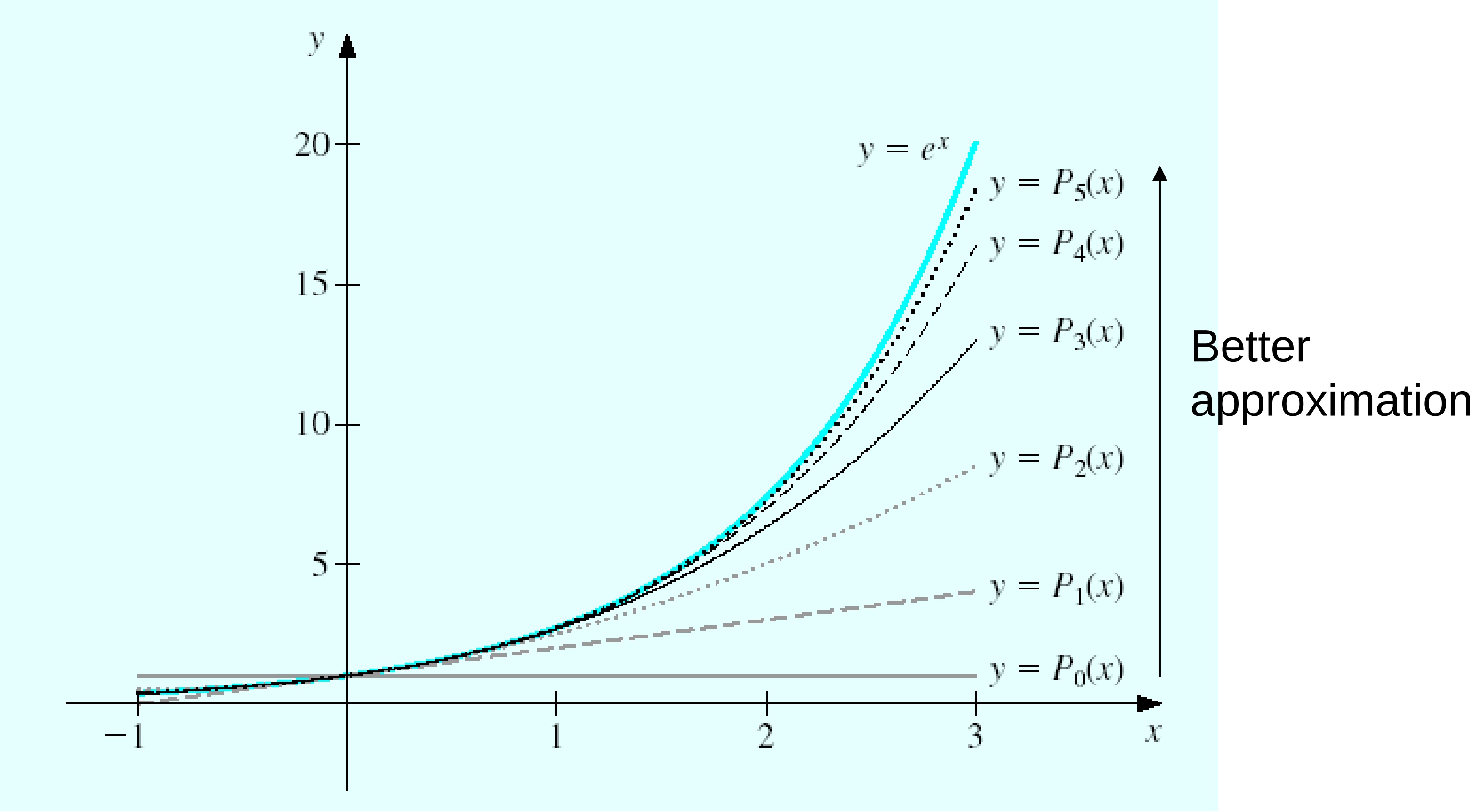
Weierstrass Approximation Theorem

Suppose that f is defined and continuous on $[a, b]$. For each $\varepsilon > 0$, there exists a polynomial $P(x)$ defined on $[a, b]$, with the property that

