MA 780

Ch.8 Interpolation / Approximation

<u>Problem</u>: (Algebraic Interpolation), Given a set of paired data

x	x_0	x_1	$\cdots$	x_n
y	y_0	y_1	$\cdots$	y_n

observed/measured sampled data from experiment - data table

$\begin{cases} x \text{ is an independent variable} \\ y \text{ is the dependent variable.} \end{cases}$ $y = y(x)$.

We want to find the value $y = y(x)$ at any point x.

<u>Interpolation solution</u>: Select a number of <u>known</u> (simple) functions

$$\varphi_0(x),\ \varphi_1(x),\ \cdots,\ \varphi_m(x)$$

and form a linear combination

$$y(x) \approx \varphi(x) = \underline{a_0}\,\varphi_0(x) + \underline{a_1}\,\varphi_1(x) + \cdots + \underline{a_m}\,\varphi_m(x)$$

If we know the coefficients $a_0, a_1, \cdots, a_m$, then we can evaluate $y(x)$ at any point x.

If $\varphi(x_i) = y_i$, for all i, such approximation is called interpolation.

x_i is called a node, or a nodal point, or sample point. generally $x_i \neq x_j$, distinct

$\{\varphi_i(x)\}$ is a set of basis function. We should chose $\{\varphi_i(x)\}$

(i) They are linear independent (no redundant)

(ii) Simple
- Polynomials, $1, x, \cdots, x^m$ or other form
- trigo-functions $\sin x, \cos x, \sin 2x, \cos 2x$ (FFT)
- Rational function Padé approximation

etc.

<u>Motivations</u>
- prediction
- simplification
- Applications
  - Quadrature
  - Multigrid
  - Solve ODE/PDE

Other approximation / interpolation ($y_i = \varphi(x_i)$),

- Non-linear model

$$\varphi(x) = \alpha e^{\beta x}, \quad \alpha, \beta \text{ are unknow}$$

$\min \| f(x) - \varphi(x) \|_p$, if $p = \infty$, The solution is called Chebyshev <u>polynomial</u> $T_n(x) = \cos(n \arccos x)$ $-1 \le x \le 1$

<u>Polynomial Interpolation</u>

$\{\varphi_i(x)\}$, $1, x, x^2, \cdots, x^m$

or $1, x-1, (x+2)^2, \cdots, (x-5)^m + 3(x-2)$

basis functions are not unique.

The problem is: Given

x	x_0	x_1	$\cdots$	x_n
y	y_0	y_1	$\cdots$	y_n

Assuming $x_i \neq x_j$

Find $p_m(x) = a_0 + a_1 x + \cdots + a_m x^m = \sum_{j=0}^{m} a_j x^j$

to approximate $y(x)$.

If $m < n$, It is curve fitting using (~~over~~-determined)

$$p_m(x_i) = \sum_{j=0}^{m} a_j x_i^j, \quad i = 0, 1, \cdots n.$$

$\Bigg\{$ Least square problem. MA580
QR method

$m > n$ (under-determined), infinite solution

SVD solution $\min_{Ax=b} \|x\|_2$.

✓ $m = n$, There is an unique solution ~~and~~ if $x_i \neq x_j$.

$p(x_i) = y_i, \quad i = 0, 1, \cdots, n.$ Interpolation

QR, SVD methods still can be used, but not very efficient. Cost $O(\frac{2n^3}{3})$, Storage $O(n^2)$

Example: Linear interpolation

x	x_0	x_1
y	y_0	y_1

A. $p_1(x) = a_0 + a_1 x$

$$y_0 = a_0 + a_1 x_0$$
$$y_1 = a_0 + a_1 x_1$$

$$a_0 = \frac{\begin{vmatrix} y_0 & x_0 \\ y_1 & x_1 \end{vmatrix}}{\begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix}} = \frac{y_0 x_1 - x_0 y_1}{x_1 - x_0}$$

$$a_1 = \frac{\begin{vmatrix} 1 & y_0 \\ 1 & y_1 \end{vmatrix}}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

B: $p_1(x) = \square\, y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$ point + slope

Basis, 1, $x - x_1$, It is called the Newton's interpolation

C: $p_1(x) = \frac{x - x_2}{x_1 - x_2} y_1 + \frac{x - x_1}{x_2 - x_1} y_2$

Basis function $\frac{x - x_2}{x_1 - x_2}$, $\frac{x - x_1}{x_2 - x_1}$, It is called the Langrange Interpolation. They are all the same (in theory), different expressions, but efficiency may vary when implemented.

Quadratic Interpolation:

x	x_0	x_1	x_2
y	y_0	y_1	y_2

$p_2(x) = a_0 + a_1 x + a_2 x^2$

A:

$$p_2(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 = y_0$$
$$p_2(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 = y_1$$
$$p_2(x_2) = a_0 + a_1 x_2 + a_2 x_2^2 = y_2$$

Unknown a_0, a_1, a_2, coefficient matrix

Vandermonde matrix of 3×3

$$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$\det(A) = \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} = \det \begin{pmatrix} 1 & x_0 & x_0^2 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 \\ 0 & x_2 - x_0 & x_2^2 - x_0^2 \end{pmatrix}$$

$$= (x_1 - x_0)(x_2^2 - x_0^2) - (x_1^2 - x_0^2)(x_2 - x_0)$$

$$= (x_1 - x_0)(x_2 - x_0)(x_2 + x_0 - x_1 - x_0)$$

How to solve? GE!

$$= (x_1 - x_0)(x_2 - x_0)(x_2 - x_1) \neq 0$$

B: ? C ?

Mathematical Theory of polynomial interpolation, Existence and uniqueness

Theorem: Give (x_i, y_i) $i = 0, 1, \cdots n$, $x_i \neq x_j$, $i \neq j$, then there is a unique polynomial of degree n (or less) $p_n(x)$ such that satisfying

$$p_n(x_i) = y_i, \quad i = 0, 1, \cdots n.$$

Prove: Set $p_n(x) = a_0 + a_1 x + \cdots + a_n x^n = \sum_{i=0}^{n} a_i x^i$

From $p_n(x_i) = y_i$, $i = 0, 1, \cdots n$.

we get

or
$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & & x_1^n \\ \vdots & \vdots & \vdots & & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

The coefficient matrix A is the Vandermonde matrix whose determinant is

HW #2
$$\det(A) = \prod_{i=1}^{n} \prod_{j=0}^{i-1} (x_i - x_j) = \prod_{0 \le j < i \le n} (x_i - x_j) \ne 0$$

Therefore $\{a_j\}$ exists and unique!

Algorithms (B. C) and analysis (error and efficiency, cost, storage, other issues)

Method 1: Lagrange Interpolation (Use "base polynomials")

$$p_n(x) = l_0(x)\, y_0 + l_1(x)\, y_1 + \cdots + l_n(x)\, y_n$$

polynomial of degreen n; number

Choose $l_i(x)$ in such a way that

$$l_i(x_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \ne j. \end{cases}$$

$l_i(x)$ is a polynomial of degree n.

Construct $l_i(x)$ from their roots

$$l_i(x) =$$

If so, then $p_n(x)$ is a polynomial of degree n.

$$p_n(x_i) = l_0(x_i)y_0 + \dots + l_{i-1}(x_i)y_{i-1} + l_i(x_i)y_i + l_{i+1}(x_i)y_{i+1} + \dots + l_n(x_i)y_n$$

$$= 0 \dots + 1 \cdot y_i + \dots 0 = y_i, \quad i = 0, 1, \dots n.$$

It is what we want!

How to construct $l_i(x)$? we know it is a polynomial of degree n, with roots $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots x_n$. so

$$l_i(x) = A\,(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)$$

$$= A \prod_{\substack{j=0\\ j\neq i}}^{n} (x-x_j), \qquad A \text{ to be determined}$$

$$l_i(x_i) = 1 = A \prod_{\substack{j=0\\ j\neq i}}^{n} (x_i-x_j) \Rightarrow A = \frac{1}{\prod_{\substack{j=0\\ j\neq i}} (x_i-x_j)}$$

$$l_i(x) = \frac{\prod_{\substack{j=0\\ j\neq i}}^{n}(x-x_j)}{\prod_{\substack{j=0\\ j\neq i}}^{n}(x_i-x_j)} = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}$$

$$= \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_n)}$$

$$p_n(x) = l_0(x)y_0 + l_1(x)y_1 + \dots + l_n(x)y_n$$

$$= \sum_{i=0}^{n} l_i(X)\,y_i$$

$$= \sum_{i=0}^{n} \left(\prod_{\substack{j=0\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)} \right) y_i$$

Very compact! One line.

