CHƯƠNG 3: NỘI SUY VÀ XẤP XỈ HÀM

§1. NỘI SUY LAGRANGE

Trong thực tế nhiều khi ta cần tính giá trị của hàm y = f(x) tại một giá trị x trong một đoạn [a, b] nào đó mà chỉ biết một số nhất định các giá trị của hàm tại một số điểm cho trước. Các giá trị này được cung cấp qua thực nghiệm hay tính toán. Vì vậy nảy sinh vấn đề toán học là trên đoạn $a \le x \le b$ cho một loạt các điểm x_i (i = 0, 1, 2...) và tại các điểm x_i này giá trị của hàm là $y_i = f(x_i)$ đã biết và ta cần tìm y = f(x) dựa trên các giá trị đã biết đó. Lúc đó ta cần tìm đa thức :

$$P_n(x) = a_o x_n + a_1 x^{n-1} + \ldots + a_{n-1}x + a_n$$

sao cho $P_n(x_i) = f(x_i) = y_i$. Đa thức $P_n(x)$ được gọi là đa thức nội suy của hàm y=f(x). Ta chọn đa thức để nội suy hàm y = f(x) vì đa thức là loại hàm đơn giản, luôn có đạo hàm và nguyên hàm. Việc tính giá trị của nó theo thuật toán Horner cũng đơn giản.

Bây giờ ta xây dựng đa thức nội suy kiểu Lagrange. Gọi L_i là đa thức:

$$L_i = \frac{(x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_n)}$$

Rõ ràng là $L_i(x)$ là một đa thức bậc n và :

$$L_i(x_j) = \begin{cases} 1 & j = i \\ 0 & j \ne i \end{cases}$$

Ta gọi đa thức này là đa thức Lagrange cơ bản.

Bây giờ ta xét biểu thức :

$$P_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$$

Ta thấy $P_n(x)$ là một đa thức bậc n vì các $L_i(x)$ là các đa thức bậc n và thoả mãn điều kiện $P_n(x_i) = f(x_i) = y_i$. Ta gọi nó là đa thức nội suy Lagrange.

Với n = 1 ta có bảng

x	x_0	x_1
y	y_0	y_1

Đa thức nội suy sẽ là :

$$P_1(x) = y_o L_0(x) + y_1 L_1(x_1)$$

$$L_0 = \frac{x - x_1}{x_0 - x_1} \qquad L_1 = \frac{x - x_0}{x_1 - x_0}$$

nên $P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}$

Như vậy $P_1(x)$ là một đa thức bậc nhất đối với x

Với n = 2 ta có bảng

x	x_0	x_1	x_2
y	y_0	y_1	y_2

Đa thức nội suy sẽ là :

$P_2(x) = y_o L_0(x) + y_1 L_1(x_1) + y_2 L_2(x_2)$

$$L_0 = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}$$

$$L_1 = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$L_2 = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Như vậy $P_1(x)$ là một đa thức bậc hai đối với x.

Ta xây dựng hàm ***lagrange()*** để thực hiện việc nội suy hàm theo thuật toán Lagrange:

```
function [l, L] = lagrange(x, y)
%Dua vao : x = [x0 x1 ... xn], y = [y0 y1 ... yn]
%ket qua: l = He so cua da thuc Lagrange bac n
% L = Da thuc Lagrange
n = length(x) - 1; %bac cua da thucl
l = 0;
for m = 1:n + 1
   p = 1;
   for k = 1:n + 1
      if k ~= m
         p = conv(p, [1 -x(k)])/(x(m) - x(k));
      end
   end
   L(m, :) = p; %da thuc Lagrange
   l = l + y(m)*p;
end
```

Cho hàm dưới dạng bảng:

x	-2	-1	1	2
y	-6	0	0	6

và tìm y(2.5) ta dùng chương trình ***ctlagrange.m***:

```
clear all, clc
x = [-2 -1 1 2];
y = [-6 0 0 6];
l = lagrange(x, y);
yx = polyval(l, 2.5)
```

§2. NỘI SUY NEWTON

Bây giờ ta xét một cách khác để xây dựng đa thức nội suy gọi là phương pháp Newton. Trước hết ta đưa vào một khái niệm mới là tỉ hiệu

Giả sử hàm y = y(x) có giá trị cho trong bảng sau:

x	x_0	x_1	x_2	…	x_{n-1}	x_n
y	y_0	y_1	y_2	…	y_{n-1}	y_n

Tỉ hiệu cấp 1 của y tại x_i, x_j là :

$$y[x_i,x_j]=\frac{y_i-y_j}{x_i-x_j}$$

Tỉ hiệu cấp hai của y tại x_i, x_j, x_k là :

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

v.v.

Với y(x) = $P_n(x)$ là một đa thức bậc n thì tỉ hiệu cấp 1 tại x, x_0 :

$$P_n[x,x_0]=\frac{P_n(x)-P_n(x_0)}{x-x_0}$$

là một đa thức bậc (n - 1). Tỉ hiệu cấp 2 tại x, x_0, x_1 :

$$P_n[x,x_0,x_1]=\frac{P_n[x,x_0]-P_n[x_0,x_1]}{x-x_1}$$

là một đa thức bậc (n-2) v.v và tới tỉ hiệu cấp (n + 1) thì :

$P_n[x, x_o,.., x_n] = 0$

Từ các định nghĩa tỉ hiệu ta suy ra :

$P_n(x) = P_n(x_0) + (x - x_0)P_n[x, x_o]$

$P_n[x, x_0] = P_n[x_0, x_1] + (x - x_1)P_n[x, x_o, x_1]$

$P_n[x, x_o, x_1] = P_n[x_0, x_1, x_2] + (x - x_2)P_n[x, x_o, x_1, x_2]$

....

$P_n[x, x_o,.., x_{n-1}] = P_n[x_0, x_1,.., x_n] + (x - x_n)P_n[x, x_o,.., x_n]$

Do $P_n[x, x_o,.., x_n] = 0$ nên từ đó ta có :

$P_n(x) = P_n(x_0) + (x - x_0)P_n[x_o, x_1] + (x - x_0)(x - x_1)P_n[x_0, x_1, x_2] + \dots$
$+ (x - x_0)\dots(x - x_{n-1})P_n[x_0,\dots, x_n]$

Nếu $P_n(x)$ là đa thức nội suy của hàm $y = f(x)$ thì:

$P_n(x_i) = f(x_i) = y_i$ với $i = 0 \div n$

Do đó các tỉ hiệu từ cấp 1 đến cấp n của P_n và của y là trùng nhau và như vậy ta có :

$P_n(x) = y_0 + (x - x_0)y[x_0, x_1] + (x - x_0)(x - x_1)y[x_0, x_1, x_2] + .. +$
$(x - x_0)(x - x_1)...(x - x_{n-1})y[x_0,..,x_n]$

Đa thức này gọi là đa thức nội suy Newton tiến xuất phát từ nút x_0 của hàm $y = f(x)$. Ngoài đa thức tiến còn có đa thức nội suy Newton lùi xuất phát từ điểm x_n có dạng như sau :

$P_n(x) = y_n + (x - x_n)y[x_n, x_{n-1}] + (x - x_n)(x - x_{n-1})y[x_n, x_{n-1}, x_{n-2}] +..+$
$(x - x_n)(x - x_{n-1})...(x - x_1)y[x_n,.., x_0]$

Trường hợp các nút cách đều thì $x_i = x_0 + ih$ với i = 0, 1,.., n. Ta gọi sai phân tiến cấp 1 tại i là :

$\Delta y_i = y_{i+1} - y_i$

và sai phân tiến cấp hai tại i:

$\Delta^2 y_i = \Delta(\Delta y_i) = y_{i+2} - 2y_{i+1} + y_i$

....

và sai phân tiến cấp n là :

$\Delta^n y_i = \Delta(\Delta^{n-1} y_i)$

Khi đó ta có:

$$y[x_0, x_1] = \frac{\Delta y_0}{h}$$

$$y[x_0, x_1, x_2] = \frac{\Delta^2 y_0}{2h^2}$$

....

$$y[x_0, x_1, x_2, ..., x_n] = \frac{\Delta^n y_0}{n! h^n}$$

Bây giờ đặt $x = x_0 + ht$ trong đa thức Newton tiến ta được:

$$P_n(x_0 + ht) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \cdots + \frac{t(t-1)\cdots(t-n+1)}{n!}\Delta^n y_0$$

thì ta nhận được đa thức Newton tiến xuất phát từ x_0 trong trường hợp nút cách đều. Với n = 1 ta có :

$$P_1(x_0 + ht) = y_0 + \Delta y_0$$

Với n = 2 ta có:

$$P_n(x_0 + ht) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0$$

Một cách tương tự ta có khái niệm các sai phân lùi tại i:

$\nabla y_i = y_i - y_{i-1}$

$\nabla^2 y_i = \nabla(\nabla y_i) = y_i - 2y_{i-1} + y_{i-2}$

....