$$|f(x) - P(x)| < \varepsilon, \qquad \text{for all } x \in [a, b].$$



Approximation error



Lagrange Interpolating Polynomial

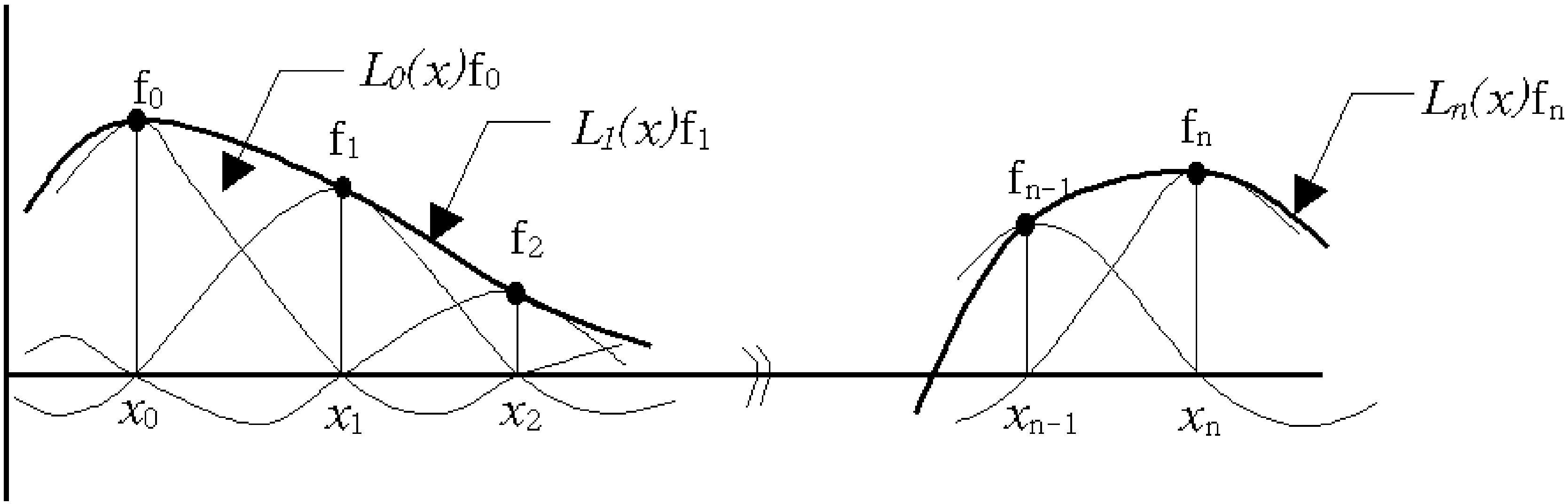
$$L_o(x) = \frac{g_o(x)}{g_o(x_o)} = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_o-x_1)(x_o-x_2)\cdots(x_o-x_n)} = \begin{cases} 1, & x = x_o \\ 0, & x = x_1, x_2, \cdots, x_n \end{cases}$$

$$L_i(x) = \prod_{k=0,\, k\neq i}^{n} \left(\frac{x - x_k}{x_i - x_k} \right)$$

$$P_n(x) = L_o(x) f_o + L_1(x) f_1 + \cdots + L_n(x) f_n$$

Illustration of Lagrange polynomial

$$P_n(x) = L_o(x) f_o + L_1(x) f_1 + \cdots + L_n(x) f_n$$



- Unique
- Too much complex

Error analysis for intpl. polynml(I)

$$\begin{aligned} E(x) &= f(x) - P_n(x) \\ &= (x-x_o)(x-x_1)\cdots(x-x_n)\ u(x) \end{aligned}$$

then

$$f(x) - P_n(x) - (x-x_o)(x-x_1)\cdots(x-x_n)u(x) = 0$$

Let

$$v(t) \equiv f(t) - P_n(t) - (t-x_o)(t-x_1)\cdots(t-x_n)u(x)$$

$v(t)=0$ has (n+2) solutions s.t.

$$\begin{cases} v(x_i)=0 \quad (i=0,1,2,\cdots,n) \\ v(x)=0 \end{cases}$$

Error analysis for intpl. polynml(II)

※**Generalized Roll's Theorem**

$$\left[\begin{array}{lll} v'(t)=0 & \Rightarrow & (n+1)\ \text{solutions} \\ v''(t)=0 & \Rightarrow & n\ \text{solutions} \\ \vdots & & \vdots \\ v^{(n+1)}(t)=0 & \Rightarrow & 1\ \text{solutions} \end{array}\right.$$