Ex.

x	0	1	4
y	0	1	2

Find $y(3)$.

$$p_2(x) = \text{———} \cdot 0 + l_1(x)\cdot 1 + l_2(x)\, 2$$

$$= \frac{(x-0)(x-4)}{(1-0)(1-4)} + \frac{(x-0)(x-1)}{(4-0)(4-1)}\, 2$$

$$= -\frac{x(x-4)}{3} + \frac{x(x-1)}{6}$$

$$= \frac{x}{6}(7-x). \qquad p_2(3) = \frac{1}{2}\, 4 = 2$$

Programming:

Sum

$$y = \sum_{i=1}^{n} S_i$$

```
y = 0
for i = 1:n
    y = y + S(i)
end
```

product

$$y = \prod S_i$$

```
y = 1
for i = 1:n
    y = y * S(i)
end
```

Input n, (x_i, y_i), t
Return $p_n(t)$

function p = lagrange1(n, x, y, t), $p_n(t)$

```
p = 0
for i = 0:n
    l = 1
    for j = 0:n
        if j ~= i
        l = l * (t - x(j))/(x(i) - x(j))
    end
    end
    p = p + l * y(i)
end
```

Ex: $x = 1:10$; $y = \sin x$; Approximate $\begin{cases} y(3.14) \\ y(\pi/2) \\ y(1.5) \end{cases}$

$p = \text{lagrange1}(10, x, y, \quad)$

Check and analysis, $\begin{cases} \text{increase } n \\ \text{more points} \\ \text{Compare with the exact soln.} \end{cases}$

(i) $y = \sqrt{x}$ $\quad 0 \le x \le 10$

(ii) $y = \dfrac{1}{1+25x^2}$ $\quad -1 \le x \le 1$.

Error Analysis. $\begin{cases} \text{Accuracy ?} \\ \text{Should we use large } n \\ \text{Implementation issues.} \end{cases}$

Define $\underbrace{f(x)}_{\text{unknown}} - p_n(x) \stackrel{\Delta}{=} E_n(x)$, Estimate it

Intuition $f(x) - p_n(x) = g(x)\,(x-x_0)(x-x_1)\cdots(x-x_n)$

Since $f(x_i) - p_n(x_i) = 0$, $\quad g(x) = ?$

Theorem: If $x_i \neq x_j$, $f(x) \in C^{n+1}$, then

$$E_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\,\omega_{n+1}(x)$$

$$\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_n), \quad \text{polynomial of degree } \underline{n+1}.$$
$$= \prod_{i=0}^{n}(x-x_i), \qquad w_0 = 1.$$

Proof:

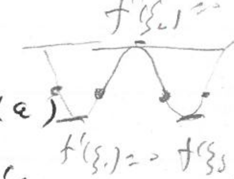
Consider

$$\varphi(t) = f(t) - p_n(t) - \frac{\omega_n(t)}{\omega_n(x)}\left(f(x) - p_n(x)\right)$$

Then $\varphi(x_i) = 0$ since $f(x_i) - p_n(x_i) = 0$

and $\underline{\varphi(x) = 0}$ $\qquad\qquad\qquad$ $\omega_{n+1}(x_i) = 0$

From the Roll's Theorem, $\qquad f(a) = f(a)$



we know that there are $\qquad \exists\, \xi \in (a,b)\ \ f'(\xi) = 0$

<u>n+1 different zeros</u> of $\varphi'(t)$, Using Roll's Th.

again, There are $n-1$ different zeros of $\varphi''(t)$

We can conclude that

	No. of distinct zeros
$\varphi(t)$	$n+2$
$\varphi'(t)$	$n+1$
$\varphi''(t)$	n
$\varphi'''(t)$	$n-1$
$\vdots$	
$\varphi^{n}(t)$	2 $\qquad$ (check with $n=1$)
$\varphi^{n+1}(t)$	1

Therefore there is a point ξ such that

$\varphi^{(n+1)}(\xi) = 0$, that is

$$\varphi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - p_n^{(n+1)}(\xi) - \frac{\omega_{n+1}^{(n+1)}(\xi)}{\omega_{n+1}(x)}\left(f(x) - p_n(x)\right)$$

$$= 0$$

$p_n^{(n+1)}(\xi) = 0$ $\qquad$ Since p_n is a polynomial of degree n, $\quad \omega_{n+1}^{(n+1)}(\xi) = \left(x^{n+1} + (\ )x^n - \cdots\ \right)^{(n+1)}$

$$= (n+1)!$$

$$\Rightarrow \quad f^{(n+1)}(\xi) - 0 - \frac{(n+1)!}{\omega_{n+1}(x)}\left(f(x) - p_n(x)\right)$$

$$\Rightarrow \quad f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\,\omega_{n+1}(x)$$

$\square$

Corollary: If $f(x)$ is a polynomial of degree k, $k \leq n$, then $f(x) \equiv p_n(x)$, that is polynomial interpolation is exact!

Ex ~~What is~~ Show that $\sum_{i=0}^{n} l_i(x) \equiv 1$ consistency

proof: $f(x) \equiv 1$, $\quad y_i = 1$, $\quad f(x) = p_n(x)$

$$f(x) = \sum_{i=1}^{n} l_i(x) y_i = \sum_{i=1}^{n} l_i(x) = 1.$$

Also $\sum_{i=0}^{n} x_i l_i(x) = x$ if $i \geq 1$.

Merits / Dis-advantage of Lagrange polynomial interpolation

- Theoretical useful
- ~~Computational expensive~~ cost can be lower
- Add new node, Need to recompute all.

Solution. Newton formula (similar to Taylor expansion $\quad f(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$

$$p_n(x) = f[x_0] + f[x_0, x_1](x-x_0) + f[x_0, x_1, x_2](x-x_0)(x-x_1)$$

$$+ \cdots + f[x_0, x_1, \cdots, x_n](x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

$$= \sum_{k=0}^{n} f[x_0, x_1, \cdots, x_k]\,\omega_k(x)$$

Divided Differences.

Define: $f[x_0] = f(x_0)$

1-st order divided fifferences Df

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \quad f[x_i, x_j] = \frac{f(x_j) - f(x_i)}{x_j - x_i}$$

$$f[x_i, x_j] = f[x_j, x_i]$$

2-nd order divided differences.

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

$$f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}$$

From k-th to k+1-th

k+1-points

$$f[x_0, x_1, \cdots, x_{k+1}] = \frac{f[x_1, x_2, \cdots, x_{k+1}] - f[x_0, x_1, \cdots x_k]}{x_{k+1} - x_0}$$

Table of divided differences.

x_0	$f[x_0]$	Df	D^2f	$\cdots$
x_1	$f[x_1]$	$f[x_0, x_1]$		
$\vdots$	$\vdots$	$\vdots$	$f[x_0, x_1, x_2]$	
			$\vdots$	
x_n	$f[x_n]$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$\cdots f[x_0, x_1, \cdots, x_n]$
x_{n+1}	$f[x_{n+1}]$	$f[x_n, x_{n+1}]$		

Then the Newton polynomial interpolation is given by

$$p_n^N(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$
$$+ \cdots + f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1})$$
$$= \sum_{k=0}^{n} f[x_0, x_1, \cdots, x_k]\, \omega_k(x)$$

with $\omega_0(x) = 1$

Ex:

0	1	4
0	1	2

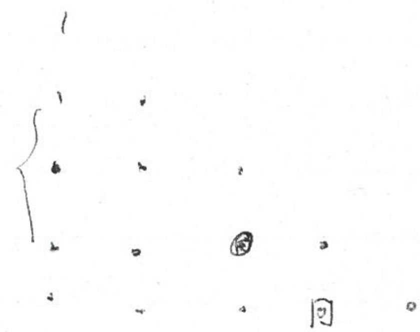
$$\begin{array}{cccc} 0 & 0 & & \\ 1 & 1 & 1 & \\ 4 & 2 & \frac{1}{3} & \frac{\frac{1}{3}-1}{4} \end{array}$$

$$p_2(x) = 0 + (x-0) - \frac{2}{12}(x-0)(x-1)$$

$$= x - \frac{x(x-1)}{6} = \frac{x(7-x)}{6}$$

Switch rows

$$\begin{array}{cccc} 4 & 2 & & \\ 0 & 0 & \frac{-2}{-4} & \\ 1 & 1 & 1 & \frac{\frac{1}{2}}{-3} = -\frac{1}{6} \end{array}$$

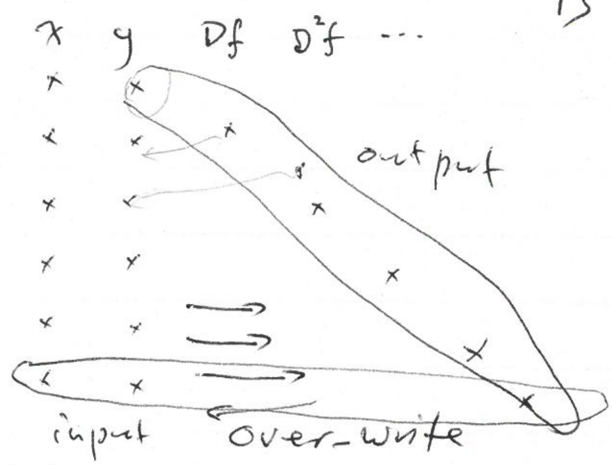


$$p_2(x) = 2 + \frac{1}{2}(x-4) - \frac{1}{6}(x-4)\,x$$

$$= -\frac{x^2}{6} + \frac{7}{6}x = \frac{x}{6}(7-x)$$ the same

$$\begin{array}{cccccc} x_0 & f_0 & Df & D^2 f & D^3 f & D^4 f \\ x_1 & f_1 & \checkmark & & & \\ x_2 & f_2 & \checkmark & \frac{?-?}{x_2-x_0} & & \\ x_3 & f_3 & \checkmark & \frac{?-?}{x_3-x_1} & \frac{?\ ?}{x_3-x_0} & \\ x_4 & f_4 & \checkmark & \frac{?\ ?}{x_4-x_2} & \frac{?\ !}{x_4-x_1} & \frac{?-?}{x_4-x_0} \end{array}$$

Pseudo code.



Need to keep the last ~~column~~ row if new nodes are going to be added.

```
function p = newton(x,y)
    n = length(y);
    p = y;
    for i=2:n                       → Think and try
        for j=n:-1:i                →
            p(j) = (p(j) - p(j-1))/(x(j) - x(j-i+1));
        end
    end.
```

Get coeffi.

Evaluate $$p_n(x) = f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)$$

```
pv = p(1);   n = length(p);   l = 1.
for i=2:n
    l = l*(t - x(i-1))
    pv = pv + l*p(i)
end
```

O(n) operation.

How do we know that $p_n^L(x) \equiv p_n^N(x)$?

properties of divided differences. Uniqueness Thm.

1. Theorem: $f[x_0, x_1, \cdots, x_n] = \sum_{i=0}^{n} \frac{f(x_i)}{\omega_{n+1}'(x_i)}$ (linear combination of fcn values.)

$$\omega_{n+1}(x) = \prod_{j=0}^{n} (x - x_j)$$

$$\omega_{n+1}'(x_i) = \prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j) + 0 + 0 + \cdots + 0$$

n-th

Proof: Induction: $n=1$

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

$$\omega_2'(x_0) = x_0 - x_1, \quad \omega_2'(x_1) = x_1 - x_0$$

$$\sum_{i=0}^{1} \frac{f(x_i)}{\omega_2'(x_i)} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \checkmark$$

Assume it is true for $f[x_0, x_1, \cdots, x_k]$, then

$$f[x_0, x_1, \cdots, x_{k+1}] = \frac{f[x_1, x_2, \cdots, x_{k+1}] - f[x_0, x_1, \cdots, x_k]}{x_{k+1} - x_0}$$

No x_0, No x_{k+1} term,

$$= \frac{\sum_{i=1}^{k+1} \frac{f(x_i)}{\omega_{k+1}'(x_i)} - \sum_{i=0}^{k} \frac{f(x_i)}{\omega_{k+1}'(x_i)}}{x_{k+1} - x_0}$$

$$= \sum_{i=0}^{k+2} \frac{f(x_i)}{\omega_{k+2}'(x_i)} \frac{x_i - x_0 - (x_i - x_{k+1})}{x_{k+1} - x_0} = \sum_{i=0}^{k+1} \frac{f(x_i)}{\omega_{k+2}'(x_i)}$$

2. Divided diffiferences are symmetric $x_i \Leftrightarrow x_j$

3. Similar to derivative properties

$f(x) = \alpha g(x) + \beta h(x)$ then

$$f[x_0, x_1, \cdots x_k] = \alpha g[x_0, x_1, \cdots, x_k] + \beta h[x_0, \cdots, x_n]$$

4. $f(x) = g(x)\, h(x)$

$$f[x_0, x_1, \cdots x_n] = \sum_{j=0}^{n} g[x_0, \cdots x_j]\; h[x_{j+1}, \cdots, x_n]$$

FDM for ODE/PDEs 17A584

5 Relation with finite difference (equally-spaced)

if $x_{i+1} - x_i = h$, for $i = 0, 1, \cdots, n-1$; Define

$$Df(x_i) = \frac{f(x_i+h) - f(x_i)}{h}$$

$$D^2 f(x_i) = \frac{Df(x_i+h) - Df(x_i)}{h} = \frac{\frac{f(x_i+h+h) - f(x_i+h)}{h} - \frac{\cdots}{h}}{h}$$

$$= \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$$

Then: $$f[x_0, x_1, \cdots, x_k] = \frac{D^k f(x_0)}{k!}$$

Equivalence

<u>Theorem</u>: The Lagrange and Newton polynomial interpolation are equivalent.

Proof: We use $P_n^L(x)$ as the Lagrange polynomial interpolation of degree n.

Consider:

$$p_n^L(x) = p_0^L(x) + (p_1^L(x) - p_0^L(x)) + (p_2^L(x) - p_1^L(x)) + \cdots + (p_n^L(x) - p_{n-1}^L(x))$$

Then $p_0^L(x) = f(x_0) = f[x_0]$

$$p_k^L(x) - p_{k-1}^L(x) = f[x_0, x_1, \cdots, x_k]\; \omega_k(x)$$

$p_n^L(x) = f(x_0] + f[x_0, x_1](x-x_0) +$

If true, then $p_n^N(x) \equiv p_n^L(x)$ Since $p_k^L(x) = p_{k-1}^L(x) + f[\ ]\omega_k(x)$ $p_{k-1}^N(x)$

We know that $p_k^L(x) - p_{k-1}^L(x) = A(x-x_0)(x-x_1)\cdots(x-x_{k-1})$
$= A\,\omega_k(x)$

Find A; by plugging in x_k

Since $p_k^L(x_j) = p_{k-1}^L(x_j)$

$$\underset{f(x_k)}{\overset{\shortparallel}{p_k^L(x_k)}} - p_{k-1}^L(x_k) = A\,\omega_k(x_k)$$

$\prod \dfrac{x-x}{}$

$$A = \frac{f(x_k) - p_{k-1}^L(x_k)}{\omega_k(x_k)}$$

$$= \frac{f(x_k) - \sum_{j=0}^{k-1} l_j(x_k) f(x_j)}{\omega_k(x_k)}$$

$\dfrac{1}{x_j - x_k} = -\dfrac{1}{x_k - x_j}$

x_{k-1}

$$\frac{\dfrac{\cancel{(x_k-x_0)}(x_k-x_1)\cdots(x_k-x_k)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_k)}}{\cancel{(x_k-x_0)}(x_k-x_1)\cdots(x_k-x_j)}$$

