

Chương 3: Đường Cong Khớp

Trịnh Anh Phúc, Vũ Văn Thiệu, Đinh Viết Sang, Nguyễn Đức Nghĩa¹

¹Bộ môn Khoa Học Máy Tính, Viện CNTT & TT,
Trường Đại Học Bách Khoa Hà Nội.

Ngày 23 tháng 3 năm 2015

1 Đặt vấn đề

2 Nội suy

- Nội suy Lagrange
- Nội suy spline

3 Hồi qui

- Hồi qui tuyến tính
- Đường cong khớp bậc cao
- Đường cong khớp tổng quát

Vấn đề đặt ra là :

Giả sử cần tính xấp xỉ hàm $f(x)$ không biết rõ công thức, nhưng bằng thực nghiệm ta xác định được giá trị $f(x)$ tại $n + 1$ điểm rời rạc

$$f_i = f(x_i) \quad i = 0, 1, \dots, N$$

Cho bộ dữ liệu $\{(x_i, f_i), i = 0, 1, \dots, N\}$, cần tìm cách xấp xỉ hàm $f(x)$ bởi một hàm (có thể được tính theo công thức giải tích) $p(x)$.

Đặt vấn đề (tiếp)



tiếp tục...

Để giải quyết vấn đề

Ta sẽ tìm hiểu hai phương pháp dùng để xây dựng đường cong khớp (curve fitting)

- Nội suy (Interpolation) : hàm $p(x)$ phải đi qua tất cả các điểm thuộc bộ dữ liệu.
- Hồi quy (Regression): cho trước dạng $p(x)$ có tham số, ta phải xác định các tham số để cực tiểu hóa một tiêu chí sai số nào đó (thường dùng tiêu chí bình phương bé nhất (least squares)).

Mở đầu

Phép nội suy có thể thực hiện với các hàm $p(x)$ là

- Hàm đa thức
- Hàm hữu tỷ
- Hàm chuỗi Fourier

Nội suy

Nội suy

Định nghĩa : Ta nói hàm $p(x)$ là hàm nội suy bộ dữ liệu $\{(x_i, f_i), i = 0, 1, \dots, N\}$ nếu thỏa mãn các điều kiện sau

$$p_i = f_i, i = 0, 1, \dots, N$$

Hệ $N + 1$ phương trình này gọi là **điều kiện nội suy**. Sở dĩ hàm đa thức được chọn vì việc tính giá trị, đạo hàm, vi phân dễ dàng (tính chất giải tích của hàm rõ ràng).

Đa thức nội suy bộ dữ liệu được gọi là đa thức nội suy (interpolating polynominal)

Định nghĩa : Ta gọi đa thức bậc không quá K là hàm

$$p_K(x) = a_0 + a_1x + \dots + a_Kx^K$$

trong đó a_0, a_1, \dots, a_K là các hằng số.

Nội suy tuyến tính

Ta có hai điểm dữ liệu (x_0, f_0) và (x_1, f_1) cần nội suy bởi đa thức $p_1(x) \equiv a_0 + a_1x$ thỏa mãn hệ phương trình