$\nabla^n y_i = \nabla(\nabla^{n-1} y_i)$

và đa thức nội suy Newton lùi khi các điểm nội suy cách đều:

$$P_n(x_0 + ht) = y_n + t\nabla y_n + \frac{t(t+1)}{2!}\nabla^2 y_n + \cdots + \frac{t(t+1)\cdots(t+n-1)}{n!}\nabla^n y_n$$

Ta xây dựng hàm ***newton()*** để nội suy:

```
function [n,DD] = newton(x,y)
%Dua vao : x = [x0 x1 ... xN]
% y = [y0 y1 ... yN]
%Lay ra: n = he so cua da thuc Newton bac N
N = length(x) - 1;
DD = zeros(N + 1, N + 1);
DD(1:N + 1, 1) = y';
for k = 2:N + 1
   for m = 1: N + 2 - k
      DD(m,k) = (DD(m + 1, k - 1) - DD(m, k - 1))/(x(m + k - 1) - x(m));
   end
end
a = DD(1, :);
n = a(N+1);
for k = N:-1:1
```

```
    n = [n a(k)] - [0 n*x(k)];
  end
```

Cho hàm dưới dạng bảng:

x	-2	-1	1	2	4
y	-6	0	0	6	60

Ta dùng chương trình ***ctnewton.m*** để nội suy:

```
clear all, clc
x = [-2 -1 1 2 4];
y = [-6 0 0 6 60];
a = newton(x, y)
yx = polyval(a, 2.5)
```

§3. NỘI SUY AITKEN - NEVILLE

Một dạng khác của đa thức nội suy được xác định bằng thuật toán Aitken - Neville. Giả sử ta có n điểm đã cho của hàm f(x). Như vậy qua hai điểm x_0 và x_1 ta có đa thức nội suy Lagrange của hàm f(x) được viết dưới dạng:

$$P_{01}(x) = \frac{\begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}}{x_1 - x_0}$$

Đây là một đa thức bậc 1:

$$P_{01}(x) = y_0\frac{x - x_1}{x_0 - x_1} + y_1\frac{x - x_0}{x_1 - x_0}$$

Khi $x = x_0$ thì:

$$P_{01}(x_0) = \frac{\begin{vmatrix} y_0 & x_0 - x_0 \\ y_1 & x_1 - x_0 \end{vmatrix}}{x_1 - x_0} = y_0$$

Khi $x = x_1$ thì:

$$P_{01}(x_1) = \frac{\begin{vmatrix} y_0 & x_0 - x_1 \\ y_1 & x_1 - x_1 \end{vmatrix}}{x_1 - x_0} = y_1$$

Đa thức nội suy Lagrange của f(x) qua 3 điểm x_0, x_1, x_2 có dạng:

$$P_{012}(x) = \frac{\begin{vmatrix} P_{01}(x) & x_0 - x \\ P_{12}(x) & x_2 - x \end{vmatrix}}{x_2 - x_0}$$

và là một đa thức bậc 2:

$$P_{012}(x) = y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

Khi $x = x_0$ thì:

$$P_{012}(x_0) = \frac{\begin{vmatrix} y_0 & x_0 - x_0 \\ P_{12}(x) & x_2 - x_0 \end{vmatrix}}{x_2 - x_0} = y_0$$

Khi $x = x_1$ thì:

$$P_{012}(x_1) = \frac{\begin{vmatrix} y_1 & x_0 - x_1 \\ y_1 & x_2 - x_1 \end{vmatrix}}{x_2 - x_0} = y_1$$

Khi $x = x_2$ thì:

$$P_{012}(x_2) = \frac{\begin{vmatrix} P_{01}(x_2) & x_0 - x_2 \\ y_2 & x_2 - x_2 \end{vmatrix}}{x_2 - x_0} = y_2$$

Tổng quát đa thức nội suy Lagrange qua n điểm là:

$$P_{012..n}(x) = \frac{\begin{vmatrix} P_{01..(n-1)}(x) & x_0 - x \\ P_{12..n}(x) & x_n - x \end{vmatrix}}{x_2 - x_0}$$

Như vậy ta có thể dùng phép lặp để xác định lần lượt các đa thức Lagrange. Sơ đồ tính toán như vậy gọi là sơ đồ Neville - Aitken.

Ta xây dựng hàm ***aitkenneville()*** để nội suy:

```
function a = aitkenneville(xData, yData, x)
% Tra ve gia tri noi suy tai x.
% Cu phap: y = aitkenneville(xData, yData, x)
n = length(xData);
y = yData;
for k = 1:n-1
    y(1:n-k) = ((x - xData(k+1:n)).*y(1:n-k)...
    + (xData(1:n-k) - x).*y(2:n-k+1))...
    ./(xData(1:n-k) - xData(k+1:n));
```

```
end
a = y(1);
```

Cho các cặp số (1, 3), (2, 5), (3, 7), (4, 9) và (5, 11), để tìm y tại x = 2.5 ta dùng chương trình ***ctaitkennevile.m***:

```
clear all, clc
x = [1  2  3  4];
y = [3  5  7  9];
yx = aitkenneville(x, y, 2.5)
```

§4. NỘI SUY BẰNG ĐƯỜNG CONG SPLINE BẬC BA

Khi số điểm cho trước dùng khi nội suy tăng, đa thức nội suy có dạng sóng và sai số tăng. Ta xét hàm thực:

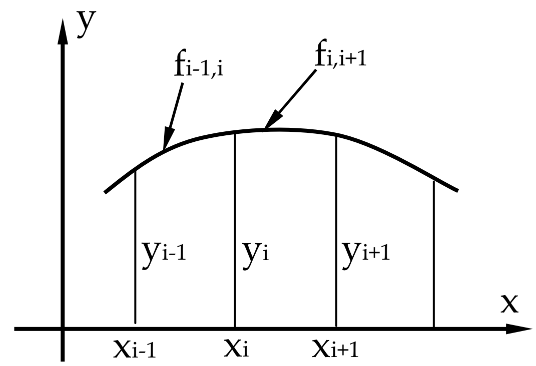
$$f31(x) = \frac{1}{1+8x^2}$$

và nội suy nó bằng thuật toán Newton nhờ chương trình ***cttestintp.m***


```
%Noi suy Newton
x1 = [-1 -0.5 0 0.5 1.0];
y1 = f31(x1);
n1 = newton(x1,y1)
x2 = [-1 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1.0];
y2 = f31(x2);
n2 = newton(x2,y2)
x3 = [-1 -0.8 -0.6 -0.4 -0.2  0  0.2  0.4  0.6  0.8  1.0];
y3 = f31(x3);
n3 = newton(x3,y3)
xx = [-1:0.02: 1]; %pham vi noi suy
yy = f31(xx); %ham thuc
yy1 = polyval(n1, xx); %ham xap xi qua 5 diem
yy2 = polyval(n2, xx); %ham xap xi qua 9 diem
yy3 = polyval(n3, xx); %ham xap xi qua 11 diem
subplot(221)
plot(xx, yy, 'k-',  xx, yy1, 'b')
subplot(224)
```

```
plot(xx, yy1-yy, 'r', xx, yy2-yy, 'g', xx, yy3-yy,'b') %do thi sai so
subplot(222)
plot(xx,yy,'k-',  xx, yy2, 'b')
subplot(223)
plot(xx, yy, 'k-',  xx, yy3, 'b')
```

và nhận được kết quả.