$(k-1$

$$\overset{\cdots}{=}$$

$$A = \sum_{j=0}^{k} \frac{f(x_j)}{\omega_{k+1}'(x_j)} = f[x_0, x_1, \ldots, x_k]$$

Since

$$p_n(x) = p_n^L(x) = p_n^N(x) = f[x_0] + f[x_0, x_1](x-x_0) + \cdots + f[x_0, x_1, \ldots, x_n]\,\omega_n(x) = P_n^N(x)$$

From the equivalence theorem and the error estimates we have the following corollary:

Corollary: $f[x_0, x_1, \ldots, x_n, x] = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}$

Proof: We know that

$$p_n(t) = p_n^L(t) = p_n^N(t) = f[x_0] + f[x_0, x_1](\cancel{t} - x_0) + f[x_0, x_1, x_2](t-t_0)(t-t_1)$$

$$+ \cdots + f[x_0, x_1, \cdots, x_n] \, \overset{(t-x_0)(t-x_1)\cdots(t-x_{n-1})}{\omega_{n}(t)}$$

and $$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

If we add a node x to $x_0, x_1, \cdots, x_n$, $(x, f(x))$ then we have

$$p_{n+1}(t) = p_n(t) + f[x_0, x_1, \ldots, x_n, x](t-x_0)(t-x_1)\cdots(t-x_n)$$

plug x in, we have

$$p_{n+1}(x) = p_n(x) + f[x_0, x_1, \ldots, x_n, x]\,\omega_{n+1}(x)$$

From the interpolation, we have $p_{n+1}(x) = f(x)$,

$$f(x) = p_n(x) + f[x_0, x_1, \ldots, x_n, x]\,\omega_{n+1}(x)$$

From the error estimate, we have

$$f(x) = p_n(x) + \underline{\underline{R_n(x)}} = p_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

$$\Rightarrow \quad f[x_0, x_1, \cdots, x_n, x]\,\omega_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

$$\Rightarrow \quad \boxed{f[x_0, x_1, \cdots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}}$$

Hermite Interpolation. Match both the solution and the derivative at nodal points.

Given (x_i, y_i, y_i') $i = 0, 1, \cdots n$, find $\quad \begin{matrix} n+1 \\ n+1 \\ \hline 2n+2 \quad -1 \end{matrix}$

$P_{2n+1}(x)$ such that $P_{2n+1}(x_i) = y_i$, $P_{2n+1}'(x_i) = y_i'$

$i = 0, 1, \cdots n$.

Method A. Basis polynomial. Method B. Divided differences.

Method A.

Set: $p_{2n+1}(x) = \sum_{i=0}^{n} \bar{l}_i(x) y_i + \sum_{i=0}^{n} \bar{\bar{l}}_i(x) y_i'$

$\bar{l}_i(x)$ is a polynomial of degree $2n+1$

Kronecker $\begin{cases} \bar{l}_i(x_j) = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \\ \bar{l}_i'(x_j) = 0 \end{cases}$ $\qquad \bar{\bar{l}}_i(x_j) = 0$, $\quad \bar{\bar{l}}_i'(x_j) = \delta_{ij} = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$

From the root factorization, we know that

$$\bar{l}_i(x) = \underbrace{l_i^2(x)}_{\text{degree } 2n}(ax+b) \qquad \bar{l}_i(x_j) = 0, \ \bar{l}_i'(x_j) = 0$$

Note: $l_i(x_i) = 1$

$$\bar{l}_i(x_i) = l_i^2(x_i)(ax_i+b) = 1 \quad \Rightarrow \quad ax_i+b = 1$$

$$\bar{l}_i'(x_i) = 2l_i(x_i)\,l_i'(x_i)\,(\overset{=1}{ax_i+b}) + l_i^2(x_i)\,a$$

$$= 2l_i'(x_i) + a = 0 \quad \Rightarrow \quad a = -2l_i'(x_i)$$

$$b = 1 - ax_i = 1 + 2l_i'(x_i)\,x_i, \quad i = 0, 1, \ldots$$

Similarly

Let $\bar{\bar{l}}_i(x) = l_i^2(x)(cx+d)$

Can set $c = 1$, $d = -x_i$.

In one line, $p_{2n+1}(x) = \sum_{i=0}^{n} \left[y_i + (x-x_i)\left(y_i' - 2y_i l_i'(x_i)\right)\right] l_i^2(x)$

Error estimate $R_{2n+1}(x) = f(x) - p_{2n+1}(x) = \dfrac{f^{(2n+2)}(\xi(x))}{(n+2)!}\,\omega_{n+1}^2(x)$

Can you prove it?

Method B:

Divided difference

$$
\begin{array}{llll}
x_0 & y_0 & & \\
x_0 & y_0 & y_0' & \\
x_1 & y_1 & f[x_0,x_1] & ? \\
x_1 & y_1 & y_1' & - - - \\
x_2 & y_2 & f[x_1,x_2] & - - \\
x_2 & y_2 & y_2' & - - - \\
- & - & - - - & - - -
\end{array}
$$

$$f[x_0, x_1 \cdots x_k] = \frac{f^{(k)}(\xi)}{k!}$$

Higher derivatives

un-determined coefficient method may be difficult.

Ex: Let $f(0)=1$, $f'(0)=-1$,
$f(1)=1$, $f'(1)=-1$, $f''(1)=2$

Find polynomial interpolation $p_4(x)$ using the divided differences.

$$
\begin{array}{cccccc}
0 & 1 & & & & \\
0 & 1 & -1 & & & \\
1 & 1 & 0 & 1 & & \\
1 & 1 & -1 & -1 & -2 & \\
1 & 1 & -1 & \boxed{\frac{2}{2}}\ f'' & 2 & 4
\end{array}
$$

$$f[x_0, x_1, \cdots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Take $n=1$, $f[x_0, x_1, x] = \dfrac{f^{(2)}(\xi)}{2!}$

Therefore
$$p_4(x) = 1 - x + x^2 - 2x^2(x-1) + 4x^2(x-1)^2$$
$$= 1 - x + x^2 - 2x^3 + 2x^2 + 4x^4 - 8x^3 + 4x^2$$
$$= 1 - x + 7x^2 - 10x^3 + 4x^4$$

$p_4(0) = 1$, $\quad p_4(1) = +1$

$$p_4'(x) = -1 + 14x - 30x^2 + 16x^3$$

$p_4'(0) = -1$, $\quad p_4'(1) = -1$,

$p_4''(x) = 14 - 60x + 48x^2 \quad p_4''(1) = 14 - 60 + 48 = 2$

Piecewise polynomial interpolation (in one-dimension)
linear, quadratic, cubic, splines

Motivation: • Better accuracy — Select $\{x_i\}$'s, orthogonal polynomials

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} w_n(x) \qquad \lim_{n\to 0} R_n(x) \not\Rightarrow 0$$

$f^{(n+1)}(x)$ may get very large

$w_{n+1}(x)$ - - - - - - - especially near the ends,

- Avoid Runge phenomenon

• localized computation (FEM)

• polynomial interpolation with large n is unstable

$(x_i, y_i) \to P_n(x)$

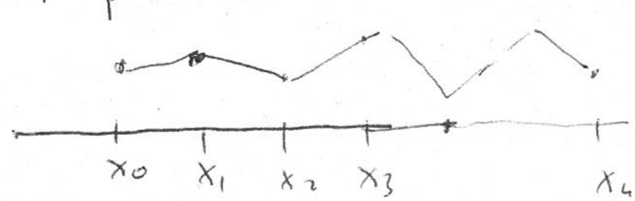
$(x_i, \tilde{y}_i) \to \tilde{P}_n(x) \qquad \|P_n(x) - \tilde{P}_n(x)\| \le \frac{2^{n+1}}{e n \log n} \varepsilon$

$\|y_i - \tilde{y}_i\| \le \varepsilon$,

- Local to global

$e = 2.7183$ naeperian number

Piecewise linear interpolation:



Require continuity but not smoothness $u_h^I(x)'$ may not exist at nodal points.

Construct piecewise linear interpolation function.

