HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Chủ đề 20: Nội suy Newton

Nguyễn Văn Nghiêm - 20206206

ONE LOVE. ONE FUTURE.

1. Bài toán Nội suy

- Trong thực tế, ta thường gặp những hàm số không biết rõ cụ thể biểu thức giải tích của chúng hoặc quá phức tạp. Bằng việc đo đạc, thực nghiệm ta thu được bảng số với các điểm y_i tại các điểm x_i tương ứng thuộc đoạn $[a, b]$ nào đó, trong khi ta muốn biết giá trị y tại các điểm $x \neq x_i$.

- Việc tìm chính xác biểu thức của hàm f là điều không thể. Tuy nhiên chúng ta có thể dựa vào các điểm (x, y) đã có để "xấp xỉ" hàm f với sai số chấp nhận được. Tư tưởng này của bài toán chính là các bài toán nội suy - xấp xỉ hàm.

1. Bài toán Nội suy

- Bảng số:
- $$f(x_i) = y_i \left(x_i \epsilon [a,b], i = \overline{0,n}\right) \ (1)$$
- Ta cần tìm giá trị của y tại điểm $\overline{x} \neq x_i \ (i = \overline{0,n})$
- Từ bảng số xây dựng $P_n(x_i) = y_i$ với $i = \overline{0,n}$ ($P_n(x_i)$ bậc n) (2)
- Việc thay hàm $f(x)$ bằng một hàm $P_n(x)$ đơn giản hơn sao cho sự sai lệch của $f(x)$ và $P_n(x)$ là chấp nhận được gọi là xấp xỉ hàm
- Đa thức $P_n(x)$ sinh ra từ bảng số (1) thỏa mãn (2) gọi là đa thức nội suy.

1. Bài toán Nội suy

- Các điểm $x_i \epsilon [a, b], i = \overline{0, n}$ là các mốc nội suy
- Đặt $\overline{y} = P_n(x) \approx f(\overline{x})$ với $\overline{x} \neq x_i, i = \overline{0, n}$ là giá trị nội suy nếu $\overline{x} \epsilon [a, b]$; gọi là giá trị ngoại suy nếu $\overline{x}$ nằm ngoài $[a, b]$

1. Bài toán Nội suy

- Sự duy nhất của đa thức nội suy:

Với bộ điểm $\{x_i, y_i = f(x_i)\}\ i = \overline{0, n}, x_i \neq x_j\ \forall\ i\ \neq j$ cho trước, đa thức nội suy tồn tại và duy nhất.

- Sự duy nhất của đa thức nội suy:

$* \ Chứng\ minh$:

$$P_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

$$P_n(x_i) = \ y_i \text{ với } i = \overline{0,n} \Leftrightarrow \begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = y_o \\ a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n = y_1 \\ \qquad\qquad\qquad \cdots \\ a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = y_n \end{cases} \text{hay}$$

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_1^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Ma trận vuông vế trái là ma trận Vandermonde, có $\det = \prod_{0 \le i < j \le n} (x_i - x_j) \neq 0$

Do đó hệ phương trình tồn tại nghiệm duy nhất.

2. Đa thức nội suy Newton

- Ý tưởng phương pháp:

Xuất pháp từ khai triển Taylor:

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

Với $x_0 = 0$, khai triển Maclaurin:

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k$$

Đạo hàm của hàm số tại điểm x_i được định nghĩa:

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

Newton nghĩa ra ý tưởng thay thế đạo hàm bằng thương giữa hiệu các giá trị với hiệu các x tương ứng trên bảng giá trị

- Tỷ hiệu (tỷ sai phân):

Xét hàm số $y = f(x); x \epsilon [a, b]$

**Định nghĩa:* Từ bảng số $y_i = f(x_i)$ $i = \overline{0, n}$ trong đó các mốc nội suy là: $a \equiv x_0 < x_1 < x_2 < \cdots < x_n \equiv b$

Ta gọi

$$f[x_{i-1}, x_i] = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \qquad i = \overline{1, n}$$

là tỷ hiệu cấp một của hàm $f(x)$

Tỷ hiệu của tỷ hiệu cấp 1 là tỷ hiệu cấp 2, ký hiệu là

$$f[x_{i-1}, x_i, x_{i+1}] = \frac{f[x_i, x_{i+1}] - f[x_{i-1}, x_i]}{x_{i+1} - x_{i-1}} \qquad i = \overline{1, n-1}$$