$t=\xi$

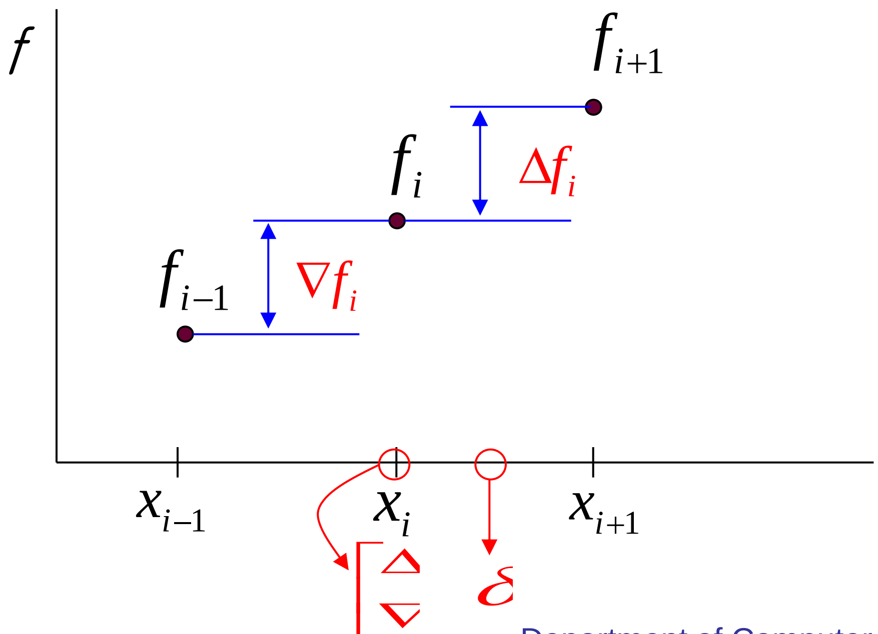
$$v^{(n+1)}(\xi) = \frac{d^{n+1}}{dt^{n+1}} v(t)\Big|_{t=\xi}$$
$$= f^{(n+1)}(\xi) - 0 - (n+1)!\,u(x) = 0,$$
$$x_0 \le \xi \le x_n$$

$$\therefore u(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

$$\boxed{\therefore E(x) = (x-x_0)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!},\quad x_0 \le \xi \le x_n}$$

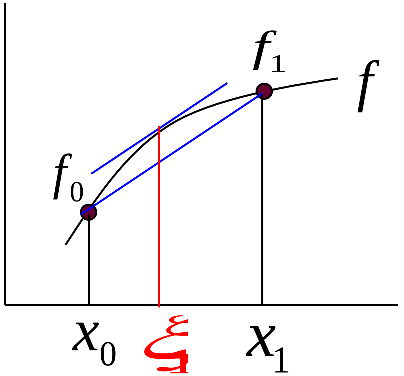
Differences

- Difference
  - Forward difference : $\Delta f_i = \Delta f(x_i) = f_{i+1} - f_i$
  - Backward difference : $\nabla f_i = \nabla f(x_i) = f_i - f_{i-1}$
  - Central difference : $\delta f_{i+\frac{1}{2}} = \delta f(x_{i+\frac{1}{2}}) = f_{i+1} - f_i$



Divided Differences

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = f'(\xi_1), \quad x_0 \le \xi_1 \le x_1$$



; 1st order divided difference

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

$$= \frac{f_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{f_1}{(x_1 - x_0)(x_1 - x_2)} + \frac{f_2}{(x_2 - x_0)(x_2 - x_1)}$$

; 2nd order divided difference

N-th divided difference

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

$$=\frac{f_0}{(x_0-x_1)\cdots(x_0-x_n)}+\cdots+\frac{f_n}{(x_n-x_0)\cdots(x_n-x_{n-1})}$$