Để tránh hiện tượng sai số lớn khi số điểm mốc tăng ta dùng nội suy nối trơn(spline). Trên các đoạn nội suy ta thay hàm bằng một đường cong. Các đường cong này được ghép trơn tại các điểm nối. Ta chọn các đường cong này là hàm bậc 3 vì hàm bậc 1 và bậc hai khó bảo đảm điều kiện nối trơn.

Cho một loạt giá trị nội suy $(x_1, y_1),\ldots,(x_i, y_i),\ldots,(x_n, y_n)$. Trên mỗi đoạn ta có một hàm bậc 3. Như vậy giữa nút i và (i +1) ta có hàm $f_{i,i+1}(x)$, nghĩa là ta dùng (n - 1) hàm bậc 3 $f_{1,2}(x)$, $f_{2,3}(x),\ldots, f_{n-1,n}(x)$ để thay thế cho hàm thực. Hàm $f_{i,i+1}(x)$ có dạng:

$$f_{i,i+1}(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \qquad (1)$$

Hàm này thoả mãn:

$$f_{i,i+1}(x_i) = a_i = y_i \qquad (3)$$

$$f_{i,i+1}(x_{i+1}) = d_ih_i^3 + c_ih_i^2 + b_ih_i + a_i = y_{i+1} \qquad (4)$$

$$f'_{i,i+1}(x_i) = b_i \qquad (5)$$

$$f'_{i,i+1}(x_{i+1}) = 3d_ih_i^2 + 2c_ih_i + b_i \qquad (6)$$

$$f''_{i,i+1}(x_i) = 2c_i = y''_i \qquad (7)$$

$$f''_{i,i+1}(x_{i+1}) = 6d_ih_i + 2c_i = y''_{i+1} \qquad (8)$$

Muốn nối trơn ta cần có đạo hàm bậc nhất liên tục và do đó:

$$f''_{i-1,i}(x_i) = f''_{i,i+1}(x_i) = k_i$$

Lúc này các giá trị k chưa biết, ngoại trừ $k_1 = k_n = 0$(ta các các mút là điểm uốn). Điểm xuất phát để tính các hệ số của $f_{i,i+1}(x)$ là biểu thức của $f''_{i,i+1}(x_i)$. Sử dụng nội suy Lagrange cho hai điểm ta có:

$$f''_{i,i+1}(x_i) = k_iL_i(x) + k_{i+1}L_{i+1}(x)$$

Trong đó:

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

Do vậy:

$$f''_{i,i+1}(x_i) = \frac{k_i(x - x_{i+1}) - k_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Tích phân biểu thức trên hai lần theo x ta có:

$$f_{i,i+1}(x_i) = \frac{k_i(x - x_{i+1})^3 - k_{i+1}(x - x_i)^3}{6(x_i - x_{i+1})} + A(x - x_{i+1}) - B(x - x_i)$$

Trong đó A và B là các hằng số tích phân

Số hạng cuối trong phương trình trên thường được viết là Cx + D.

Đặt C = A - B và D = $-Ax_{i+1} + Bx_i$ để dễ dàng tính toán. Từ điều kiện $f_{i,i+1}(x_i) = y_i$ ta có:

$$\frac{k_i(x_i - x_{i+1})^3}{6(x_i - x_{i+1})} + A(x_i - x_{i+1}) = y_i$$

nên:

$$A = \frac{y_i}{x_i - x_{i+1}} - \frac{k_i(x_i - x_{i+1})}{6}$$

Tương tự, điều kiện $f_{i,i+1}(x_{i+1}) = y_{i+1}$ cho ta:

$$B = \frac{y_{i+1}}{x_i - x_{i+1}} - \frac{k_{i+1}(x_i - x_{i+1})}{6}$$

Kết quả là:

$$f_{i,i+1}(x_i) = \frac{k_i}{6}\left[\frac{(x - x_{i+1})^3}{x_i - x_{i+1}} - (x - x_{i+1})(x_i - x_{i+1})\right]$$

$$- \frac{k_{i+1}}{6}\left[\frac{(x - x_i)^3}{x_i - x_{i+1}} - (x - x_i)(x_i - x_{i+1})\right]$$

$$+ \frac{y_i(x - x_{i+1}) - y_{i+1}(x - x_i)}{x_i - x_{i+1}}$$

Đạo hàm cấp 2 k_i tại các nút bên trong được tính từ điều kiện:

$$f'_{i-1,i}(x_i) = f'_{i,i+1}(x_i)$$

Sau khi biến đổi ta có phương trình:

$$k_{i-1}(x_{i-1} - x_i) + 2k_i(x_{i-1} - x_{i+1}) + k_{i+1}(x_i - x_{i+1})$$

$$= 6\left(\frac{y_{i-1} - y_i}{x_{i-1} - x_i} - \frac{y_i - y_{i+1}}{x_i - x_{i+1}}\right)$$

Khi các điểm chia cách đều $(x_{i+1} - x_i) = h$ ta có:

$$k_{i-1} + 4k_i + k_{i+1} = \frac{6}{h^2}\left(y_{i-1} - 2y_i + y_{i+1}\right) \qquad i = 2, 3,\dots, n - 1$$

Ta xây dựng hàm ***cubicspline()*** để nội suy:

```
function y = cubicspline(xData, yData, x)
%Ham nay xap xi bang da thuc bac 3 spline
%Cu phap: [yi,f] = cubicspline(xData, yData, x)
n = length(xData);
c = zeros(n-1, 1); d = ones(n, 1);
e = zeros(n-1, 1); k = zeros(n, 1);
c(1:n-2) = xData(1:n-2) - xData(2:n-1);
d(2:n-1) = 2*(xData(1:n-2) - xData(3:n));
e(2:n-1) = xData(2:n-1) - xData(3:n);
k(2:n-1) = 6*(yData(1:n-2) - yData(2:n-1))...
./(xData(1:n-2) - xData(2:n-1))...
- 6*(yData(2:n-1) - yData(3:n))...
./(xData(2:n-1) - xData(3:n));
[c, d, e] = band3(c, d e);
k = band3sol(c, d, e, k);
i = findseg(xData, x);
h = xData(i) - xData(i+1);
y = ((x - xData(i+1))^3/h - (x - xData(i+1))*h)*k(i)/6.0...
- ((x - xData(i))^3/h - (x - xData(i))*h)*k(i+1)/6.0...
+ yData(i)*(x - xData(i+1))/h...
- yData(i+1)*(x - xData(i))/h;
```

Ta có chương trình ***ctcubicspline.m*** dùng nội suy:

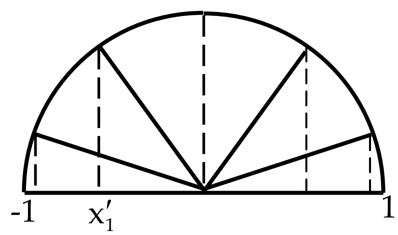
```
clear all, clc
x1 = 0:0.1:5;
y1 = (x1+1).^2;
while 1
  x = input('x = ');
   if isempty(x)
     fprintf('Ket thuc');
     break
```

```
    end
    y = cubicspline(xData, yData, x)
    fprintf('\n')
end
```

§5. NỘI SUY BẰNG ĐA THỨC CHEBYSHEV

Khi nội suy bằng đa thức Newton hay Lagrange, nghĩa là thay hàm thực bằng đa thức xấp xỉ, có khoảng cách cách đều thì sai số giữa đa thức nội suy và hàm thực có xu hướng tăng tại hai mút nội suy. Ta thấy rõ điều này khi chạy chương trình ***cttestintp.m***.