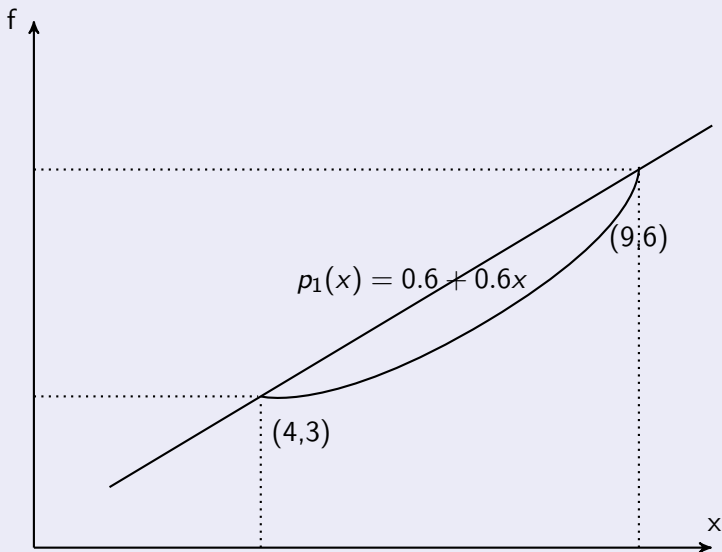
$$p_1(x_0) \equiv a_0 + a_1x_0 = f_0$$

$$p_1(x_1) \equiv a_0 + a_1x_1 = f_1$$

Giải ra, ta có

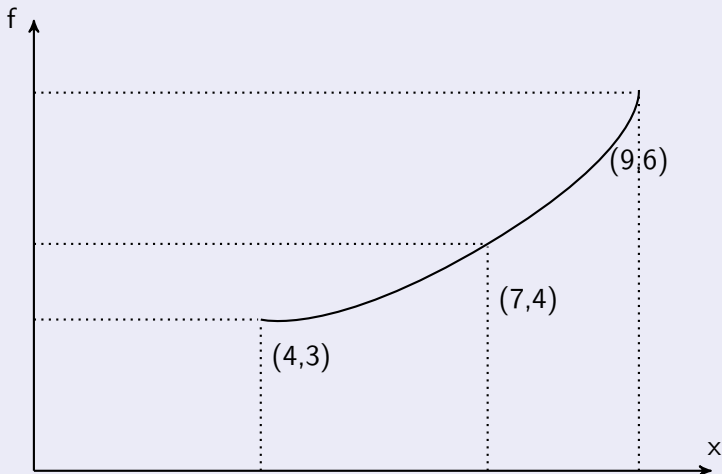
- Khi $x_0 \neq x_1$ xảy ra hai trường hợp
 - ▶ nếu $f_0 \neq f_1$ thì $p_1(x) = \frac{x_1 f_0 - x_0 f_1}{x_1 - x_0} + \frac{f_1 - f_0}{x_1 - x_0} x$
 - ▶ nếu $f_0 = f_1$ thì lời giải là hằng số \Rightarrow đa thức bậc không.

Nội suy tuyến tính (tiếp)



Công thức nội suy Lagrange

Cần xây dựng đường cong đi qua 3,4 hay nhiều điểm (xem hình).



Công thức nội suy Lagrange (tiếp)

Xét đa thức

$$p_M(x) = a_0 + a_1x + a_2x^2 + \cdots + a_Mx^M$$

Vậy đa thức $p_M(x)$ nội suy bộ dữ liệu $\{(x_i, f_i), i = 0, 1, \dots, N\}$ nếu các điều kiện nội suy :

$$p_M(x_0) \equiv a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_Mx_0^M = f_0$$

$$p_M(x_1) \equiv a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_Mx_1^M = f_1$$

...

$$p_M(x_N) \equiv a_0 + a_1x_N + a_2x_N^2 + \cdots + a_Mx_N^M = f_N$$

Đây là một hệ phương trình tuyến tính nên ta có thể biểu diễn bằng công thức dạng ma trận.

Nội suy

Công thức nội suy Lagrange (tiếp)

Như vậy dạng ma trận của điều kiện nội suy

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^M \\ 1 & x_1 & \cdots & x_1^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^M \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_N \end{pmatrix}$$

Giả thiết là x_0, x_1, \dots, x_N là phân biệt, vậy theo lý thuyết giải hệ phương trình tuyến tính trong chương trước ta có

- $N=M$: hệ phương trình có nghiệm duy nhất
- $M < N$: có thể chọn bộ dữ liệu để nó vô nghiệm
- $M > N$: nếu hệ có nghiệm thì nó có vô số nghiệm

Nội suy

Công thức nội suy Lagrange (tiếp)

Khi $N = M$ ma trận hệ số của hệ phương trình trên là ma trận Vandermonde (Không phải Voldemort)

$$V_N = \begin{pmatrix} 1 & x_0 & \cdots & x_0^M \\ 1 & x_1 & \cdots & x_1^M \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^M \end{pmatrix}$$

Có định thức của V_n là :

$$V_N = \prod_{i>j} (x_i - x_j)$$

suy ra hệ có nghiệm khi x_0, x_1, \dots, x_N là phân biệt từng đôi.

Công thức nội suy Lagrange (tiếp)

Định lý (Tính duy nhất của đa thức nội suy) : Nếu các nút dữ liệu x_0, x_1, \dots, x_N là khác nhau từng đôi thì tồn tại duy nhất đa thức nội suy $p_N(x)$ bậc không quá N nội suy bộ dữ liệu $\{(x_i, f_i), i = 0, 1, \dots, N\}$.

Bậc của đa thức nội suy có thể nhỏ hơn N

Ví dụ, khi 3 điểm dữ liệu $(x_0, f_0), (x_1, f_1), (x_3, f_3)$ thẳng hàng thì phương trình nội suy trở thành bậc không quá hai do nó là đường thẳng.

Ví dụ 1 :

Xác định các hệ số của đa thức nội suy $p_2(x) = a_0 + a_1x + a_2x^2$ nội suy bộ dữ liệu $\{(-1,0), (0,1), (1,3)\}$, ta có điều kiện nội suy

$$a_0 + (-1)a_1 + 1a_2 = 0$$

$$a_0 + 0a_1 + 0a_2 = 1$$

$$a_0 + 1a_1 + 1a_2 = 3$$

Giải hệ phương trình ta có $a_0 = 1$, $a_1 = 1.5$ và $a_2 = 0.5$ vậy đa thức nội suy là

$$p_2(x) = 1 + 1.5x + 0.5x^2$$

Công thức nội suy Lagrange (tiếp)

Việc dùng giải trực tiếp hệ phương trình điều kiện nội suy dùng phép khử Gauss chẳng hạn cần đòi hỏi nhiều tính toán $\mathcal{O}(N^3)$. Thêm nữa, vì lũy thừa của hệ số x có thể rất cao nên độ sai số lại càng lớn do hiệu ứng việc làm tròn số, thậm chí còn tràn số (nhớ lại là số dấu phẩy động IEEE 754/85 độ chính xác kép chỉ là $\approx \pm 10^{38}$).