x	$f(x)$	First Divided Differences	Second Divided Differences	Third Divided Differences
x_0	$f[x_0]$			
		$f[x_0,x_1]=\frac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f[x_1]$		$f[x_0,x_1,x_2]=\frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
		$f[x_1,x_2]=\frac{f[x_2]-f[x_1]}{x_2-x_1}$		$f[x_0,x_1,x_2,x_3]=\frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_2	$f[x_2]$		$f[x_1,x_2,x_3]=\frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	
		$f[x_2,x_3]=\frac{f[x_3]-f[x_2]}{x_3-x_2}$		$f[x_1,x_2,x_3,x_4]=\frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_3	$f[x_3]$		$f[x_2,x_3,x_4]=\frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	
		$f[x_3,x_4]=\frac{f[x_4]-f[x_3]}{x_4-x_3}$		$f[x_2,x_3,x_4,x_5]=\frac{f[x_3,x_4,x_5]-f[x_2,x_3,x_4]}{x_5-x_2}$
x_4	$f[x_4]$		$f[x_3,x_4,x_5]=\frac{f[x_4,x_5]-f[x_3,x_4]}{x_5-x_3}$	
		$f[x_4,x_5]=\frac{f[x_5]-f[x_4]}{x_5-x_4}$		
x_5	$f[x_5]$			

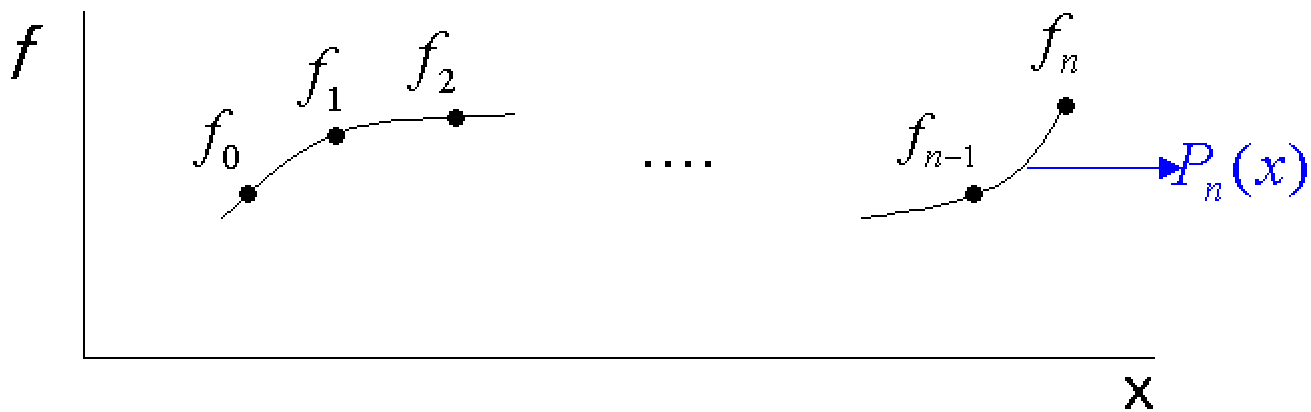
Newton's Intpl. Polynomials(I)

Assume $(n+1)$ points $(x_0, f_0), \cdots, (x_n, f_n)$ are on

$$P_n(x) = f_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \cdots + (x - x_0)(x - x_1)\cdots(x - x_{n-1})a_n$$

Then

$$P_n(x) - P_{n-1}(x) = (x - x_0)(x - x_1)\cdots(x - x_{n-1})a_n$$



Newton's Intpl. Polynomials(II)

Since $P_n(x_n) = f_n$,

$$a_n = \frac{f_n - P_{n-1}(x_n)}{(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1})}$$

It is easy to show that

$$a_n = f[x_0, x_1, \cdots, x_n]$$

Therefore

$$\boxed{\begin{aligned} P_1(x) &= f_0 + (x - x_0) f[x_0, x_1] \\ &\vdots \\ P_n(x) &= f_0 + (x - x_0) f[x_0, x_1] + \cdots \\ &\quad + (x - x_0)(x - x_1)\cdots(x - x_{n-1}) f[x_0, \cdots, x_n] \end{aligned}}$$

Different interpretation

Finding the coefficients of Newton polynomial
= Solving linear equations

$$\begin{bmatrix} 1 & & \ldots & & 0 \\ 1 & x_1 - x_0 & & & \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & & \vdots \\ \vdots & \vdots & & \ddots & \\ 1 & x_k - x_0 & \ldots & \ldots & \prod_{j=0}^{k-1}(x_k - x_j) \end{bmatrix} \begin{bmatrix} a_0 \\ \\ \vdots \\ \\ a_k \end{bmatrix} = \begin{bmatrix} y_0 \\ \\ \vdots \\ \\ y_k \end{bmatrix}$$