Do vậy ta nên chọn các điểm mốc nội suy ở hai mút dày hơn ở giữa. Một trong những cách chọn phân bố các điểm mốc là hình chiếu lên trục x của các điểm cách đều trên đường tròn tâm tại điểm giữa của đoạn nội suy. Như vậy với đoạn nội suy [-1, 1] ta có:



$$x'_k = \cos\frac{2n+1-2k}{2(n+1)}\pi \qquad k = 1, 2,\ldots,n \qquad (1)$$

Với đoạn nội suy [a, b] bất kì:

$$x_k = \frac{b-a}{2}x'_k + \frac{b+a}{2} = \frac{b-a}{2}\cos\frac{2n+1-2k}{2(n+1)}\pi + \frac{a+b}{2} \qquad k = 1, 2,\ldots,n \quad (2)$$

Các nút nội suy này được gọi là các nút Chebyshev. Đa thức nội suy dựa trên các nút Chebyschev gọi là đa thức nội suy Chebyshev.

Ta xét hàm thực:

$$f(x) = \frac{1}{1+8x^2}$$

Ta chọn số nút nội suy lần lượt là 5, 9, 11 và xây dựng các đa thức Newton (hay Lagrange) $c_4(x)$, $c_8(x)$ và $c_{10}(x)$ đi qua các nút này và vẽ đồ thị của hàm thực cũng như sai số khi nội suy bằng chương trình ***ctcomchebynew.m*** với các N khác nhau.

```
x1 = [-1 -0.5 0 0.5 1.0];
y1 = f31(x1);
n1 = newton(x1,y1);
xx = [-1:0.02: 1]; %pham vi noi suy
yy1 = polyval(n1,xx); %ham xap xi qua 5 diem
yy = f31(xx); %ham thuc
```

```
subplot(221)
plot(xx,yy,'k-', x, y, 'o', xx, yy1, 'b');
title('Newton')
subplot(223)
plot(xx, yy1-yy, 'r') %do thi sai so
N = 4;
k = [0:N];
x = cos((2*N + 1 - 2*k)*pi/2/(N + 1));
y = f31(x);
c = newton(x, y) %da thuc noi suy dua tren cac nut Chebyshev
xx = [-1:0.02: 1]; %doan noi suy
yy = f31(xx); %do thi ham thuc
yy1 = polyval(c, xx); %do thi ham xap xi
subplot(222)
plot(xx, yy, 'k-',  x, y, 'o',  xx, yy1, 'b')
title('Chebyshev')
subplot(224)
plot(xx, yy1-yy, 'r') %do thi sai so
```


Khi tăng số điểm mốc, nghĩa là tăng bậc của đa thức Chebyschev, sai số giảm. Đa thức Chebyshev bậc n được xác định bằng:

$$T_{n+1}(x') = \cos((n+1)\arccos(x')) \qquad (3)$$

và các nút Chebyshev cho bởi (1) là nghiệm của (3).

Ta có:

$$\begin{aligned} T_{n+1}(x') &= \cos(\arccos(x') + n\arccos(x')) \\ &= \cos(\arccos(x'))\cos(n\arccos(x') - \sin(\arccos(x'))\sin(n\arccos(x')) \\ &= x'T_n(x') + 0.5\left[\cos((n+1)\arccos(x') - \cos((n-1)\arccos(x')\right] \\ &= x'T_n(x') + 0.5T_{n+1}(x') - 0.5T_{n-1}(x') \end{aligned}$$

nên:

$$T_{n+1}(x') = 2xT_n(x') - T_{n-1}(x') \qquad n \geq 1 \qquad (4)$$

và $T_0(x') = 1$ $\qquad T_1(x') = \cos(\arccos(x') = x'$ $\qquad$ (5)

Các đa thức Chebyshev đến bậc 6 là:

$T_0(x) = 1$

$T_1(x') = x'$

$T_2(x') = 2x'^2 - 1$

$T_3(x') = 4x'^3 - 3x'$
$T_4(x') = 8x'^4 - 8'x^2 + 1$
$T_5(x') = 16x'^5 - 20'x^3 + 5x'$
$T_6(x') = 32x'^6 - 48x'^4 + 18x'^2 - 1$
$T_7(x') = 64x'^7 - 112x'^5 + 56x'^3 - 7x'$

Hàm f(x) được xấp xỉ bằng:

$$f(x) = \sum_{m=0}^{N} d_m T_m(x')\Big|_{x'=\frac{2}{b-a}\left(x-\frac{a+b}{2}\right)} \qquad (6)$$

Trong đó:

$$d_0 = \frac{1}{n+1}\sum_{k=0}^{n} f(x_k)T_0(x'_k) = \frac{1}{n+1}\sum_{k=0}^{n} f(x_k) \qquad (7)$$

$$\begin{aligned} d_m &= \frac{2}{n+1}\sum_{k=0}^{n} f(x_k)T_m(x'_k) \\ &= \frac{2}{n+1}\sum_{k=0}^{n} f(x_k)\cos\frac{m(2n+1-2k)}{2(n+1)}\pi \qquad m=1,2,\dots,n \end{aligned} \qquad (8)$$

Ta xây dựng hàm ***cheby()*** để tìm đa thức nội suy Chebyshev:


```
function [c, x, y] = cheby(f, N, a, b)
%vao : f = ten ham tren doan [a, b]
%Ra: c = Cac he so cua da thuc Newton bac N
% (x,y) = cac nut Chebyshev
if nargin == 2
   a = -1;
   b = 1;
end
k = [0: N];
theta = (2*N + 1 - 2*k)*pi/(2*N + 2);
xn = cos(theta); %pt.(1)
x = (b - a)/2*xn +(a + b)/2; %pt.(2)
y = feval(f,x);
d(1) = y*ones(N + 1,1)/(N+1);
for m = 2: N + 1
        cos_mth = cos((m-1)*theta);
   d(m) = y*cos_mth'*2/(N + 1); %pt.(7)
end
xn = [2 -(a + b)]/(b - a); %nghich dao cua t. (2)
```

```
T_0 = 1; T_1 = xn; %pt.(5)
c = d(1)*[0 T_0] +d(2)*T_1; %pt.(6)
for m = 3: N + 1
    tmp = T_1;
    T_1 = 2*conv(xn,T_1) -[0 0 T_0]; %pt.(4)
    T_0 = tmp;
    c = [0 c] + d(m)*T_1; %pt.(6)
end
```

Để tìm đa thức Chebyshev dùng xấp xỉ hàm $f(x)=\dfrac{1}{1+8x^2}$ ta dùng chương trình ***ctcheby.m***:

```
clear all, clc
N = 2;
a = -2;
b = 2;
[c, x1, y1] = cheby('f31', N, a, b) %da thuc Chebyshev
%so sanh voi da thuc Lagrange/Newton
k = [0:N];
xn = cos((2*N + 1 - 2*k)*pi/2/(N + 1));%pt.(1):nut Chebyshev
x = ((b-a)*xn +a + b)/2; %pt.(2)
y = f31(x);
n = newton(x, y)
l = lagrange(x, y)
```

§6. XẤP XỈ HÀM BẰNG PHÂN THỨC HỮU TỈ

Xấp xỉ Padé dùng để xấp xỉ hàm f(x) tại x_0 bằng hàm hữu tỉ:

$$P_{m,n}(x-x_0)=\frac{Q_m(x-x_0)}{D_n(x-x_0)}$$
$$=\frac{q_0+q_1(x-x_0)+q_2(x-x_0)^2+\cdots+q_m(x-x_0)^m}{1+d_1(x-x_0)+d_2(x-x_0)^2+\cdots+d_n(x-x_0)^n} \qquad (1)$$

với m = n hay m = n + 1

Trong đó $f(x_0)$, $f'(x_0)$,…, $f^{(m+n)}(x_0)$ đã cho

Trước hết ta khai triển Taylor hàm f(x) tại $x = x_0$ đến bậc (m + n).