A Direct approach

$$u_h^I(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} y_i + \frac{x - x_i}{x_{i+1} - x_i} y_{i+1} \qquad x_i \le x \le x_{i+1},$$

$$h = \max_{0 \le i \le n-1} \{ |x_{i+1} - x_i| \}$$

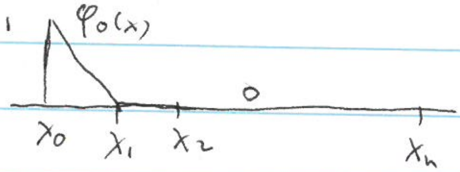
B. Use local basis function $\qquad V_h \Longleftrightarrow R^{n+1}$

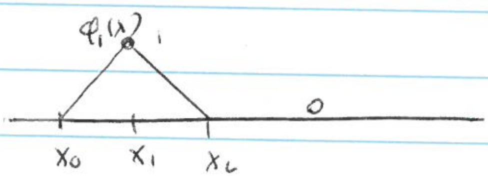
$$u_h^I(x) = \sum_{i=0}^{n} \varphi_i(x) y_i,$$

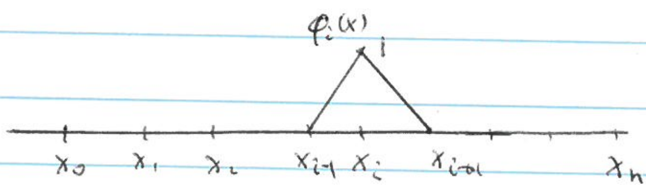
$\varphi_i(x)$ is a piecewise continuous linear function satisfying

$$\varphi_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \ne j. \end{cases}$$

They are called the hat functions.







$$\varphi_i(x) = \begin{cases} 0 & < x \le x_{i-1} \\ \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & x_{i-1} \le x \le x_i \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i} & x_i \le x \le x_{i+1} \\ 0 & \end{cases}$$

Error estimate: Theorem:

If $u(x) \in C^2[a,b]$, i.e. $u(x)$ has up to 2-nd order continuous derivatives in $[a,b]$, then

$$\| u(x) - u_h^I(x) \|_\infty \le C h^2 \| u'' \|_\infty \qquad C = \frac{1}{8}$$

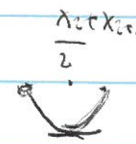
where $\quad \| u'' \|_\infty = \max_{a \le x \le b} | u''(x) |$

$$\| (u(x) - u_h^I(x))' \|_{L^2(a,b)} \le \bar{C} h \| u'' \|_\infty$$

$$\| u(x) \|_{L^2(a,b)} = \sqrt{\int_a^b |u|^2 \, dx}$$

Proof for the 1-st inequality

If $x \in [x_i, x_{i+1}]$, we know that



Why? $\quad u(x) - u_h^I(x) = \frac{u''(\xi)}{2} (x - x_i)(x - x_{i+1})$

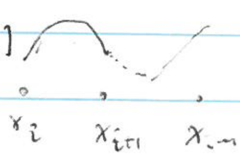
$$| u(x) - u_h^I(x) | \le \frac{1}{2} \| u'' \|_\infty \left(\frac{x_{i+1} - x_i}{2} \right)^2$$

$$\le \frac{h^2}{8} \| u'' \|_\infty \qquad C = \frac{1}{8}.$$

Note that $\lim_{h \to 0} \| u(x) - u_h^I(x) \|_\infty = 0$, if $u(x) \in C^2(a,b]$, we call it second order accurate interpolation. No derivative at nodal points.

- Compact support of the basis function, zero almost everywhere.

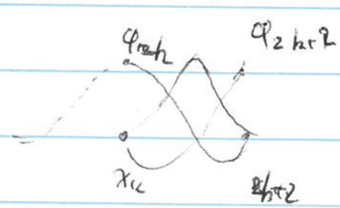
<u>Piecewise quadratic:</u> in $C[x_0, x_n]$



$\varphi_i(x_j) = \delta_{ij}$

$\varphi_{2k}(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \ne j \end{cases}$

$\varphi_{2k+1}(x_j) = 0$, Add a mid point



$\varphi_{2i+1}\left(\frac{x_i+x_{i+1}}{2}\right)=1$, $\quad \varphi_{2i}(x_j)=0$ — continuous but not differentiable at nodes.

$$u_h^I(x) = \sum_{j=0}^{n} \alpha_{2j}\varphi_{2j}(x) + \sum_{j=0}^{n-1}\alpha_{2j+1}\varphi_{2j+1}(x),$$

→ Type II: Require continuous derivative.

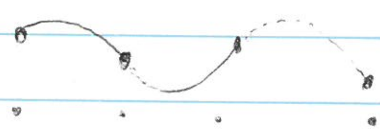
Spline interpolation: Piecewise polynomial

Interpolates $f(x)$

has certain smoothness (derivatives) continuous

piecewise quadratic is C^1: seldom used, why?

Cubic splines: $S(x)$
- Piecewise cubic function
- Interpolates $f(x)$
- has continuous 0-th, 1-st, 2-nd order derivatives. C^2

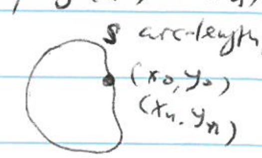


Need additional boundary condition at x_0 and x_n.

DOF.

Boundary Conditions: why?

1. A Natural Splines: $S''(x_0)=0$, $S''(x_n)=0$
2. A clamped spline: $S'(x_0)$ and $S'(x_n)$ are specified
3. A periodic spline: $S(x_0)=S(x_n)$, $S'(x_0)=S'(x_n)$, $S''(x_0)=S''(x_n)$



4. Mixed of above.

Derivation of cubic spline interpolation $S_3(x)$.

(primary) cardinal spline, global

$S_3(x)$ is a piecewise cubic function

$S_3'(x)$ is " quadratic "

$S_3''(x)$ is " linear function. $C^2[x_0, x_n]$

(importance or effect)
(in influence

Set $S_3''(x_j) = M_j$, $j = 0, 1, \cdots, n$. (they are called the moments), then we know

x_{j-1} x_j

$$S_3''(x) = M_{j-1} \frac{x - x_j}{x_{j-1} - x_j} + M_j \frac{x - x_{j-1}}{x_j - x_{j-1}} \qquad x_{j-1} \le x \le x_j$$

Determined by $\{y_j\}_{j=1}^{n-1}$ + BC's.

Let $h_j = x_j - x_{j-1} > 0$, then

$$S_3''(x) = M_{j-1} \frac{x_j - x}{h_j} + M_j \frac{x - x_{j-1}}{h_j} \qquad x_{j-1} \le x \le x_j$$

Integrate once to get

$$S_3'(x) = -M_{j-1} \frac{(x_j - x)^2}{2h_j} + M_j \frac{(x - x_{j-1})^2}{2h_j} + C_{j-1}$$

Integrate one more time to get

$$S_3(x) = M_{j-1} \frac{(x_j - x)^3}{6h_j} + \frac{M_j (x - x_{j-1})^3}{6h_j} + C_{j-1}(x - x_{j-1}) + \tilde{C}_{j-1}$$

We use the interpolation condition to determine the constant

$$S_3(x_{j-1}) = y_{j-1}, \qquad S_3(x_j) = y_j$$

$$M_{j-1} \frac{h_j^2}{6} + \tilde{C}_{j-1} = y_{j-1}, \qquad \frac{M_j h_j^2}{6} + C_j h_j + \tilde{C}_j = y_j$$

use (x_{j-1}, x_j) use (x_j, x_{j+1})

$$\tilde{c}_{j-1} = y_{j-1} - \frac{M_{j-1}}{6} h_j^2, \qquad c_{j-1} = \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(M_j - M_{j-1})$$

or

$$S_3(x) = M_{j-1}\frac{(x_j - x)^3}{6h_j} + \frac{M_j (x - x_{j-1})^3}{6h_j} + (x - x_{j-1})\frac{y_j - y_{j-1}}{h_j}$$

$$- \frac{(x - x_{j-1}) h_j}{6}(M_j - M_{j-1}) + y_{j-1} - \frac{M_{j-1}}{6} h_j^2 \qquad x_{j-1} \le x \le x_j$$

Use the continuity condition to determine a system of equations for $\{M_j\}$.