\Rightarrow ta có thể dùng **công thức nội suy Lagrange** để tính hằng số của đa thức nội suy mà không cần giải hệ phương trình tuyến tính trên.

Nội suy

Công thức nội suy Lagrange (tiếp)

Xét bộ dữ liệu $\{(x_i, f_i), i = 0, 1, \dots, N\}$. Dạng Lagrange của đa thức nội suy đối với bộ dữ liệu là như sau :

$$p_N(x) = f_0 V_0(x) + f_1 V_1(x) + \dots + f_N V_N(x)$$

trong đó $V_i(x) : i = 0, 1, \dots, N$ là các đa thức bậc N thỏa mãn điều kiện :

$$V_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Họ các đa thức $V_i(x) : i = 0, 1, \dots, N$ được gọi là họ các đa thức cơ sở. Dễ dàng kiểm tra được rằng :

$$p_N(x_i) = f_i \quad i = 0, 1, \dots, N$$

Công thức nội suy Lagrange (tiếp)

Một trong lớp đa thức cơ sở có thể xây dựng theo công thức như sau

$$V_i(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

rõ ràng thỏa mãn

$$V_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Vậy ta có thể tóm tắt lại công thức trên như sau

$$V_i(x) = \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

Công thức nội suy Lagrange (tiếp)

Vậy công thức nội suy Lagrange

$$\begin{aligned} p_N(x) &= \frac{(x - x_1)(x - x_2) \cdots (x - x_N)}{(x_0 - x_1) \cdots (x_0 - x_N)} f_0 \\ &\dots \\ &+ \frac{(x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)} f_i \quad (1) \\ &\dots \\ &+ \frac{(x - x_0)(x - x_1) \cdots (x - x_{N-1})}{(x_N - x_0) \cdots (x_N - x_{N-1})} f_N \end{aligned}$$

Công thức (1) tuy dài nhưng dễ nhớ và tính toán hơn.

Cài đặt $V_i(x)$ bằng Matlab

$v = \text{polyinterp}(x,y,u)$ tính $v(j) = P(u(j))$ trong đó P là đa thức bậc $d = \text{length}(x) - 1$ thỏa mãn $P(x(i)) = y(i)$. Sử dụng công thức nội suy Lagrange. Hai vectơ x,y chứa hoành độ và tung độ của bộ dữ liệu.

```
function v = polyinterp(x,y,u)
n = length(x);
v = zeros(size(u));
for k = 1:n
    w = ones(size(u));
    for j = [1:k-1 k+1:n]
        w = (u-x(j))./(x(k)-x(j)).*w;
    end
    v = v + w*y(k);
end
```

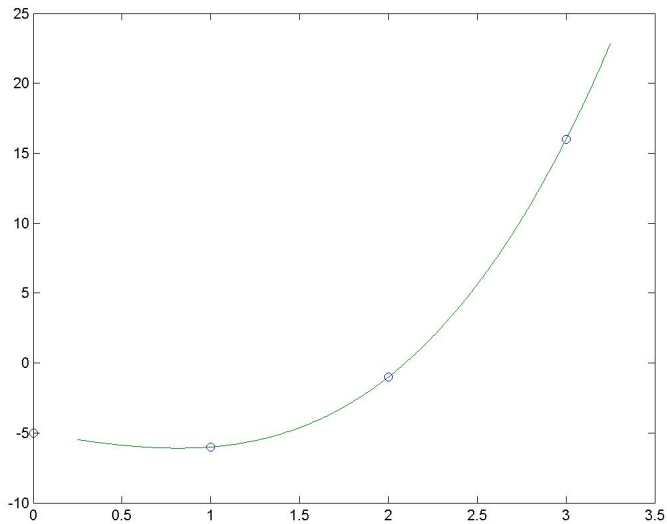
Cài đặt $V_i(x)$ bằng Matlab (tiếp)

```
function v = polyinterp(x,y,u)
n = length(x);
v = zeros(size(u));
for k = 1:n
    w = ones(size(u));
    for j = [1:k-1 k+1:n]
        w = (u-x(j))./(x(k)-x(j)).*w;
    end
    v = v + w*y(k);
end
```

- Hai vectơ **x,y** chứa hoành độ và tung độ của bộ dữ liệu.
- **u** là vec tơ chứa các hoành độ của các điểm cần tính theo công thức nội suy Lagrange.
- **v** là vectơ chứa giá trị hàm nội suy tại các điểm cho trong **u**

Ví dụ 2 :

```
x = [0 1 2 3];  
y = [-5 -6 -1 16];  
u = [0.25:0.01:3.25];  
v = polyinterp(x,y,u);  
plot(x,y,'o',u,v,'-')
```