$$
\begin{aligned}
f(x) \approx T_{m+n}(x) &= f(x_0) + f'(x_0)(x - x_0) \\
&+ \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(m+n)}(x_0)}{(m+n)!}(x - x_0)^{m+n} \qquad (2) \\
&= a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + \cdots + a_{m+n}(x - x_0)^{m+n}
\end{aligned}
$$

Để đơn giản ta coi $x_0 = 0$. Ta cần tính các hệ số của $D_n(x)$ và $Q_m(x)$ sao cho:

$$T_{m+n}(x) - \frac{Q_m(x)}{D_n(x)} = 0 \text{ hay } T_{m+n}(x)D_n(n) - Q_m(x) = 0$$

nghĩa là:

$$(a_0 + a_1x + \cdots + a_{m+n}x^{m+n})(1 + d_1x + \cdots + d_nx^n) = (q_0 + q_1x + \cdots + q_mx^m) \quad (3)$$

Cân bằng các số hạng cùng bậc ở hai vế ta có:

$$
\begin{cases}
a_0 = q_0 \\
a_1 + a_0d_1 = q_1 \\
a_2 + a_1d_1 + a_0d_2 = q_2 \\
\cdots \\
a_m + a_{m-1}d_1 + a_{m-2}d_2 + \cdots + a_{m-n}d_n = q_m
\end{cases} \qquad (4)
$$

$$
\begin{cases}
a_{m+1} + a_md_1 + a_{m-1}d_2 + \cdots + a_{m-n+1}d_n = 0 \\
a_{m+2} + a_{m+1}d_1 + a_md_2 + \cdots + a_{m-n+2}d_n = 0 \\
\cdots \\
a_{m+n} + a_{m+n-1}d_1 + a_{m+n-2}d_2 + \cdots + a_md_n = 0
\end{cases} \qquad (5)
$$

Trước hết ta giải (5) để tìm d_i và sau đó thay vào (4) để tìm q_i.

Ta xây dựng hàm ***padeapp()*** để tính xấp xỉ:

```
function [num, den] = padeapp(f, xo, M, N, x0, xf)
%Vao : f = Ham can xap xi trong doan [xo, xf]
%Ra: num = Cac he so cua tu so
% den = Cac he so cua mau so
a(1) = feval(f, xo);
h = .01;
tmp = 1;
for i = 1:M + N
    tmp = tmp*i*h; %i!h^i
    dix = difapx(i, [-i i])*feval(f, xo + [-i:i]*h)'; %dao ham
    a(i + 1) = dix/tmp; %he so chuoi Taylor
```

```
end
for m = 1:N
n = 1:N;
    A(m, n) = a(M + 1 + m - n);
    b(m) = -a(M + 1 + m);
end
d = A\b'; %pt.(5)
for m = 1: M + 1
    mm = min(m - 1,N);
    q(m) = a(m:-1:m - mm)*[1; d(1:mm)]; %pt.(4)
end
num = q(M + 1:-1:1)/d(N); den = [d(N:-1:1)' 1]/d(N); %giam dan
if nargout == 0 % ve ham thuc, khai trien taylor va ham Pade
    if nargin < 6
        x0 = xo - 1;
        xf = xo + 1;
    end
    x = x0 + [xf - x0]/100*[0:100];
    yt = feval(f, x);
    x1 = x - xo;
    yp = polyval(num,x1)./polyval(den,x1);
    yT = polyval(a(M + N + 1:-1:1),x1);
    clf,  plot(x, yt, 'k', x, yp, 'r',  x, yT, 'b')
end
```

Để xấp xỉ hàm e^x ta dùng chương trình ***ctpadeapp.m***:

```
f1 = inline('exp(x)', 'x');
M = 3;
N = 2; %bac cua Q(x) va D(x)
xo = 0; %tam cua chuoi Taylor
[n,d] = padeapp(f1, xo, M, N) %tinh cac he so cua Q(x)/P(x)
x0 = -3.5;
xf = 0.5; %bien trai va phai cua khoang xap xi
padeapp(f1, xo, M, N, x0, xf) %xem do thi
```

§7. NỘI SUY BẰNG ĐA THỨC HERMIT

Trong một số trường hợp, ta cần tìm hàm đa thức không những đi qua các điểm cho trước mà còn phải thoả mãn điều kiện về đạo hàm tại các điểm đó. Ta gọi đa thức như vậy là đa thức nội suy Hermit. Để đơn giản, ta khảo sát một đa thức bậc 3:

$$h(x) = H_3x^3 + H_2x^2 + H_1x + H_0 \qquad (1)$$

đi qua hai điểm (x_0, y_0), (x_1, y_1) và có các đạo hàm là y'_0, y'_1. Ta tìm các hệ số H_i bằng cách giải hệ phương trình:

$$\begin{cases} h(x_0) = H_3x_0^3 + H_2x_0^2 + H_1x_0 + H_0 = y_0 \\ h(x_1) = H_3x_1^3 + H_2x_1^2 + H_1x_1 + H_0 = y_1 \\ h'(x_0) = 3H_3x_0^2 + 2H_2x_0 + H_1 = y'_0 \\ h'(x_1) = 3H_3x_1^2 + 2H_2x_1 + H_1 = y'_1 \end{cases} \qquad (2)$$

Các đạo hàm bậc nhất được tính gần đúng bằng:

$$\begin{aligned} y'_0 &= \frac{h(x_0+\varepsilon) - h(x_0)}{\varepsilon} = \frac{y_2 - y_0}{\varepsilon} \\ y'_1 &= \frac{h(x_1) - h(x_1-\varepsilon)}{\varepsilon} = \frac{y_1 - y_3}{\varepsilon} \end{aligned} \qquad (3)$$

Bây giờ ta tìm đa thức nội suy Lagrange hay Newton đi qua 4 điểm:

(x_0, y_0), $(x_2 = x_0 + \varepsilon, y_2 = y_0 + y'_0\varepsilon)$, $(x_3 = x_1 - \varepsilon, y_3 = y_1 - y'_1\varepsilon)$, (x_1, y_1)

Hàm ***hermit()*** tạo nên phương trình (2):

```
function H = hermit(x0, y0, dy0, x1, y1, dy1)
A = [x0^3 x0^2 x0 1; x1^3 x1^2 x1 1;
      3*x0^2 2*x0 1 0; 3*x1^2 2*x1 1 0];
b = [y0 y1 dy0 dy1]'; %Pt.(2)
H = (A \ b)';
```


Hàm ***hermits()*** dùng hàm ***hermit()*** để tính các hệ số của đa thức Hermit trên nhiều đoạn và giá trị nội suy:

```
function [H,yi]  = hermits(x, y, dy, xi)
% Tim cac he so cua c da thuc Hermite tren c doan
clc
for n = 1:length(x)-1
   H(n,:) = hermit(0, y(n), dy(n), x(n + 1)-x(n), y(n + 1), dy(n + 1));
```

```
end
yi = ppval(mkpp(x, H),xi)
```

Để nội suy ta dùng chương trình ***cthermite.m***:

```
clear all, clc
x = [0  1  2  3];
y = [1  2  4  5];
dy = [0  2  4  6];
[h, y] = hermits(x, y, dy, 1.5)
```

§8. BIẾN ĐỔI FOURIER

1. Biến đổi Fourrier: Tín hiệu thực tế thường bao gồm các thành phần có tần số khác nhau. Chuỗi Fourier và phép bíến đổi Fourier là công cụ toán học dùng để phân tích đặc tính tần số của tín hiệu. Có 4 định nghĩa tương tự nhau về chuỗi và phép biến đổi Fourier, gồm: chuỗi Fourier liên tục theo t(CFS), phép biến đổi Fourier liên tục theo t(CFT), chuỗi Fourier gián đoạn theo t(DFS) và phép biến đổi Fourier gián đoạn theo t(DFT). Trong các công cụ này, DFT dễ dàng lập trình trên máy tính nên trong phần này ta sẽ chú ý đến nó.