[from Wikipedia]

Newton's Forward Difference Interpolating Polynomials(I)

- Equal Interval *h*

$$x_i = x_0 + i \cdot h$$

- Derivation

n=1

$$a_1 = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f_1 - f_0}{h} = \frac{1}{1!\,h}\Delta f_0$$

n=2

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

$$= \frac{\Delta f_1 / h - \Delta f_0 / h}{x_2 - x_0} = \frac{\Delta f_1 - \Delta f_0}{2h^2} = \frac{1}{2!\,h^2}\Delta^2 f_0$$

Newton's Forward Difference Interpolating Polynomials(II)

Generalization

$$a_n = f[x_0, \cdots, x_n] = \frac{f[x_1, \cdots, x_n] - f[x_0, \cdots, x_{n-1}]}{nh}$$

$$= \frac{1}{n! h^n} \Delta^n f_0$$

$$P_n(x) = f_0 + s\Delta f_0 + \frac{s(s-1)}{2!}\Delta^2 f_0 + \cdots + \frac{s(s-1)(s-n+1)}{n!}\Delta^n f_0$$

$$= \sum_{k=0}^{n} \underbrace{\binom{s}{k}}_{\text{Binomial coef.}} \Delta^k f_0 \qquad \boxed{(x = x_0 + sh)}$$

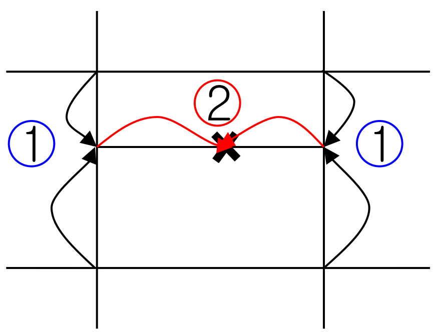
- Error Analysis

$$E(x) = \binom{s}{n+1} h^{n+1} f^{(n+1)}(\xi) = 0(h^{n+1}), \qquad x_0 \le \xi \le x_n$$

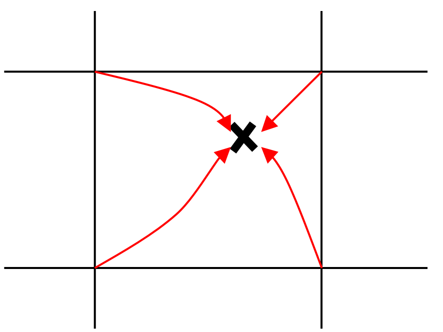
$$\cong \binom{s}{n+1} \Delta^{n+1} f_0$$