Nội suy

Sai số nội suy

Định nghĩa : Sai số của đa thức nội suy $g(x)$ được xác định bởi công thức :

$$e(x) = f(x) - g(x)$$

Độ lớn của $e(x)$ phụ thuộc vào

- Hoàng độ của các điểm dữ liệu.
- Kích thước của miền nội suy $D = x_n - x_0$.
- Bậc của đa thức nội suy

Đối với đa thức nội suy Lagrange, sai số được xác định bởi công thức :

$$e(x) = f(x) - g(x) = L(x)f^{(N+1)}(\xi)$$

trong đó $f^{(N+1)}$ là đạo hàm bậc $N + 1$ của hàm $f(x)$ còn

$$L(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(N+1)!}$$

Sai số nội suy (tiếp)

tiếp tục ... công thức :

$$e(x) = f(x) - g(x) = L(x)f^{(N+1)}(\xi)$$

ξ phụ thuộc x nhưng thỏa mãn điều kiện $x_0 < \xi < x_N$. Cũng theo công thức trên nếu $f(x)$ là đa thức bậc không quá N thì đạo hàm $f^{(N+1)}$ sẽ triệt tiêu, nghĩa là sai số bằng không. Do $L(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_n)}{(N+1)!}$ nên ta có các kết luận sau

- Với một lưới cách đều, biên độ dao động của $L(x)$ nhỏ nhất tại tâm và tăng dần về biên của miền nội suy.
- Kích thước nội suy tăng lên, độ lớn của dao động tăng nhanh chóng.

Nội suy Lagrange



Nội suy spline

Đặt vấn đề

Trong trường hợp này ta xét phép nội suy bởi một tập các các đa thức bậc thấp thay vì một đa thức bậc cao duy nhất

- Nội suy bởi spline tuyến tính
- Nội suy bởi spline bậc ba

Nhớ lại công thức đánh giá sai số nội suy Lagrange

$$f(x) - p_N(x) = [w_{N+1}(x)/(N+1)!]f^{(N+1)}(\xi_x)$$

trong đó $w_{N+1}(x) = (x - x_0)(x - x_1) \cdots (x - x_N)$ và ξ_x là một giá trị trong khoảng (x_0, x_N) .

Nội suy spline

Đặt vấn đề (tiếp)

tiếp tục ... Giả sử mọi điểm dữ liệu $x_i \in [a, b]$ thì ta có

$$\max_{x \in [a, b]} |f(x) - p_N(x)| \leq |b - a|^{N+1} \times \frac{\max_{x \in [a, b]} |f^{(N+1)}(x)|}{N!}$$

Đánh giá này cho biết nếu muốn giảm sai số mà vẫn giữ nguyên số lượng điểm dữ liệu N thì cần tìm cách giảm kích thước của $|b - a|$.

Hướng tiếp cận của ta là xấp xỉ đa thức từng khúc (piecewise polynomial approximation): Như vậy đoạn $[a, b]$ sẽ được chia thành nhiều đoạn nhỏ không giao nhau và **các đa thức khác nhau sẽ được xấp xỉ trên từng đoạn con.**

Spline tuyến tính từng khúc

Đối với bộ dữ liệu

$$\{(x_i, f_i) : i = 0, 1, \dots, N\}$$

trong đó

$$a = x_0 < x_1 < \dots < x_N = b, h \equiv \max_i |x_i - x_{i-1}|,$$

spline tuyến tính $S_{1,N}(x)$ là hàm liên tục nội suy dữ liệu đã cho và được xây dựng từ các hàm tuyến tính được xác định bởi các đa thức nội suy hai điểm dữ liệu :

Nội suy spline

Spline tuyến tính từng khúc (tiếp)

tiếp tục...

$$S_{1,N}(x) = \begin{cases} \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) + f_0, & \text{nếu } x \in [x_0, x_1] \\ \vdots \\ \frac{f_i - f_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}) + f_{i-1}, & \text{nếu } x \in [x_{i-1}, x_i] \\ \vdots \\ \frac{f_N - f_{N-1}}{x_N - x_{N-1}}(x - x_{N-1}) + f_{N-1}, & \text{nếu } x \in [x_{N-1}, x_N] \end{cases}$$

Dễ thấy

$$L_i(x) = \frac{f_i - f_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}) + f_{i-1}$$

là phương trình đường thẳng qua hai điểm (x_i, f_i) và (x_{i-1}, f_{i-1})

Nội suy spline

Spline tuyến tính từng khúc (tiếp)

Sử dụng công thức sai số cho đa thức nội suy với $x \in [a, b]$ cho từng đoạn $[x_{i-1}, x_i]$ ta được

$$\begin{aligned} \max_{z \in [x_{i-1}, x_i]} |f(z) - S_{1,N}(z)| &\leq \frac{|x_i - x_{i-1}|^2}{8} \times \max_{x \in [x_{i-1}, x_i]} |f^{(2)}(x)| \\ &\leq \frac{h^2}{8} \times \max_{x \in [x_{i-1}, x_i]} |f^{(2)}(x)| \end{aligned}$$

trong đó $h \equiv \max_i |x_i - x_{i-1}|$.