Giả sử chuỗi số liệu { x[n] = x(nT), n = 0 : M - 1} với T là chu kì lấy mẫu có được bằng cách lấy mẫu một tín hiệu liên tục x(t) T lần trong một giây. N cặp điểm DFT và iDFT được định nghĩa bằng:

DFT: $$X(k)=\sum_{n=0}^{N-1}x[n]e^{-j2\pi nk/N} \tag{1a}$$

iDFT: $$x[n]=\frac{1}{N}\sum_{n=0}^{N-1}X(k)e^{j2\pi nk/N} \tag{1b}$$

Nói chung hệ số DFT của X(k) là một số phức và nó xác định biên độ và pha của thành phần tín hiệu có tần số số $\Omega_k = k\Omega_0$(rad), tương ứng với tần số tương tự $\omega_k = k\omega_0 = k\Omega_0/T = 2\pi k/NT$ (rad/s). Ta gọi $\Omega_0 = 2\pi/N$ và $\omega_0 = 2\pi/NT$ là các tần số cơ bản số và tương tự (tần số phân giải) vì đây là hiệu tần số có thể phân biệt bởi N điểm DFT.

DFT và DFS có cùng bản chất nhưng khác nhau về phạm vi thời gian/tần số. Cụ thể là tín hiệu x[n] và DFT X[k] của nó kéo dài hữu hạn trên phạm vi thời gian/tần số {0 ≤ n ≤ N-1} và {0 ≤ k ≤ N-1}. Tín hiệu x[n] được phân tích bởi DFS và DFS của nó X(k) là chu kì tín hiệu với chu kì N trên toàn bộ tập số nguyên.

Biến đổi Fourier nhanh FFT là thuật toán hiệu quả để tính DFT và iDFT được xây dựng bằng cách dùng tính chu kì và tính đối xứng cuả nhân tử $e^{i2\pi nk/N}$ để giảm bớt số nhân tử phức từ N^2 thành $(N/2)\log_2 N$)N thể hiện kích thước của DFT. Hàm MATLAB fft() và ifft() thực hiện thuật toán đối với N = 2^l (l là số nguyên không âm). Nếu độ dài M của chuỗi số liệu ban đầu không phải là bội số của 2, có thể mở rộng bằng cách đệm thêm số 0 vào cuối chuỗi và gọi là đệm zero.

Ta xem xét hiệu qủa này bằng cách thực hiện đoạn lệnh trong ***ctcompdftfft.m***.

```
%So sanh phep bien doi Fourier nhanh va roi rac
clear, clf
N = 2^10;
n = [0:N - 1];
x = cos(2*pi*200/N*n)+ 0.5*sin(2*pi*300/N*n);
tic %ngung dong ho
for k = 0:N - 1
   X(k+1) = x*exp(-j*2*pi*k*n/N).';
end %DFT
k = [0:N - 1];
for n = 0:N - 1
   xr(n + 1) = X*exp(j*2*pi*k*n/N).';
end %IDFT
time_dft = toc
plot(k,abs(X))
pause, hold on
tic
X1 = fft(x); %FFT
xr1 = ifft(X1); %IFFT
time_fft = toc %dua ra thoi gian thuc hien
clf, plot(k,abs(X1),'r') %pho bien do
```

Chạy đoạn lệnh và so sánh thời gian thực hiện 1024 điểm tính DFT/iDFT và FFT/iFFT.

2. Ý nghĩa vật lý của biến đổi Fourrier rời rạc: Để hiểu được ý nghĩa vật lí của FFt ta thực hiện các lệnh trong chương trình ***ctmeanning.m***. Chương trình cho ta phổ biên độ của tín hiệu

$$x(t) = \sin(1.5\pi t) + 0.5\cos(3\pi t) \qquad (2)$$

được lấy mẫu mỗi T s.

Từ các kết quả ta thấy khi T = 0.1 và N = 32 thì $X_a(k)$ lớn tại k = 2 và k= 5. Lúc đó $k\omega_0 = 2\pi k/NT = 2\pi k/3.2 \approx 1.5\pi$ và $3.125\pi \approx 3\pi$.

Khi T = 0.05 và N = 64 thì $X_b(k)$ cũng lớn tại k = 2 và k = 5. Lúc đó $k\omega_0 = 1.25\pi \approx 1.5\pi$ và $3.125\pi \approx 3\pi$.

Khi T = 0.1 và N = 64 thì $X_c(k)$ lớn tại k = 4 ,k = 5, k = 9 và k = 10. Lúc đó $k\omega_0 = 2\pi k/NT = 2\pi k/6.4 \approx 1.25\pi \sim 1.5625\pi$ và $2.8125\pi \sim 3\pi$.

Khi T = 0.1 và N = 64 thì $X_d(k)$ lớn tại k = 5 và k = 10. Lúc đó $k\omega_0 = 1.5625\pi \approx 1.5\pi$ và $3.125\pi \approx 3\pi$.

Tồn tại nhiều phổ DFT khác nhau của cùng một tín hiệu tương tự, tuỳ thuộc vào kích thước DFT, chu kì lấy mẫu, khoảng biến thiên của hàm và đệm zero. So sánh với phổ tại T = 0.1s, phổ tại T = 0.05s có phạm vi tần số tương tự $[0, 2\pi/T_b]$ rộng hơn nhưng có cùng tần số phân giải tương tự là $\omega_0 = \Omega_0/T_b = 2\pi/N_bT_b = \pi/1.6 = 2\pi/N_aT_a$. Phổ khi có đệm zero trơn.


```
clear, clf
w1 = 1.5*pi;
w2 = 3*pi;
N = 32;
n = [0:N - 1];
T = 0.1; %chu ki lay mau
t = n*T;
xan = sin(w1*t) + 0.5*sin(w2*t);
subplot(421)
stem(t,xan,'.')
k = 0:N - 1;
Xa = fft(xan);
dscrp=norm(xan-real(ifft(Xa)))
subplot(423)
stem(k,abs(Xa),'.')
N = 32;
n = [0:N - 1];
```

```
T = 0.1;
t = n*T;
xan = sin(w1*t) + 0.5*sin(w2*t);
subplot(422)
stem(t,xan,'.')
k = 0:N - 1;
Xa = fft(xan);
Dscrp = norm(xan - real(ifft(Xa)))
subplot(424)
stem(k, abs(Xa),'.')
N = 64;
n = [0:N - 1];
T = 0.05;
t = n*T;
xbn = sin(w1*t) + 0.5*sin(w2*t);
subplot(425)
stem(t,xbn,'.')
k = 0:N - 1;
Xb = fft(xbn);
subplot(427)
stem(k,abs(Xb),'.')
N = 64;
n = [0:N-1];
T = 0.1;
t = n*T;
xbn = sin(w1*t) + 0.5*sin(w2*t);
subplot(426)
stem(t, xbn,'.')
k = 0:N - 1;
Xb = fft(xbn);
subplot(428)
stem(k, abs(Xb),'.')
```


Ta có nhiều phổ DFT cho cùng một tín hiệu tương tự, tuỳ thuộc vào kích thước DFT, chu kì lấy mẫu, khoảng lấy mẫu và đệm zero. So sánh phổ khi giảm chu kì lấy mẫu T từ 0.1s đến 0.05s

3. Nội suy bằng các dùng biến đổi Fourrier rời rạc: Ta dùng DFS/DFT để nội suy dãy x[n] nhận được từ kết quả lấy mẫu tín hiệu ở khoảng cách cách đều. Thủ tục gồm hai bước: lấy N điểm FFT X(k) của x[n] và dùng công thức:

$$\hat{x}(t) = \frac{1}{N}\sum_{|k|<N/2}\tilde{X}(k)e^{j2\pi kt/NT}$$
$$= \frac{1}{N}\left\{X(0) + 2\sum_{k=1}^{N/2-1}\mathrm{Real}\left[X(k)e^{j2\pi kt/NT}\right] + X(N/2)\cos(\pi/T)\right\} \qquad (5)$$