$$S_3'(x_j-) = S_3'(x_j+)$$

$$S_3'(x) = \begin{cases} -M_{j-1}\dfrac{(x_j - x)^2}{2h_j} + M_j \dfrac{(x - x_{j-1})^2}{2h_j} + \dfrac{y_j - y_{j-1}}{h_j} - \dfrac{h_j}{6}(M_j - M_{j-1}) & x_{j-1} \le x \le x_j \\ -M_j \dfrac{(x_{j+1} - x)^2}{2h_{j+1}} + M_{j+1}\dfrac{(x - x_j)^2}{2h_{j+1}} + \dfrac{y_{j+1} - y_j}{h_{j+1}} - \dfrac{h_{j+1}}{6}(M_{j+1} - M_j) & \end{cases}$$

$\Rightarrow$

$$\frac{M_j}{2} h_j + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(M_j - M_{j-1}) = -\frac{M_j}{2} h_{j+1} + \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{h_{j+1}}{6}(M_{j+1} - M_j)$$

$$(\quad) M_{j-1} + (\quad) M_j + (\quad) M_{j+1} = \frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j}$$

$$\frac{h_j}{6} M_{j-1} + \frac{2(h_j + h_{j+1})}{6} M_j + \frac{h_{j+1}}{6} M_{j+1} = d_j \qquad j = 1, 2, \dots, n-1.$$

or

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = d_j, \quad j = 1, 2, \dots, n-1$$

$$\mu_j = \frac{h_j}{h_j + h_{j+1}} \qquad 0 < \mu_j < 1, \qquad 0 < \lambda_j = \frac{h_{j+1}}{h_{j+1} + h_j} < 1$$

$\lambda_j + \mu_j = 1 < 2$, strictly diagonally dominant.

$$d_j = \frac{6}{h_j + h_{j+1}} \left(\frac{y_{j+1} - y_j}{h_{j+1}} - \frac{y_j - y_{j-1}}{h_j} \right)$$

For the natural spline, we have $M_0 = M_n = 0$

The system of equations is

$$\begin{bmatrix} 2 & \lambda_1 & & 0 \\ \mu_2 & 2 & \lambda_2 & \\ & \ddots & \ddots & \lambda_{n-2} \\ 0 & & \mu_{n-1} & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \end{bmatrix}$$

A clamped spline $S'(x_0) = y_0'$

$S'(x_n) = y_n'$

We know that

$$S_j'(x) = -M_{j-1} \frac{(x_j - x)^2}{2h_j} + M_j \frac{(x - x_{j-1})^2}{2h_j} + \frac{y_j - y_{j-1}}{h_j} - \frac{h_j}{6}(M_j - M_{j-1})$$

Take $j = 1$ and $x = x_0$, we have

$$S_1'(x_0) = -M_0 \frac{h_1}{2} + \frac{y_1 - y_0}{h_1} - \frac{h_1}{6}(M_1 - M_0) = y_0'$$

$$\Rightarrow \quad -\frac{2h_1}{6} M_0 - \frac{h_1}{6} M_1 = y_0' - \frac{y_1 - y_0}{h_1}$$

(*) That is $2M_0 + M_1 = \frac{6}{h_1}\left(\frac{y_1 - y_0}{h_1} - y_0' \right)$

Similarly

(**) $M_{N-1} + 2M_n = \frac{6}{h_n}\left(y_n' - \frac{y_n - y_{n-1}}{h_n} \right)$

- Periodic BC. Homework.

Properties of splines.

(i) The cubic spline exists and it is unique for the three types of boundary conditions.

$x_0 = a \quad x_n = b$

(ii) minimum property (smoothness). If $f(x) \in C^2[a,b]$, $S_{3,f}$ is the cubic spline interpolation, $(S_{3,f}(x_i) = f_i)$ then

$$\int_a^b |S_{3,f}''|^2\,dx \le \int_a^b |f''|^2\,dx, \quad \text{"="} \text{ ~~if and~~ only if } f \equiv S_{3,f}.$$

(iii) Best approximation property

minimum curvature

$$\int_a^b |f'' - S_{3,f}''|^2\,dx \le \int_a^b |f'' - S_3''|^2\,dx$$

Do not need $S_3(x_i) = f(x_i)$

Where $S_3''(x)$ is a piecewise cubic ~~that~~ satisfies the same BC.

Proof of (ii) and (iii) for natural spline BC, $S''(a) = S''(b) = 0$.

Let $S^* = S_{3,f}$ if there is no confusion

$$\int_a^b (f'' - S_*'')^2\,dx = \int_a^b \{|f''|^2 - 2f''S_*'' + (S_*'')^2\}\,dx$$

$$= \int_a^b \{f''^2 - 2(f'' - S_*'')S'' - |S_*''|^2\}\,dx \quad \ge 0$$

So if $\int_a^b (f'' - S'')S_*''\,dx = 0$, then and $\int_a^b f''^2\,dx \ge \int |S''|^2\,dx$ ✓

$$\int_a^b (f'' - S_*'')^2\,dx = \int_a^b |f''|^2\,dx - \int_a^b |S_*''|^2\,dx \le \int_a^b |f''|^2\,dx$$

Lemma

$$\sum \int_{x_i}^{x_{i+1}} g'\,dx = \sum g\Big|_{x_i}^{x_{i+1}} = g(b) - g(a)$$

$$\int_a^b (f'' - S'')S''\,dx = (f' - S')S''\Big|_a^b - \int_a^b (f' - S')S'''\,dx$$

$$= (f' - S')S''\Big|_a^b - (f - S)S'''\Big|_a^b + \int_a^b (f - S)S^{(4)}\,dx$$

$$= (f' - S')S''\Big|_a^b - 0 + 0$$

interpolation $\quad S^{(4)}(x) \equiv 0$

True for natural and clamped if $f'(a) = S'$, $f'(b) = S'(b)$

It may only make sense in each interval.

not needed

$$\sum_{i=1}^{n} (f'-s')s''\Big|_{x_{i-1}}^{x_i} = (f'(x_1)-s'(x_1))s''(x_1)$$
$$- (f'(x_0)-s'(x_0))s''(x_0) + \cdots + (f'(x_n)-s'(x_n))s''(x_n)$$
$$- (f'(x_{n-1})-s'(x_{n-1}))s''(x_{n-1})$$
$$= (f'(b)-s'(b))s''(b) - (f'(a)-s'(a))s''(a) = 0$$

(iii) Natural spline $S_{3,f}''(a) = S_3''(a) = S_{3,f}''(b) = S_3''(b) = 0$

$$\int_a^b |f''-S_3''|^2 dx = \int_a^b |f''-S_3''+S_{3,f}''-S_{3,f}''|^2 dx$$

Any

$$= \int_a^b \left\{ |f''-S_{3,f}''|^2 - 2(S_3''-S_{3,f}'')(f''-S_{3,f}'') + |S_3''-S_{3,f}''|^2 \right\} dx$$

If this term is zero, then

$$= \int_a^b \left\{ |f''-S_{3,f}''|^2 + |S_3''-S_{3,f}''|^2 \right\} dx \geq \int_a^b |f''-S_{3,f}''|^2 dx$$

$$\int_a^b (S_3''-S_{3,f}'')(f''-S_{3,f}'')dx = (f'-S_{3,f}')(S_3''-S_{3,f}'')\Big|_b^a$$
$$- \int_a^b (f'-S_{3,f}')(S_3''-S_{3,f}'')dx$$

$= \sum g(x_i)$

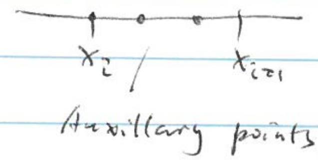
$= f(x_n) - f(x_0)$

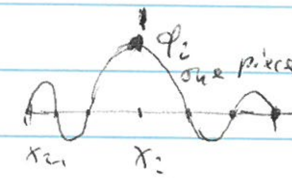
$$= \sum_{i=1}^{n} \left\{ (f'-S_{3,f}')(S_3''-S_{3,f}'') - (f-S_{3,f})(S_3'''-S_{3,f}''') \right\}\Big|_{x_{i-1}}^{x_i}$$
$$+ \int_a^b (f-S_{3,f})(S_3^{(4)}-S_{3,f}^{(4)})dx$$
$$= (f'(b)-S_{3,f}'(b))(S_3''(b)-S_{3,f}''(b)) - (f'(a)-S_{3,f}'(a))$$
$$\cdot (S_3''(a)-S_{3,f}''(a))$$
$$= 0.$$

– Splines using localized basis function.

piecewise cubic interpolation (Not necessarily smooth) in C^1

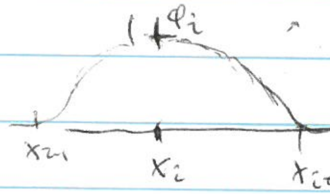
A. continuous only





B. continuity + 1-st derivative
Hermite interpolation.

First



$$\begin{cases} \varphi_i(x_j) = \delta_{ij} \\ \varphi_i'(x_j) = 0 \end{cases}$$

$$\begin{cases} \psi_i(x_i) = 0 \\ \psi_i'(x_j) = \delta_{ij} \end{cases} \qquad u_h^I(x) = \sum_i \alpha_i \varphi_i(x) + \sum \bar{\alpha}_i \psi_i(x)$$

2-nd derivative is not continuous.