Nội suy spline

Ví dụ 3 :

Cho bộ dữ liệu $\{(-1,0),(0,1),(1,3)\}$, ta xây dựng spline tuyến tính như sau

$$S_{1,2}(x) = \begin{cases} \frac{f_1-f_0}{x_1-x_0}(x-x_1) + f_1 = \frac{f_1-f_0}{x_1-x_0}(x-x_0) + f_0, & \text{nếu } x \in [x_0, x_1] \\ \frac{f_2-f_1}{x_2-x_1}(x-x_2) + f_2 = \frac{f_2-f_1}{x_2-x_1}(x-x_1) + f_1, & \text{nếu } x \in [x_1, x_2] \end{cases}$$

suy ra với bộ dữ liệu tương ứng

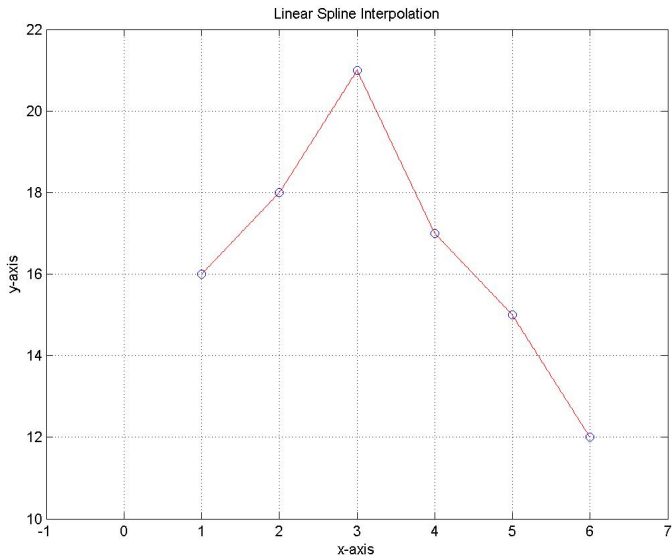
$$S_{1,2}(x) = \begin{cases} \frac{1-0}{0-(-1)}x + 1 = x + 1, & \text{nếu } x \in [-1, 0] \\ \frac{3-1}{1-0}(x-1) + 3 = 2x + 1, & \text{nếu } x \in [0, 1] \end{cases}$$

Nội suy spline

Ví dụ 4 :

```
» clear;  
» x=1:6;  
» y=[16 18 21 17 15 12];  
» plot(x,y,'o')  
» hold on; grid on;  
» axis([-1 7 10 22])  
» xx = [1:0.01:6];  
» yy = piecelin(x,y,xx);  
» plot(xx,yy,'r')  
» hold off  
» xlabel('x-axis');ylabel('y-axis');  
» title('Linear Spline Interpolation');
```


Nội suy spline



Spline bậc ba

Thay vì các đoạn thẳng, Spline bậc 3 sử dụng các đa thức bậc 3 để xấp xỉ đa thức từng khúc. Ta cũng có bộ dữ liệu

$$\{(x_i, f_i) : i = 0, 1, \dots, N\}$$

trong đó

$$a = x_0 < x_1 < \dots < x_N = b, h \equiv \max_i |x_i - x_{i-1}|,$$

Nội suy Lagrange



Nội suy spline

Spline bậc ba (tiếp)

Định nghĩa : Spline bậc ba $S_{3,N}(x)$ nội suy bộ dữ liệu đã cho

- $S_{3,N}(x)$ là hàm bậc ba giữa hai nút $[x_{i-1}, x_i]$

$$S_{3,N}(x) = \begin{cases} p_1(x) = a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3, & \text{nếu } x \in [x_0, x_1] \\ \vdots \\ p_i(x) = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + a_{i,3}x^3, & \text{nếu } x \in [x_{i-1}, x_i] \\ \vdots \\ p_N(x) = a_{N,0} + a_{N,1}x + a_{N,2}x^2 + a_{N,3}x^3, & \text{nếu } x \in [x_{N-1}, x_N] \end{cases}$$

- $S_{3,N}(x)$ là hàm liên tục có đạo hàm cấp một và hai liên tục trên đoạn $[x_0, x_N]$ (kể cả tại các nút)

Nội suy spline

Spline bậc ba (tiếp)

Với $x_{i-1} < x < x_i$, $S_{3,N}(x)$ có giá trị $p_i(x) = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + a_{i,3}x^3$	Với $x_i < x < x_{i+1}$, $S_{3,N}(x)$ có giá trị $p_{i+1}(x) = a_{i+1,0} + \dots + a_{i+1,3}x^3$
Giá trị của $S_{3,N}(x)$ khi $x \rightarrow x_i^-$ $p_i(x_i)$ $p'_i(x_i)$ $p''_i(x_i)$	Giá trị của $S_{3,N}(x)$ khi $x \rightarrow x_i^+$ $p_{i+1}(x_i)$ $p'_{i+1}(x_i)$ $p''_{i+1}(x_i)$