Intpl. of Multivariate Function

- Successive univariate polynomial
- Direct mutivariate polynomial

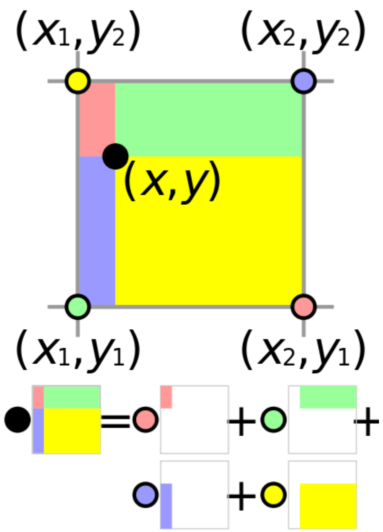


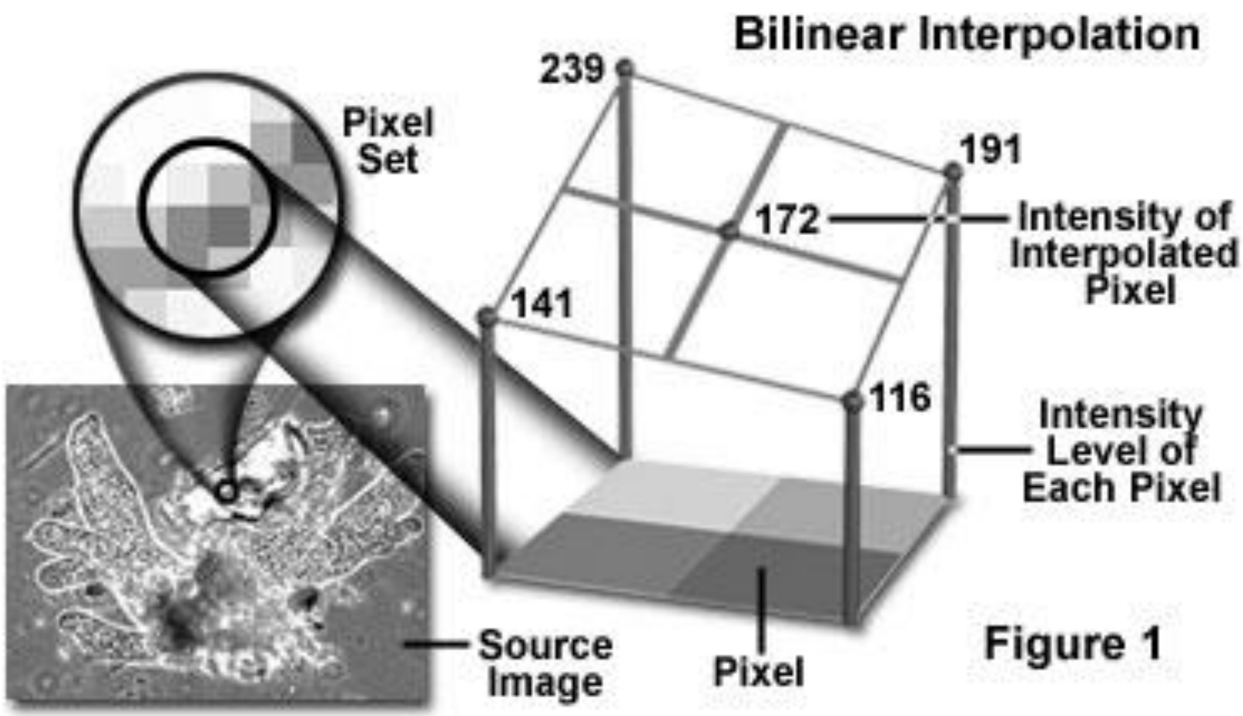
Successive univariate



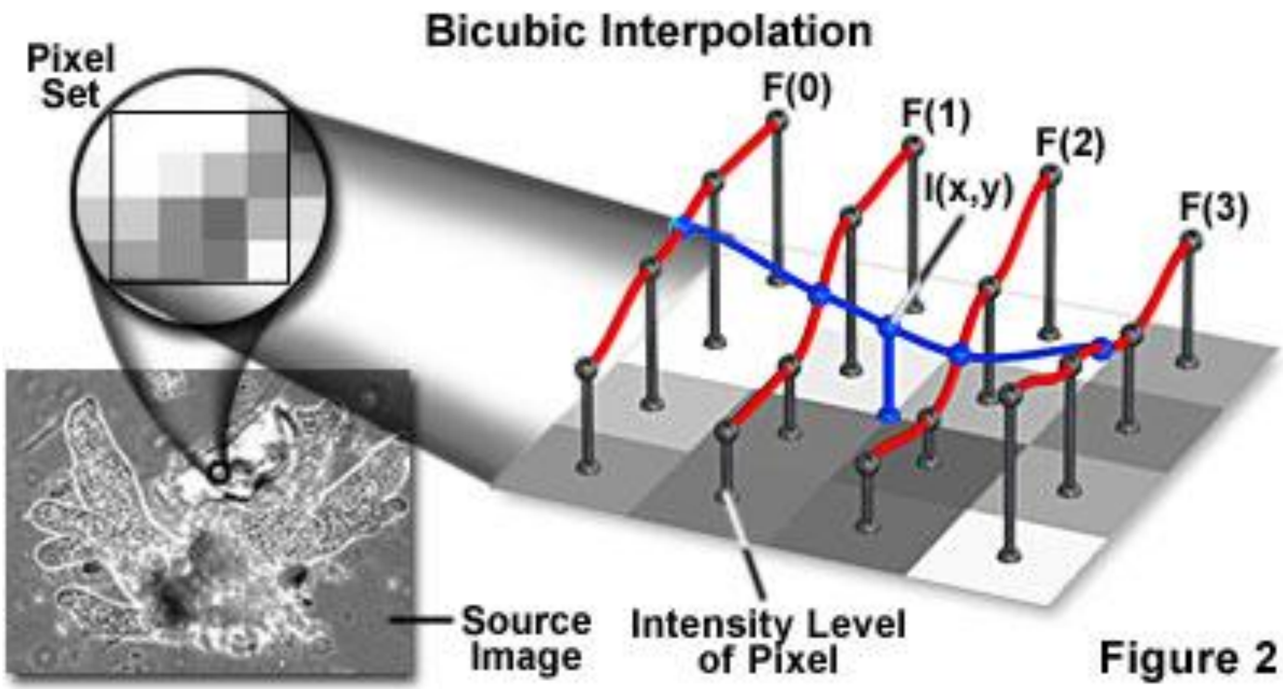
direct multivariate

Eg. Bilinear Interpolation





Eg. Bicubic interpolation



Inverse Interpolation

= finding x(f)

- Utilization of Newton's polynomial

$$f(x) \cong P_n(x) = f_i + (x - x_i) f[x_i, x_{i+1}]$$
$$+ (x - x_i)(x - x_{i+1}) f[x_i, x_{i+1}, x_{i+2}]$$
$$\vdots$$
$$+ (x - x_i)\cdots(x - x_{i+n-1}) f[x_i, \cdots, x_{i+n}]$$

Solve for x