__Splines.__ piecewise cubic, continuity of 0-th, 1-st, 2-nd derivatives. in C^2

B-splines

A. Has been derived

Not interpolated

B. Local support. Cardinal splines

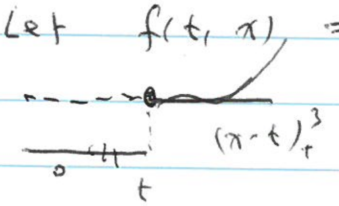
$\varphi_i(x_j) = \delta_{ij}$, $\varphi_i'(x_j)$, $\varphi_i''(x_j)$ are continuous, can be obtained using the method A.

C. B-spline (support, 9-nodes)

- small support, non-negative (better stability)
- Does not satisfy $\varphi_i(x_j) = \delta_{ij}$.
  Still $\varphi_i(x)$, $\varphi_i'(x)$, $\varphi_i''(x)$ are continuous.

Construct B-spline basis.

Let $f(t, x) = (x-t)_+^3 = \begin{cases} (x-t)^3, & x \geq t \\ 0 & x < 0 \end{cases}$



Expand the nodal points

$$x_0, x_1, \cdots, x_n$$

$$\underbrace{x_{-3}\ x_{-2}\ x_{-1},}_{\text{One or several}} \qquad\qquad , x_{n+1}, x_{n+2}, x_{n+3}$$

Can be the same as x_0 (repeated knots)

For $i = -3, -2, -1, \cdots, n-2, n-1$, we can define the basis function:

$f(t,x) = (t-x)_+^3 = \begin{cases} (t-x)^3, & x \le t \\ 0 \end{cases}$ with t variable

$$N_i(x) = (x_{i+4} - x_i)\, f[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}],$$

with x fixed. (implicit parameter)

Where $f[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}]$ is defined as the divided difference. For repeated knots, we use the derivative(s).

$$f[x_0, x_0] = f'(x_0), \qquad f[x_0, x_0, x_0] = \frac{f''(x_0)}{2!}$$

If all nodes are distinct, then from

$$f[x_0, x_1, \cdots, x_n] = \sum_{j=0}^{n} \frac{f(x_j)}{\omega_{n+1}'(x_j)}, \quad \text{we get}$$

$$N_i(x) = (x_{i+4} - x_i) \sum_{j=i}^{i+4} \frac{(x - x_j)_+^3}{\prod\limits_{\substack{l=i \\ l \ne j}}^{i+4} (x_j - x_l)} \qquad \in C^2$$

If not, often on boundary, use table.

We can use this to programming.

<u>Theorem</u>: $N_i(x) \equiv 0$ if $x < x_i$ or $x > x_{i+4}$.

$N_i(x) \ge 0$.

Proof: If $x < x_i$, then $(x - x_j)_+^3 = 0$ according to the definition for $j = i, i+1, \cdots, i+4$.

/

<u>Example:</u> $x_i = i, \quad i = 0, 1, \cdots$

$0, 1, 2, 3, 4$

$$N_8(x) = (4-0) \begin{cases} 0 & x < 0 \\ \dfrac{x^3}{(0-1)(0-2)(0-3)(0-4)} & 0 \le x < 1 \\ \dfrac{x^3}{4!} + \dfrac{(x-1)^3}{(1-0)(1-2)(1-3)(1-4)} & 1 \le x < 2 \\ \dfrac{x^3}{4!} + \dfrac{(x-1)^3}{(-3!)} + \dfrac{(x-2)^3}{(2-0)(2-1)(2-3)(2-4)} & 2 \le x < 3 \\ \dfrac{x^3}{4!} - \dfrac{(x-1)^3}{3!} + \dfrac{(x-2)^3}{4} + \dfrac{(x-3)^3}{(3-0)(3-1)(3-2)(3-4)} & 3 \le x < 4 \\ 0 & \end{cases}$$

$N_0(x)$

If $x > x_{i+4}$, then $(x-x_j)_+^3 = (x-x_j)^3$, for $j = i, \dots, i+4$, therefore

$$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}] = \frac{\frac{\partial^4 (x-t)^3}{\partial t}\Big|_{t=\xi}}{4!} \equiv 0.$$

$$= \frac{f^{(4)}(\xi)}{4!}.$$

$N_i(x) \geq 0$ is more difficult to prove. We can see from the plots of several basis functions.

Use the basis function (B-splines) to get cubic spline interpolation

$$S_3(x) = \sum_{i=-3}^{n-1} \alpha_i N_i(x), \qquad \text{or } \sum_i \qquad \overset{x_0}{a} \leq x \leq \overset{x_n}{b}$$

There are $n+3$ unknowns. $(n-1) - (-3) + 1$ $n-1+3+1$

$S_3(x_i) = f_i$, $i = 0, 1, \dots n$ $\qquad n+1$

Two more conditions are needed.

A. $S_3''(x_0)$, $S''(x_n)$ are given triple nodes

B. $S'(x_0)$, $S'(x_n)$ are given.

C. $S'(x_0) = S'(x_n)$, $S''(x_0) = S''(x_n)$.

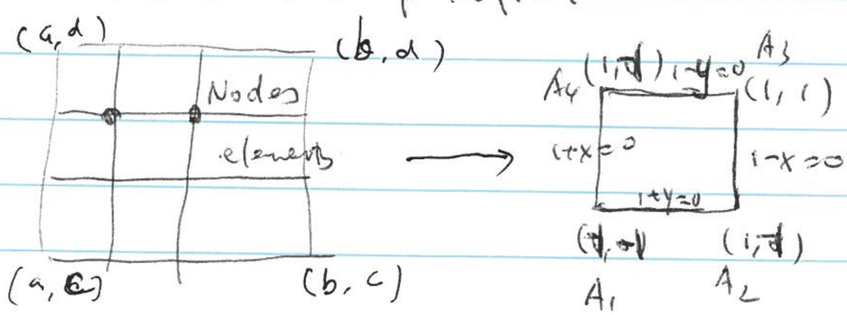
Can also be generated from a recursive relation

Two dimensional Interpolation (polynomial)

Use basis functions.

· Bi-linear interpolation

How:



Recursive relation for B-splines   Bernstein

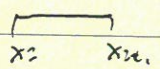
local basis

$$B_{i,1}(x) = \begin{cases} 1 & x \in [x_i, x_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

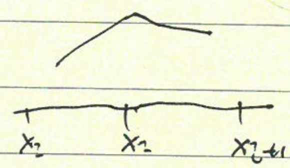
$$B_{i,k+1}(x) = \frac{x - x_i}{x_{i+k} - x_i} B_{i,k}(x) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_{i+1}} B_{i+1,k}(x),$$

$B_{i,2} \to B_{i,3} \to B_{i,4}$

spline



$B_{i,1}$



$B_{i,2}$

. . .

Preferred if we evaluate B-spline at a given point.

Problem: Given (x_i, y_i), $i = 0, 1, \dots n$, find piecewise polynomials in C^0 / C^1 to interpolate the data, $u_h^I(x)$, $u_h^I(x_i) = y_i$, and related study. {wellposedness, convergency etc, implem.}

C^0, P_1 elements, $(pu')' = f$

P_2

$\left\{ \begin{array}{l} \text{linear} \\ \text{quadratic} \\ \text{cubic} \end{array} \right.$ for different applicate.