Nhận xét :

- Điều kiện về tính trơn tại điểm x_i thì

$$p'_i(x_i) = p'_{i+1}(x_i); \quad p''_i(x_i) = p''_{i+1}(x_i); \quad i = 1, 2, \dots, N-1$$

- Điều kiện nội suy dữ liệu đảm bảo

$$p_i(x_i) = p_{i+1}(x_i) = f_i; \quad i = 0, 1, 2, \dots, N-1$$

Spline bậc ba (tiếp)

- Điều kiện biên tự nhiên (Natural Boundary Condition)

$$p_1''(x_0) = 0; p_N''(x_N) = 0;$$

- Điều kiện đạo hàm cấp hai (Second Derivative Condition)

$$p_1''(x_0) = f''(x_0); p_N''(x_N) = f''(x_N);$$

- Điều kiện sát biên (Not-a-knot Condition)

$$p_1'''(x_1) = p_2'''(x_1); p_{N-1}'''(x_{N-1}) = p_N'''(x_{N-1});$$

Hàm tính Spline nội suy bậc 3 trong MatLab:

$p = \text{spline}(x, y)$: trả lại dạng đa thức từng khúc của spline bậc 3 đối với bộ dữ liệu (x, y) .

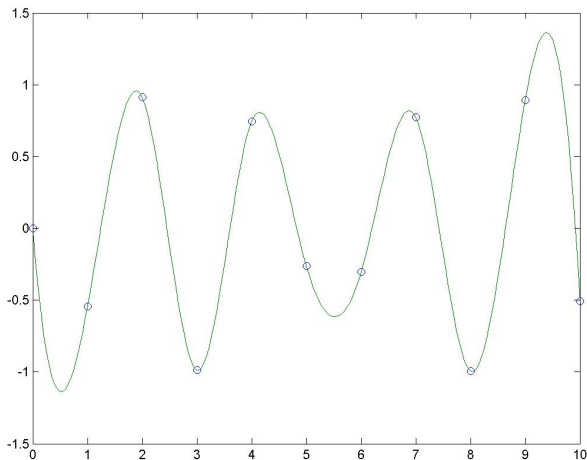
$v = \text{ppval}(p, x)$: Tính giá trị của đa thức nội suy tại đầu vào x (p xác định nhờ hàm spline).

$yy = \text{spline}(x, y, xx)$: Xác định spline nội suy bậc 3 đối với bộ dữ liệu (x, y) và trả lại yy là vectơ giá trị của spline nội suy tại các điểm trong vectơ xx .

Ví dụ 4 :

```
» x = 0:10; y = sin(10*x);  
» p = spline(x,y);  
» xx = linspace(0,10,200);  
» yy = ppval(p,xx);  
» plot(x,y,'o',xx,yy)
```


Nội suy spline cấp 3



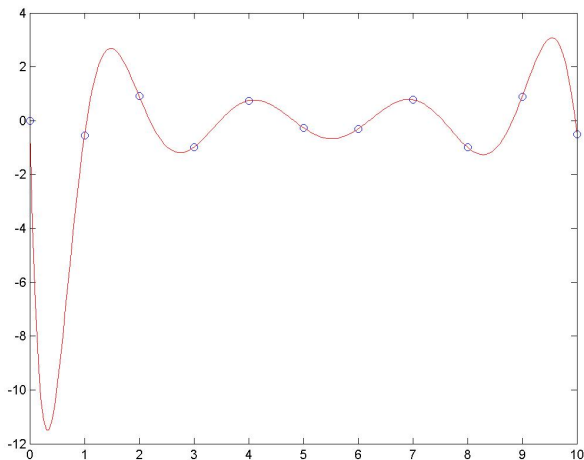
Ví dụ 5 :

```
» x = 0:10; y = sin(10*x);  
» xx = linspace(0,10,200);  
» coef = polyfit(x,y,10);
```

Warning: Polynomial is badly conditioned. Remove repeated data points or try centering and scaling as described in HELP POLYFIT. (Type "warning off MATLAB:polyfit:RepeatedPointsOrRescale" to suppress this warning.)

```
» yy = polyval(coef,xx);  
» plot(x,y,'o',xx,yy,'r')
```

Nội suy spline cấp 3



Như vậy khi ta nội suy bằng spline bậc 3 thì tốt hơn nội suy đa thức với hàm $\sin(10x)$