Ta xây dựng hàm nội suy ***interpdfs()***:

```
function [xi,Xi] = interpdfs(T, x, Ws, ti)
%T : chu li lay mau
%x : thu tu roi rac hoa
%Ws: tan so dung chuan (1.0 = pi[rad])
%ti: khoang thoi gian noi suy
if nargin < 4
   ti = 5;
end
if nargin < 3 | Ws > 1
   Ws = 1;
end
N = length(x);
if length(ti) == 1
   ti = 0:T/ti:(N-1)*T; %khoang con duoc chia cho ti
end
ks = ceil(Ws*N/2);
Xi = fft(x);
Xi(ks + 2:N - ks) = zeros(1,N - 2*ks - 1); %pho da loc
xi = zeros(1,length(ti));
for k = 2:N/2
   xi = xi+Xi(k)*exp(j*2*pi*(k - 1)*ti/N/T);
end
xi = real(2*xi+Xi(1)+Xi(N/2+1)*cos(pi*ti/T))/N; %pt.(.5)
```

Để nội suy ta dùng chương trình ***ctfourier.m***:

```
clear, clf
```

```
w1 = pi;
w2 = .5*pi; %hai tan so
N = 32;
n = [0:N - 1];
T = 0.1;
t = n*T;
x = sin(w1*t)+0.5*sin(w2*t)+(rand(1,N) - 0.5); %0.2*sin(20*t);
ti = [0:T/5:(N - 1)*T];
subplot(411), plot(t,x,'k.') %so lieu ban dau
title('So lieu ban dau va ket qua noi suy')
[xi,Xi] = interpdfs(T,x,1,ti);
hold on, plot(ti,xi,'r') %tai tao tin hieu
k = [0:N - 1];
subplot(412), stem(k,abs(Xi),'k.') %pho ban dau
title('Pho ban dau')
[xi,Xi] = interpdfs(T,x,1/2,ti);
subplot(413), stem(k,abs(Xi),'r.') %pho da  loc
title('Pho da loc')
subplot(414), plot(t,x,'k.', ti,xi,'r') %tin hieu da loc
title('Tin hieu da loc')
```

§9. XẤP XỈ HÀM BẰNG PHƯƠNG PHÁP BÌNH PHƯƠNG BÉ NHẤT

1. Khái niệm chung: Trong các mục trước ta đã nội suy giá trị của hàm. Bài toán đó là cho một hàm dưới dạng bảng số và phải tìm giá trị của hàm tại một giá trị của đối số không nằm trong bảng.

Trong thực tế, bên cạnh bài toán nội suy ta còn gặp một dạng bài toán khác. Đó là tìm công thức thực nghiệm của một hàm.

Nội dung bài toán là từ một loạt các điểm cho trước (có thể là các giá trị của một phép đo nào đó) ta phải tìm một hàm xấp xỉ các giá trị đã cho. Ta sẽ dùng phương pháp bình phương tối thiểu để giải bài toán.

Giả sử có mẫu quan sát (x_i, y_i) của hàm y = f(x). Ta chọn hàm f(x) có dạng:

$$f(x) = a_0f_0(x) + a_1f_1(x) + a_2f_2(x)... \quad (1)$$

Trong đó các hàm $f_0(x)$, $f_1(x)$, $f_2(x)$ v.v. là (m+1) hàm độc lập tuyến tính mà ta có thể chọn tuỳ ý và các hệ số a_i là tham số chưa biết mà ta phải xác định dựa vào hệ hàm đã chọn và các điểm quan sát. Sai số giữa trị đo được và trị tính theo (1) là :

$$e_i = y_i - f(x_i) \qquad (2)$$

Sai số này có thể âm hay dương tuỳ từng giá trị của y_i. Khi dùng phương pháp bình phương bé nhất ta xét bình phương của sai số tại một điểm:

$$e_i^2 = \left[y_i - f(x_i)\right]^2 \qquad (3)$$

Với n điểm tổng bình phương của sai số sẽ là :

$$S = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left\{y_i - \left[a_0 f_0(x_i) + a_1 f_1(x_i) + \cdots + a_m f_m(x_i)\right]\right\}^2$$

Rõ ràng S là hàm của các giá trị cần tìm a_i và chúng ta sẽ chọn các a_i sao cho S đạt giá trị min, nghĩa là các đạo hàm $\dfrac{\partial S}{\partial a_i}$ phải bằng không.

Ta sẽ xét các trường hợp cụ thể.

2. Hàm xấp xỉ có dạng đa thức: Trong trường hợp tổng quát ta chọn hệ hàm xấp xỉ là một đa thức, nghĩa là:

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$$

Vậy hàm S là :

$$S = \left(y_i - a_0 + a_1x + a_2x^2 + \cdots + a_mx^m\right)^2$$

Theo điều kiện đạo hàm $\dfrac{\partial S}{\partial a_i} = 0$ ta nhận được hệ phương trình:

$$\begin{cases}
a_m \sum_{i=1}^{n} x_i^m + a_{m-1} \sum_{i=1}^{n} x_i^{m-1} + \cdots + n a_0 = \sum_{i=1}^{n} y_i \\
a_m \sum_{i=1}^{n} x_i^{m+1} + a_{m-1} \sum_{i=1}^{n} x_i^{m} + \cdots + a_0 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \\
a_m \sum_{i=1}^{n} x_i^{m+2} + a_{m-1} \sum_{i=1}^{n} x_i^{m+1} + \cdots + a_0 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i^2 y_i \\
a_m \sum_{i=1}^{n} x_i^{m+3} + a_{m-1} \sum_{i=1}^{n} x_i^{m+2} + \cdots + a_0 \sum_{i=1}^{n} x_i^3 = \sum_{i=1}^{n} x_i^3 y_i \\
\cdots \\
a_m \sum_{i=1}^{n} x_i^{2m} + a_{m-1} \sum_{i=1}^{n} x_i^{2m-1} + \cdots + a_0 \sum_{i=1}^{n} x_i^m = \sum_{i=1}^{n} x_i^m y_i
\end{cases}$$

Đây là một hệ phương trình tuyến tính. Giải nó ta nhận được các gía trị a_i.

Ta xây dựng hàm ***polynomfit()*** thực hiện thuật toán trên:

```
function x = polyfits(xData, yData, m)
%Dung de tinh he so cua da thuc xap xi
% Cu phap: x = polyfits(xData, yData, m)
m = m+1;
A = zeros(m);
b = zeros(m, 1);
s = zeros(2*m-1, 1);
for i = 1:length(xData)
    temp = yData(i);
    for j = 1:m
        b(j) = b(j) + temp;
        temp = temp*xData(i);
    end
    temp = 1;
    for j = 1:2*m-1
        s(j) = s(j) + temp;
        temp = temp*xData(i);
    end
end
for i = 1:m
    for j = 1:m
        A(i, j) = s(i+j-1);
    end
end
x = A\b;
% Sap xep lai he so tu so mu cao nhat
x = flipdim(x, 1);
```


Để xấp xỉ một dãy số liệu bằng hàm đa thức ta dùng chương trình ***ctpolynomfit.m***:

```
clear all, clc
xData = [0 1 2 3 4];
yData = [1  8 24  63 124];
x = polyfits(xData, yData, 3);
y = 0:0.1:4;
```

```
z = polyval(x', y);
hold on
plot(y, z,'-b', xData, yData, 'ro');
```

3.Hàm dạng Ae^{cx}: Khi các số liệu thể hiện một sự biến đổi đơn điệu ta dùng hàm xấp xỉ là $y = Ae^{cx}$. Lấy logarit hai vế ta có :

$$\ln y = \ln A + cx\ln e$$

Theo điều kiện đạo hàm $\frac{\partial S}{\partial a_i} = 0$ ta có hệ phương trình :

$$\begin{cases} c\sum_{i=1}^{n} x_i + n\ln A = \sum_{i=1}^{n} \ln y_i \\ c\sum_{i=1}^{n} x_i^2 + \ln A\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i \ln y_i \end{cases}$$

Giải hệ phương trình này ta có các hệ số A và c.