- Tỷ hiệu (tỷ sai phân):

Tỷ hiệu của tỷ hiệu cấp $n-1$ là tỷ hiệu cấp n và ký hiệu

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

Tỷ hiệu cấp n cần $n+1$ mốc

x	$f(x)$	$f[\ldots,\ldots]$	$f[\ldots,\ldots,\ldots]$	…	$f[x_0, x_1, \ldots, x_n]$
x_0	$f(x_0)$			$\ldots$	
x_1	$f(x_1)$	$f[x_0, x_1]$		$\ldots$	
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	$\ldots$	
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$\ldots$	
…	$\ldots$	$\ldots$	$\ldots$	$\ldots$	
x_{n-2}	$f(x_{n-2})$	$f[x_{n-3}, x_{n-2}]$	$f[x_{n-4}, x_{n-3}, x_{n-2}]$	$\ldots$	
x_{n-1}	$f(x_{n-1})$	$f[x_{n-2}, x_{n-1}]$	$f[x_{n-3}, x_{n-2}, x_{n-1}]$	$\ldots$	
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$\ldots$	$f[x_0, \ldots, x_n]$

2. Đa thức nội suy Newton

- Tính chất của tỷ hiệu:

$$f[x_0, x_1, \ldots, x_k] = \sum_{i=0}^{k} \frac{f(x_i)}{\omega'_{k+1}(x_i)}$$

$$\omega_{k+1}(x) = (x - x_0)(x - x_1)\ldots(x - x_k)$$

$$\omega'_{k+1}(x_i) = (x_i - x_0)\ldots(x_i - x_{i-1})(x_i - x_{i+1})\ldots(x_i - x_k)$$

Chứng minh

- Với $k = 1$

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

- Với $k = 2$

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

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \left(\frac{1}{x_1 - x_2} - \frac{1}{x_1 - x_0}\right) . \frac{f(x_1)}{x_2 - x_0} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}$$

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} = \sum_{i=0}^{2} \frac{f(x_i)}{\omega_3'(x_i)}$$

2. Đa thức nội suy Newton

- Giả sử đúng với $k = m$

$$f[x_0, x_1, \ldots, x_m] = \sum_{i=0}^{m} \frac{f(x_i)}{\omega'_{m+1}(x_i)}$$

- Cần chứng minh đúng với $k = m + 1$

$$f[x_0, x_1, \ldots, x_{m+1}] = \frac{f[x_1, \ldots, x_{m+1}] - f[x_0, x_1, \ldots, x_m]}{x_{m+1} - x_0}$$

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

$$= \frac{f(x_0)}{\omega'_{m+2}(x_0)} + \frac{\sum_{i=1}^{m} \left(\frac{x_i . f(x_i) - x_0 . f(x_i)}{\omega'_{m+1}(x_i)(x_i - x_{m+1})} - \frac{f(x_i)}{\omega'_{m+1}(x_i)} \right)}{x_{m+1} - x_0} + \frac{f(x_{m+1})}{\omega'_{m+2}(x_i)}$$

$$= \frac{f(x_0)}{\omega'_{m+2}(x_0)} + \sum_{i=1}^{m} \frac{f(x_i)}{\omega'_{m+2}(x_i)} + \frac{f(x_{m+1})}{\omega'_{m+2}(x_i)} \quad = \sum_{i=0}^{m+1} \frac{f(x_i)}{\omega'_{m+2}(x_i)}$$

- Tính chất tuyến tính:

$$(\alpha f+\beta g)[x_0,x_1,\dots,x_k]=\alpha f[x_0,x_1,\dots,x_k]+\beta g[x_0,x_1,\dots,x_k]$$