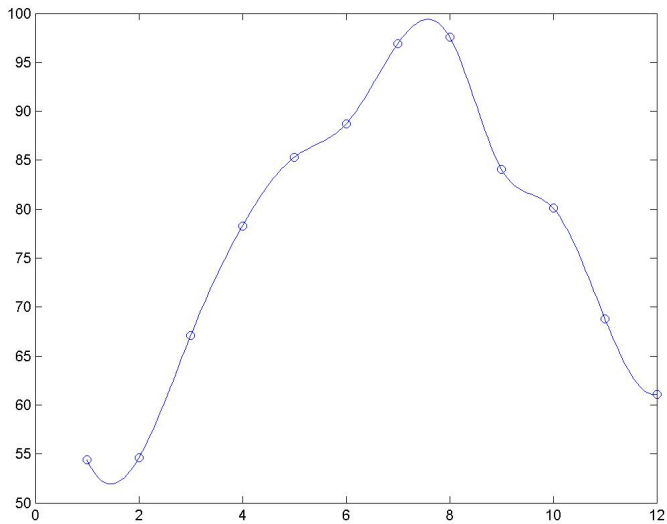
Ví dụ 6 :

Công suất tiêu thụ điện của một thiết bị trong 12 giờ như sau

1	2	3	4	5	6
54.4	54.6	67.1	78.3	85.3	88.7
7	8	9	10	11	12
96.9	97.6	84.1	80.1	68.8	61.1

Dùng spline bậc ba để nội suy tập điểm dữ liệu trên, ta có được

Nội suy spline cấp 3



Nội suy spline cấp 3



Đặt vấn đề

Cho bộ dữ liệu $\{(x_i, f_i), i = 1, \dots, N\}$, cần tìm cách xấp xỉ hàm $f(x)$ bởi một hàm $p(x)$. Ta cần quan tâm đo sai số tổng cộng trên toàn đoạn.

- Sai số cực đại $E_\infty(f) = \max_{1 \leq k \leq n} |f(x_k) - y_k|$
- Sai số trung bình $E_1(f) = \frac{1}{n} \sum_{k=1}^n |f(x_k) - y_k|$
- Căn bậc hai của tổng các bình phương sai số

$$E_2(f) = \sqrt{\frac{1}{n} \sum_{k=1}^n (f(x_k) - y_k)^2}$$

Hồi qui tuyến tính

Hồi qui tuyến tính là việc tìm một đường thẳng khớp với các điểm dữ liệu theo nghĩa bình phương tối thiểu.

- Cho tập N cặp điểm dữ liệu $\{(x_i, f_i) : i = 1, \dots, N\}$
- Tìm hệ số m và hằng số tự do b của đường thẳng

$$y(x) = mx + b$$

sao cho đường thẳng này khớp với dữ liệu theo tiêu chí bình phương tối thiểu.

$$L(m, b) = \sum_{i=1}^N (f_i - (mx_i + b))^2$$

Hồi qui tuyến tính (tiếp)

Nghĩa là ta cần tìm m và b để đạt cực tiểu hàm

$$L(m, b) = \sum_{i=1}^N (f_i - (mx_i + b))^2$$

để tìm cực tiểu này, ta cần giải hệ phương trình tìm điểm dừng

$$\frac{\partial L}{\partial m} = \sum_{i=1}^N 2(f_i - (mx_i + b))(-x_i) = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^N 2(f_i - (mx_i + b))(-1) = 0$$

Hồi qui tuyến tính (tiếp)

$$\begin{aligned}\left[\sum_{i=1}^N x_i^2 \right] m + \left[\sum_{i=1}^N x_i \right] b &= \sum_{i=1}^N x_i f_i \\ \left[\sum_{i=1}^N x_i \right] m + \left[\sum_{i=1}^N 1 \right] b &= \sum_{i=1}^N f_i\end{aligned}$$

hay dưới dạng ma trận,

$$\begin{pmatrix} \sum_{i=1}^N x_i^2 & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & N \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N x_i f_i \\ \sum_{i=1}^N f_i \end{pmatrix}$$

Giải hệ phương trình ta có được hệ số m và b .

Ví dụ 6 :

Xây dựng đường cong khớp cho bộ dữ liệu sau

i	1	2	3	4
x_i	1	3	4	5
f_i	2	4	3	1

Vậy hệ phương trình cần giải là

$$13m + 4b = 10$$

$$51m + 13b = 31$$

từ đó tìm được $b = 107/35$ và $m = -6/35$, đường thẳng cần tìm là
 $y = (-6/35)x + 107/35$

Đường cong khớp bậc cao

Xây dựng đường cong khớp

$$p_M(x) = a_0 + a_1x + a_2x^2 + \cdots + a_Mx^M$$

khớp với bộ dữ liệu $\{(x_i, f_i) | i = 1, \dots, N\}$ theo tiêu chí bình phương tối thiểu

$$\sigma_M \equiv (p_M(x_1) - f_1)^2 + (p_M(x_2) - f_2)^2 + \cdots + (p_M(x_N) - f_N)^2$$