$$x = \frac{f(x) - f_i - \{(x - x_i)(x - x_{i+1}) f[x_i, x_{i+1}, x_{i+2}] + \cdots\}}{f[x_i, x_{i+1}]} + x_i$$

1st approximation

$$x_1 = \frac{f(x) - f_i}{f[x_i, x_{i+1}]} + x_i$$

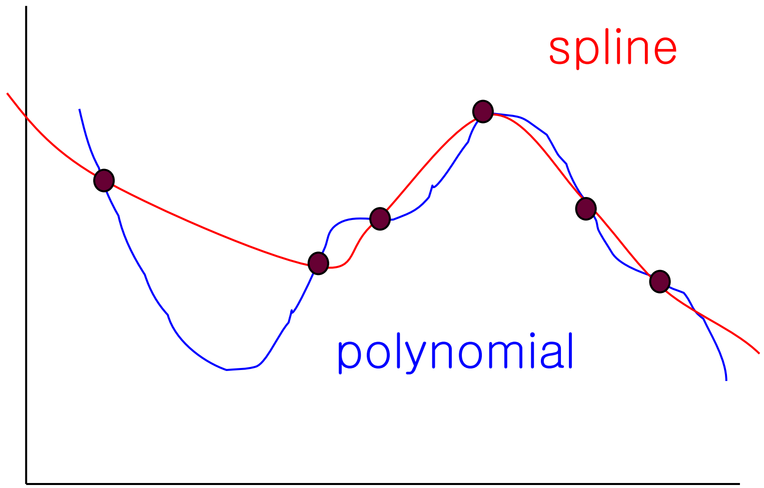
2nd approximation

$$x_2 = \frac{f(x) - f_i - (x_1 - x_i)(x_1 - x_{i+1}) f[x_i, x_{i+1}, x_{i+2}]}{f[x_i, x_{i+1}]} + x_i$$

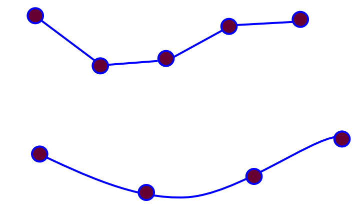
Repeat until a convergence

Spline Interpolation

- Why spline?

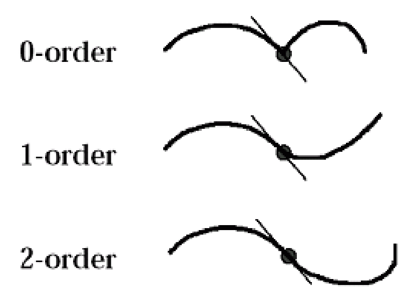


- Linear spline
- Quadratic spline
- Cubic spline



- $f_{i-1}'(x_i) = f_i'(x_i)$
- $f_{i-1}''(x_i) = f_i''(x_i)$

Continuity



- Good approximation !!
- Moderate complexity !!