- Chứng minh:

$$(\alpha f+\beta g)[x_0,x_1,\dots,x_k]=\sum_{i=0}^{k}\frac{(\alpha f+\beta g)(x_i)}{\omega'_{k+1}(x_i)}=\sum_{i=0}^{k}\frac{\alpha f(x_i)+\beta g(x_i)}{\omega'_{k+1}(x_i)}$$

$$=\alpha\sum_{i=0}^{k}\frac{f(x_i)}{\omega'_{k+1}(x_i)}+\beta\sum_{i=0}^{k}\frac{g(x_i)}{\omega'_{k+1}(x_i)}$$

$$=\alpha f[x_0,x_1,\dots,x_k]+\beta g[x_0,x_1,\dots,x_k]$$

- Từ tính chất $f[x_0, x_1, \ldots, x_k] = \sum_{i=0}^{k} \frac{f(x_i)}{\omega'_{k+1}(x_i)}$ có thể thấy rằng tỷ hiệu cấp k của f không phụ thuộc vào trình tự sắp xếp các mốc nội suy

- Tính chất đối xứng:

$$f[x_0, x_1, \ldots, x_k] = f\left[x_{\sigma(0)}, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\right]$$

2. Đa thức nội suy Newton

- Tính chất tỷ hiệu của đa thức:

1) Tỷ hiệu của hằng số thì bằng "0"

Chứng minh:

Nếu $f(x) = C\ (const)$ thì

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{C - C}{x_{i+1} - x_i} = 0$$

2) Tỷ hiệu cấp m của đa thức bậc n có tính chất:

- Nếu $m = n$ thì tỷ hiệu cấp n là hằng số
- Còn $m > n$ thì tỷ hiệu cấp m bằng "0"
- Chứng minh:

Xét hàm: $f(x) = P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n (a_n \neq 0)$

Xét: $h(x) = x^k \qquad k \in \mathbb{N}$

$$h[x_i, x_{i+1}] = \frac{h(x_{i+1}) - h(x_i)}{x_{i+1} - x_i} = \frac{x_{i+1}^k - x_i^k}{x_{i+1} - x_i} = x_{i+1}^{k-1} + x_{i+1}^{k-2}x_i + \ldots + x_i^{k-1}$$

Là đa thức cấp $k - 1$. Tiếp tục xét tỷ hiệu cấp hai, ta được đa thức cấp $k - 2$ ... tỷ hiệu cấp k của $h(x)$ là hằng số, tỷ hiệu cấp $k + 1$ thì bằng "0"

- Đa thức nội suy Newton:

Định lý 1:

$$P_n(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \cdots +$$

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

Là đa thức nội suy tương ứng với các mốc nội suy $y_i = f(x_i)\ i = \overline{0, n}$

- Chứng minh

Gọi $L_k(x)$ là đa thức nội suy Lagrange bậc k có các mốc nội suy $x_i, i = \overline{0, k}$

Ta có: $P_k(x) = L_k(x) - L_{k-1}(x)$ là đa thức bậc k và $P_k(x_i) = 0, i = \overline{0, k-1}$

$$\Rightarrow P_k(x) = A(x - x_0)(x - x_1)...(x - x_{k-1}) = A\omega_k(x)$$

Và $P_k(x_k) = L_k(x_k) - L_{k-1}(x_k) = A\omega_k(x_k)$

2. Đa thức nội suy Newton

$$A = \frac{L_k(x_k) - L_{k-1}(x_k)}{\omega_k(x_k)} = \frac{f(x_k) - L_{k-1}(x_k)}{\omega_k(x_k)}$$

$$= \frac{f(x_k)}{\omega_k(x_k)} + \frac{\omega_k(x_k)\sum_{i=0}^{k-1}\frac{f(x_i)}{(x_k - x_i)\omega'_k(x_i)}}{\omega_k(x_k)} = \frac{f(x_k)}{\omega_k(x_k)} + \sum_{i=0}^{k-1}\frac{f(x_i)}{\omega'_{k+1}(x_i)}$$

Có: $\omega_k(x_k) = (x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1}) = \omega'_{k+1}(x_k)$

$$\Rightarrow A = \frac{f(x_k)}{\omega'_{k+1}(x_k)} + \sum_{i=0}^{k-1}\frac{f(x_i)}{\omega'_{k+1}(x_i)} = \sum_{i=0}^{k}\frac{f(x_i)}{\omega'_{k+1}(x_i)} = f[x_0, x_1, \ldots, x_k]$$

$$P_k(x) = L_k(x) - L_{k-1}(x)$$

$$\Rightarrow L_k\ (x) = P_k(x) + L_{k-1}(x) = A\omega_k(x) + L_{k-1}(x)$$

$$= f[x_0, x_1, \dots, x_k](x - x_0)(x - x_1)\dots(x - x_{k-1}) + L_{k-1}(x)$$

$$= f[x_0, x_1, \dots, x_k](x - x_0)(x - x_1)\dots(x - x_{k-1})$$

$$+ f[x_0, x_1, \dots, x_{k-1}](x - x_0)(x - x_1)\dots(x - x_{k-2})$$

$$+ \dots + f[x_0, x_1](x - x_0) + f(x_0)$$

2. Đa thức nội suy Newton

- Đa thức nội suy Newton:

- Định nghĩa*:*

Công thức (1) được gọi là công thức nội suy Newton, đa thức trong (1) được gọi là đa thức nội suy Newton

- Khi $x_0 < x_1 < x_2 <\ldots< x_n$ thì (1) được gọi là đa thức nội suy Newton tiến

- Khi $x_0 > x_1 > x_2 >\ldots> x_n$ thì (1) được gọi là đa thức nội suy Newton lùi

2. Đa thức nội suy Newton

- Công thức (1) sử dụng các yếu tố đầu bảng của bảng tỷ hiệu còn được gọi là đa thức nội suy Newton tiến xuất phát từ x_0

- Nếu các mốc nội suy đánh số lại theo thứ tự:

$$x_n, x_{n-1}, \dots, x_1, x_0$$

$$P_n(x) = f(x_n) + (x - x_n)f[x_n, x_{n-1}] + (x - x_n)(x - x_{n-1})f[x_n, x_{n-1}, x_{n-2}] + \cdots + \\ + (x - x_n)(x - x_{n-1})\dots(x - x_1)f[x_n, x_{n-1}, .., x_0] \quad (2)$$

- Công thức (2) sử dụng các yếu tố cuối bảng của bảng tỷ hiệu được gọi là đa thức nội suy Newton lùi xuất phát từ x_n

Định lý 2:

- $R_n(x) = f(x) - P_n(x)$
- $x \neq x_i, i = \overline{0, n} \Rightarrow R_n(x) = f[x, x_0, x_1, x_2, ..., x_n]\omega_{n+1}(x)$
- Chứng minh

$$f[x, x_0] = \frac{f(x) - f(x_0)}{x - x_0} \Rightarrow f(x) = f(x_0) + (x - x_0)f[x, x_0]$$

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

$$\Rightarrow f(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x, x_0, x_1]$$

....

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

$$= P_n(x) + R_n(x)$$

- Sai số:

Công thức đánh giá sai số:

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

2. Đa thức nội suy Newton

- Nhận xét:

  - Chỉ nên sử dụng đa thức nội suy để tính gần đúng $x \epsilon [a, b]$, nằm ngoài $[a, b]$ sẽ có sai số rất lớn

  - Với $n + 1$ mốc nội suy cho trước, ta sẽ chỉ thu được một đa thức nội suy và một sai số cố định. Để có được đa thức nội suy với sai số chấp nhận được thì chúng ta chỉ có cách là xử lí dữ liệu đầu vào, tối ưu mốc nội suy

  - Với đa thức nội suy Newton, chúng ta không cần phải sắp xếp các mốc nội suy, cũng như chúng ta có thể xuất phát từ mốc x_k bất kì.

2. Đa thức nội suy Newton

- Nhận xét:

  - Đa thức nội suy Newton cũng chính là đa thức nội suy Lagrange chỉ khác về cách trình bày

  - Nếu thêm mốc nội suy (x_{n+1}, y_{n+1}) đa thức $P_{n+1}(x)$ được tính như sau:

$$P_{n+1}(x) = P_n(x) + (x - x_0)(x - x_1)\ldots(x - x_n)f[x_0, x_1, \ldots, x_{n+1}]$$

  - Giải quyết nhược điểm của phương pháp nội suy Lagrange: Khi bổ sung thêm mốc nội suy ta phải tính toán lại từ đầu chứ không thể dùng kết quả cũ để tính toán với số lượng phép toán ít hơn.