C^1 $u_{xxxx} = f$

$\varphi_1(x,y) = 0$ along A_2A_3, A_3A_4, therefore

$$\varphi_1(x,y) = C\,(1-x)(1-y)$$

$$\varphi_1(-1,-1) = C\;2\cdot 2 = 1 \quad \Rightarrow \quad C = \frac{1}{4}$$

$$\begin{cases} \varphi_1(x,y) = \frac{1}{4}(1-x)(1-y) \\ \varphi_2(x,y) = \frac{1}{4}(1+x)(1-y) \\ \varphi_3(x,y) = \frac{1}{4}(1+x)(1+y) \\ \varphi_4(x,y) = \frac{1}{4}(1-x)(1+y) \end{cases} \qquad \varphi_i(x,y) = \frac{(1+x_i x)(1+y_i y)}{4}$$

Interpolation $\quad u(x,y) = \sum_{i=1}^{4} u(x_i, y_i)\,\frac{(1+x_i x)(1+y_i y)}{4}$

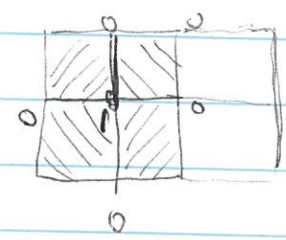
If $x \in [x_A, x_B]$, then

x_A	-1
x_B	1

$$\tilde{x} = \frac{x - x_B}{x_A - x_B}(-1) + \frac{x - x_A}{x_B - x_A}(1)$$

$$= \frac{2x - x_B - x_A}{x_B - x_A} \qquad \text{Similarly} \quad \tilde{y} = \frac{2y - x_c - x_D}{y_D - y_C}$$

Continuity



On each side of the edge,

$\varphi_i(x,y)$ is a linear function which is uniquely determined by the function values at two ends. So we have ~~the~~ continuity but the first order derivative is discontinuous.

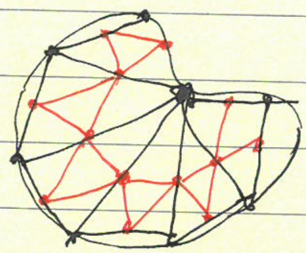
2D and 3D interpolation

Domain —— Discretize —— Mesh

structured often for regular domain

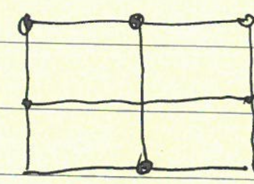
unstructured often for complicated domain

$\Omega \to \Omega_h.$



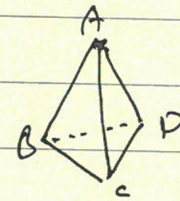
refine
Middle
point rule

structured



A fixed patter between subdomains and nodal points, e.g. Cartesian

3D:
tetrahedron



Linear interpolation over triangle mesh $\{\vec{x}_k\}_{k=1}^{DoF} = \{(x_k, y_k)\}_{k=1}^{DOF}$

Thm: A piecewise linear function in $C(\Omega_h)$ is uniquely determined by its values at nodal points $\{\vec{x}_k\}$.

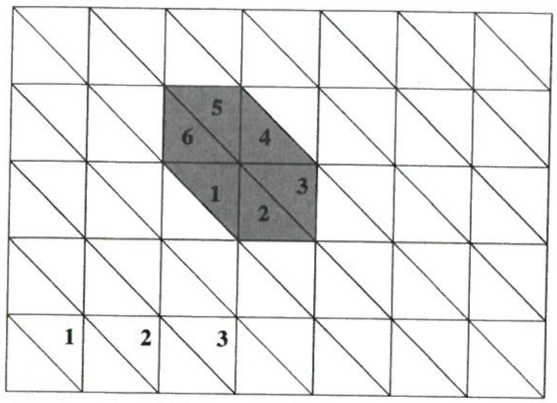
(Local) Basis function $\varphi_k(x)$

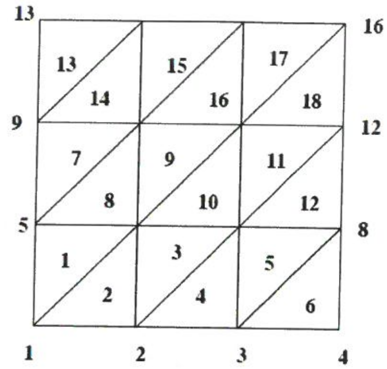
$$\varphi_k(\vec{x}_j) = \delta_i^j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{otherwise} \end{cases}$$

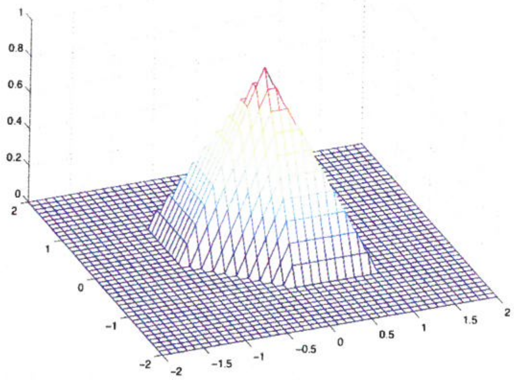
A tent function without a door.

Then

$$u_h(x) = \sum_{k=1}^{DoF} f(\vec{x}_k)\,\varphi_k(\vec{x}) \qquad \| u_h(\vec{x}) - f(\vec{x}) \|_\infty \le Ch^2 \|D^2 f\|_\infty$$



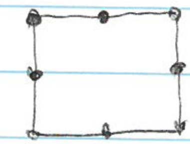




$$
\phi_{j(n-1)+i} = \begin{cases} \dfrac{x-(i-1)h+y-(j-1)h}{h} - 1 & \text{Region 1} \\ \dfrac{y-(j-1)h}{h} & \text{Region 2} \\ \dfrac{h-(x-ih)}{h} & \text{Region 3} \\ 1 - \dfrac{x-ih+y-jh}{h} & \text{Region 4} \\ \dfrac{h-(y-jh)}{h} & \text{Region 5} \\ \dfrac{x-(i-1)h}{h} & \text{Region 6} \\ 0 & \text{otherwise}\,. \end{cases}
$$

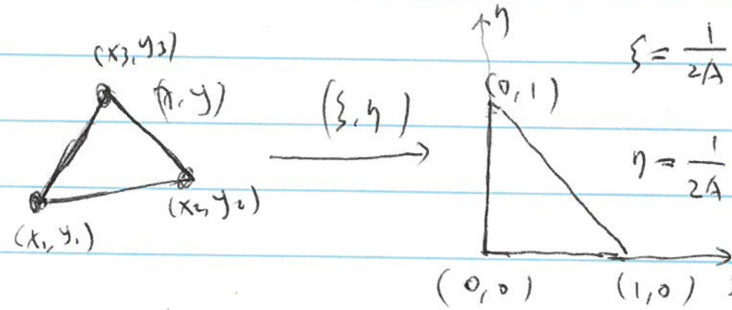
bi-quadratic (quadratic on each side)

$$Q_2(x,y) = a_1 + a_2 x + a_3 y + a_4 xy + a_5 x^2 + a_6 y^2 + a_7 x^2 y + a_8 x y^2$$



4 auxiliary points

Triangles

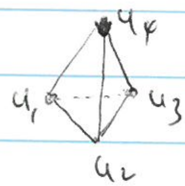


$$\xi = \frac{1}{2A}\left[(y_3 - y_1)(x - x_1) - (x_3 - x_1)(y - y_1)\right]$$

$$\eta = \frac{1}{2A}\left[-(y_2 - y_1)(x - x_1) + (x_2 - x_1)(y - y_1)\right]$$

$$\varphi_1(\xi,\eta) = 1 - \xi - \eta, \quad \varphi_2(\xi,\eta) = \xi, \quad \varphi_3(\xi,\eta) = \eta$$

3D.



$$u(x,y,z) = a + bx + cy + dz.$$

Trigonometric - Interpolation (FFT) later.

tetrahedron