Cubic spline interpolation(I)

- Cubic Spline Interpolation at an interval $[x_i, x_{i+1}]$

$$f_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

4 unknowns for each interval
4n unknowns for n intervals

Conditions 1) $f_i(x_i) = y_i, \quad i = 0,1,\cdots, n-1$ n

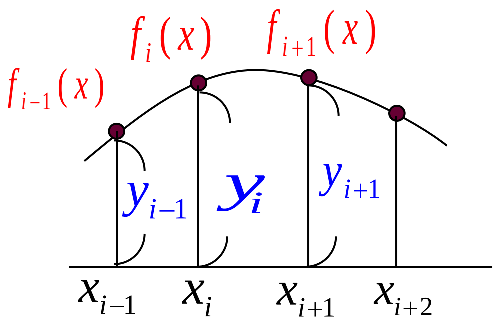
2) $f_i(x_{i+1}) = y_{i+1}, \quad i = 0,1,\cdots, n-1$ n

3) continuity of f'

$f'_{i-1}(x_i) = f'_i(x_i), \quad i = 1,2,\cdots, n-1$ n−1

4) continuity of f''

$f''_{i-1}(x_i) = f''_i(x_i), \quad i = 1,2,\cdots, n-1$ n−1



Cubic spline interpolation(II)

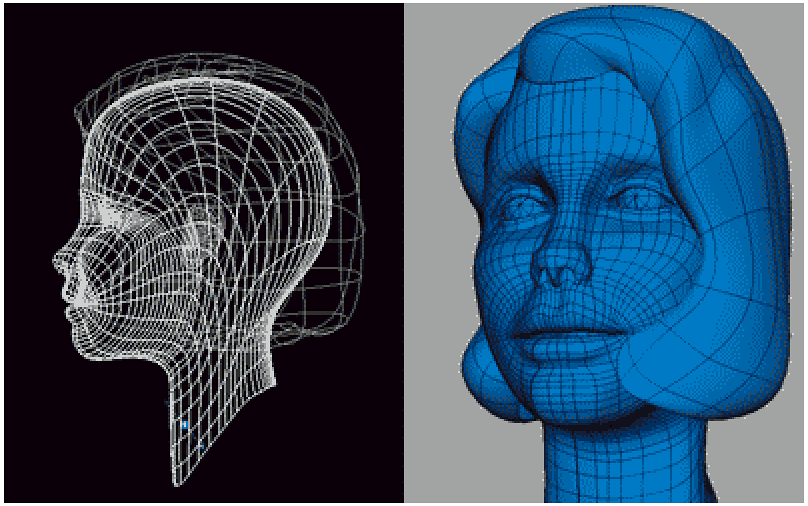
- Determining boundary condition

Method 1 : $f_0''(x_0) = 0,\ f_{n-1}''(x_n) = 0$ natual cubic

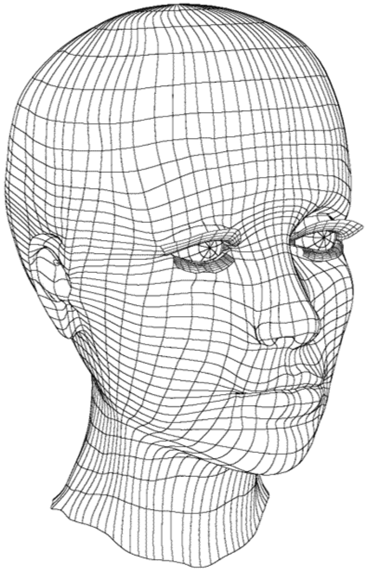
Method 2 : $f_0''(x_0) = f_0''(x_1),\ f_{n-1}''(x_{n-1}) = f_{n-1}''(x_n)$

Method 3 : $f_1''(x_1) = \dfrac{f_0''(x_0) + f_2''(x_2)}{2},\ f_{n-1}''(x_{n-1}) = \dfrac{f_{n-2}''(x_{n-2}) + f_{n-1}''(x_n)}{2}$

Eg. CG modeling



Non-Uniform Rational B-Spline



Homework 5

[Due: 17 Nov.]

- Programming: Resampling of image
  - Read an image file and identify the resolution and resample it to a specified resolution
  - Input: Target resolution(M’xN’)
  - Output: Resampled image
  - Method: Bilinear interpolation