Ta xây dựng hàm ***expfit()*** để xấp xỉ

```
function [c,A] = expfit(x, y)
a = sum(x);
b = size(x,2);
c = sum(log(y));
d = sum(x.^2);
e = sum(x.*log(y));
d1 = a*a - d*b;
d2 = c*a - e*b;
d3 = a*e - c*d;
c = d2/d1;
A = exp(d3/d1);
```

Ta dùng chương trình ***ctexpfit.m*** để xấp xỉ dãy số liệu đã cho

```
clear all, clc
x = [1.2  2.8  4.3    5.4      6.8    7.9];
y = [7.5        16.1   38.9   67       146.6 266.2];
[c, A] = expfit(x, y);
t = 0:0.1:8;
z = A*exp(c*t);
plot(t, z, '-b', x, y, 'ro');
```

4. Hàm dạng Ax^q: Khi các số liệu thể hiện một sự biến đổi đơn điệu ta cũng có thể dùng hàm xấp xỉ là y = Ax^q. Lấy logarit hai vế ta có:

$$\ln y = \ln A + q\ln x$$

Theo điều kiện đạo hàm triệt tiêu ta có hệ phương trình :

$$\begin{cases} q\sum\limits_{i=1}^{n}\ln x_i + n\ln A = \sum\limits_{i=1}^{n}\ln y_i \\ q\sum\limits_{i=1}^{n}\ln^2 x_i + \ln A\sum\limits_{i=1}^{n}\ln x_i = \sum\limits_{i=1}^{n}\ln x_i \ln y_i \end{cases}$$

Giải hệ phương trình này ta có các hệ số
A và q.
Ta xây dựng hàm ***powerfit()*** để xấp xỉ:

```
function [q, A] = powerfit(x, y)
a = sum(log(x));
b = size(x, 2);
c = sum(log(y));
d = sum(log(x).^2);
e = sum(log(x).*log(y));
d1 = a*a - d*b;
d2 = c*a - e*b;
d3 = a*e - c*d;
q = d2/d1;
A = exp(d3/d1);
```


Ta dùng chương trình ***ctpowerfit.m*** để xấp xỉ dãy số liệu đã cho:

```
clc
x = [   1       2       3       4     5];
y = [1.5       15.1      52.5    130.5       253];
[q,A] = powerfit(x, y)
t = 0.1:0.1:5;
z = exp(log(A)+q*log(t));
plot(t, z, '-b', x, y, 'ro');
```


5. Hàm lượng giác: Khi quan hệ y = f(x) có dạng tuần hoàn ta dùng hàm xấp xỉ là tổ hợp tuyến tính của các hàm sin và cosin dạng:

$$f(x) = a_0 + \sum_{i=1}^{n} a_i \cos(i\omega x) + \sum_{i=1}^{n} b_i \sin(i\omega x)$$

Để đơn giản trước hết ta xét hàm chỉ có một số hạng sin-cos, nghĩa là :

$$f(x) = a_0 + a_1 \cos\omega x + b_1 \sin\omega x$$

Hàm S sẽ có dạng :

$$S = \sum_{i=1}^{n} \left[y_i - (a_0 + a_1\cos\omega x + b_1\sin\omega x) \right]^2$$

Theo điều kiện đạo hàm triệt tiêu ta có hệ phương trình đối với các hệ số dạng:

$$\begin{bmatrix} n & \sum\cos\omega x_i & \sum\sin\omega x_i \\ \sum\cos\omega x_i & \sum\cos^2\omega x_i & \sum\cos\omega x_i\sin\omega x_i \\ \sum\sin\omega x_i & \sum\cos\omega x_i\sin\omega x_i & \sum\sin^2\omega x_i \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i\cos\omega x_i \\ \sum y_i\sin\omega x_i \end{bmatrix}$$

Do:

$$\frac{\sum\sin\omega x_i}{n} = 0 \qquad \frac{\sum\cos\omega x_i}{n} = 0$$

$$\frac{\sum\sin^2\omega x_i}{n} = \frac{1}{2} \qquad \frac{\sum\cos^2\omega x_i}{n} = \frac{1}{2}$$

$$\frac{\sum\cos\omega x_i\sin\omega x_i}{n} = 0$$

nên hệ phương trình có dạng đơn giản :

$$\begin{bmatrix} n & 0 & 0 \\ 0 & n/2 & 0 \\ 0 & 0 & n/2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i\cos\omega x_i \\ \sum y_i\sin\omega x_i \end{bmatrix}$$

Giải hệ ta có :

$$a_0 = \frac{\sum y_i}{n} \qquad a_1 = \frac{2}{n}\sum y_i\cos\omega x_i \qquad b_1 = \frac{2}{n}\sum y_i\sin\omega x_i$$

Trong trường hợp tổng quát, một cách tương tự ta có:

$$a_0 = \frac{\sum y}{n} \qquad a_i = \frac{2}{n}\sum y\cos i\omega x \qquad b_i = \frac{2}{n}\sum y\sin i\omega x$$

Ta xây dựng hàm ***sinfit()*** để xấp xỉ:

```
function [a, b, c, omega] = sinfit(x, y, T)
%T la chu ki
omega = 2*pi/T;
n = size(x,2);
```

```
a = sum(y)/n;
b = (2/n)*sum(y.*cos(omega*x));
c = (2/n)*sum(y.*sin(omega*x));
```

Ta dùng chương trình ***ctsinfit.m*** để tính:

```
c ear all, clc
x = [0 0.15   0.3     0.45   0.6     0.75   0.9     1.05   1.2     1.3];
y = [2.2         1.5951.031 0.722 0.786 1.2     1.81   2.369 2.678 2.614];
T = 1.5;
[a, b, c, omega] = sinfit(x, y, T)
t = 0.:0.01:1.5;
z = a + b*cos(omega*t) + c*sin(omega*t);
plot(t, z,'-b', x, y, 'ro');
```

6. Hàm hữu tỉ: Khi quan hệ y = f(x) có dạng đường cong bão hoà hay dạng arctan, tan v.v ta dùng hàm xấp xỉ là hàm hữu tỉ dạng đơn giản:

$$y = \frac{ax}{b+x}$$

Lấy nghịch đảo của nó ta có :

$$\frac{1}{y} = \frac{b}{a}\frac{1}{x} + \frac{1}{a}$$

Đặt 1/y = Y, 1/x = X, b/a = B và 1/a = A phương trình trên sẽ có dạng:

$$Y = A + BX$$

và là một đa thức bậc một. Do vậy ta có hệ phương trình đối với các hệ số A và B là:

$$\begin{cases} nA + B\sum_{i=1}^{n}\frac{1}{x_i} = \sum_{i=1}^{n}\frac{1}{y_i} \\ A\sum_{i=1}^{n}\frac{1}{x_i} + B\sum_{i=1}^{n}\frac{1}{x_i^2} = \sum_{i=1}^{n}\frac{1}{x_i y_i} \end{cases}$$

và từ đó tính được a và b.

Ta xây dựng hàm ***racfit()*** để xấp xỉ:

```
function [a, b] = racfit(x, y)
a1 = size(x, 2);
b1 = sum(1./x);
c1 = sum(1./y);
```

```
d1 = sum(1./x.^2);
e1 = sum((1./x).*(1./y));
del = a1*d1 - b1*b1;
del1 = c1*d1 - e1*b1;
del2 = a1*e1 - b1*c1;
A = del1/del;
B = del2/del;
a = 1/A;
b = B/A;
```

Để xấp xỉ ta dùng chương trình ***ctracfit.m***:

```
clear all, clc
x = [1  2  3  4  5];
y = [0.3333333        0.5    0.6        0.66666      0.7142857];
[a, b] = racfit(x, y)
t = 0.:0.01:5;
z = a*t./(b+t)
plot(t, z,'-b', x, y, 'ro');
```