Giải hệ phương trình tìm điểm dừng

$$\frac{\partial}{\partial a_0} \sigma_M = 0; \quad \frac{\partial}{\partial a_1} \sigma_M = 0; \quad \cdots \quad \frac{\partial}{\partial a_M} \sigma_M = 0;$$

Đường cong khớp bậc cao (tiếp)

Viết dưới dạng ma trận hệ phương trình chuẩn

$$\begin{pmatrix} \sum_{i=1}^N 1 & \sum_{i=1}^N x_i & \cdots & \sum_{i=1}^N x_i^M \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 & \cdots & \sum_{i=1}^N x_i^{M+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^N x_i^M & \sum_{i=1}^N x_i^{M+1} & \cdots & \sum_{i=1}^N x_i^{2M} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_M \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N f_i \\ \sum_{i=1}^N f_i x_i \\ \vdots \\ \sum_{i=1}^N f_i x_i^M \end{pmatrix}$$

Điểm đặc biệt của ma trận trên là đối xứng xác định dương nên ta có thể dùng phép khử Gauss mà không cần thay đổi dòng, để giải hệ phương trình chuẩn.

Ví dụ 7 :

Xác định các tham số a, b, c của đường cong

$$y = ax^2 + bx + c$$

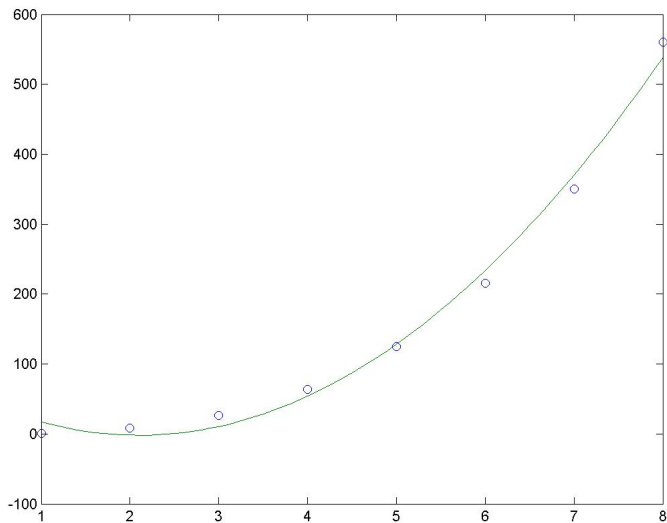
khớp với bộ dữ liệu sau

x	1	2	3	4	5	6	7	8
y	1	8	27	64	125	216	350	560

Ví dụ 7 : (tiếp)

```
x = [1 2 3 4 5 6 7 8];  
y = [1 8 27 64 125 216 350 560];  
s0 = length(x); s1 = sum(x); s2 = sum(x.^2);  
s3 = sum(x.^3); s4 = sum(x.^4);  
A = [s4 s3 s2; s3 s2 s1; s2 s1 s0];  
b = [sum(x.^2.*y); sum(x.*y);sum(y)];  
c0 = A\b;  
c = polyfit(x,y,2); % Tìm đa thức hồi qui khớp dữ liệu  
c0 % Đưa ra hệ số tính được  
c % Đưa ra hệ số tìm được bởi polyfit  
xx = 1:0.1:8;  
yy = polyval(c0,xx);  
plot(x,y,'o',xx,yy)
```

Hồi qui



Đường cong khớp tổng quát

Giả sử ta muốn khớp N điểm đa cho $\{(x_i, f_i) : i = 1, \dots, N\}$ bởi họ gồm m hàm độc lập tuyến tính $\varphi_j(x), j = 1, 2, \dots, m$ tức là hàm $f(x)$ cần tìm có dạng

$$f(x) = \sum_{j=1}^m c_j \varphi_j(x)$$

Chẳng hạn các họ hàm sau

- $\varphi_j(x) = x^j, j = 1, 2, \dots, m$
- $\varphi_j(x) = \sin(jx), j = 1, 2, \dots, m$
- $\varphi_j(x) = \cos(jx), j = 1, 2, \dots, m$

Đường cong khớp tổng quát (tiếp)

Ta cần xác định các số c_1, c_2, \dots, c_m để cực tiểu hóa sai số căn bậc hai của tổng các bình phương sai số :

$$E(c_1, c_2, \dots, c_m) = \sum_{k=1}^n (f(x_k) - f_k)^2 = \sum_{k=1}^n \left[\left(\sum_{j=1}^m c_j \varphi_j(x) \right) - f_k \right]^2$$

Để tìm các hệ số c_1, c_2, \dots, c_m , ta cần giải hệ phương trình

$$\frac{\partial E}{\partial c_j} = 0, j = 1, 2, \dots, m$$

Hồi qui



Tổng kết

- Đặt vấn đề đường cong khớp
- Định nghĩa nội suy - interpolation
 - ▶ Nội suy Lagrange
 - ▶ Nội suy Spline
- Định nghĩa hồi qui - regression
 - ▶ Hồi qui tuyến tính
 - ▶ Hồi qui đa thức bậc cao