  - Khối lượng phép toán: $\dfrac{1}{2}(7n^2 + 11n)$

2. Đa thức nội suy Newton

- Thuật toán:

Các gói sử dụng:

1. $main$: thuật toán chính
2. $Input$: nhập x, y từ file
3. $BuildBTH$: kết nạp mốc $x[i]$ vào bảng tỷ hiệu
4. $BuildBTT$: kết nạp mốc $x[i]$ vào bảng tính tích theo lược đồ Hoocner
5. fx: kết nạp mốc $x[i]$ vào đa thức f
6. $Newton\_interpolation\_Forward$: xây dựng đa thức nội suy Newton tiến
7. $Newton\_interpolation\_Backward$: xây dựng đa thức nội suy Newton lùi
8. $fx0$: ước tính giá trị tại $x0$ theo lược đồ Hoocner
9. add_Newton: thêm mốc nội suy
10. $plot$: vẽ đồ thị

- $main$:

Input: file các mốc nội suy, điểm cần ước lượng giá trị: $x0$, số lượng thêm mốc: k

Output: hệ số đa thức: f, giá trị ước lượng tại $x0$: $value$, đồ thị đa thức f

$x, y \leftarrow Input()$

$n \leftarrow \text{len}(x) - 1$

In các mốc nội suy x

Khởi tạo các ma trận không:

$BTH(n+1), BTT(n+1), f(n+1)$

$BTH[0], BTT[0], f[0] \leftarrow y[0], 1, y[0]$

$f, BTH, BTT \leftarrow Newton\_interpolation\_Forward(BTH, BTT, f, x, y, n)$

$value \leftarrow fx0(f, n, x0)$

In hệ số đa thức f và giá trị $value$

$plot(x, y, f, n)$

if $k\ \ <=\ \ 0$ **then**: Dừng chương trình

end if

for $i = 1 \rightarrow k$ **do**

Nhập mốc và giá trị nội suy: $\{xt, yt\}$

if xt **not in** x **then**

$n \leftarrow n\ +\ 1$

append $\{xt, yt\}$ vào $\{x, y\}$

$f, BTH, BTT\ \ \leftarrow add\_Newton(BTH, BTT, f, x, y, n)$

end if

end for

In hệ số đa thức f và giá trị $value$

$plot(x, y, f, n)$

- $BuildBTH$:

Input: BTH, x, y, i

Output: TH_row_i

Khởi tạo ma trận không $TH\_row\_i(\text{len}(x))$

$TH\_row\_i[0] \leftarrow y[i]$

for $j = 1 \rightarrow i$ **do**

$TH\_row\_i[j] \leftarrow (TH\_row\_i[j-1] - BTH[j-1]) \,/\, (x[i] - x[i-j])$

end for

return TH_row_i

- $BuildBTT$:

Input: BTT, x, i

Output: TT_row_i

Khởi tạo ma trận không $TT\_row\_i(len(x))$

$TT\_row\_i[i\ +\ 1]\ \leftarrow 1$

for $j = i\ \rightarrow 1$ **do**

$TT\_row\_i[j] \leftarrow BTT[j - 1] - x[i] * BTT[j]$

end for

$TT\_row\_i[0] \leftarrow -x[i] * BTT[0]$

return TT_row_i

2. Đa thức nội suy Newton

- fx:

Input: BTH, BTT, i, f

Output: f

for $j = 0 \rightarrow i$ **do**

$\quad f[j] \leftarrow f[j] + BTH[i] * BTT[j]$

end for

return f

2. Đa thức nội suy Newton

- $Newton\_interpolation\_Forward$:

Input: BTH, BTT, f, x, y, n

Output: f, BTH, BTT

for $i = 0 \rightarrow n$ **do**

$BTH \leftarrow BuildBTH(BTH, x, y, i)$

$BTT \leftarrow BuildBTT(BTT, x, i - 1)$

$f \leftarrow fx(BTH, BTT, i, f)$

end for

return f, BTH, BTT

- $Newton\_interpolation\_Backward$:

Input: BTH, BTT, f, x, y, n

Output: f, BTH, BTT

x.reverse()

y.reverse()

$BTH \leftarrow y[0]$

$f[0] \leftarrow y[0]$

return $Newton\_interpolation\_Forward$

2. Đa thức nội suy Newton

- $fx0$:

Input: $f, n, x0$

Output: $value$

$value \leftarrow f[n]$

for $i = n - 1 \rightarrow 0$ **do**

    $value \leftarrow value * x0 + f[i]$

end for

return $value$

- add_Newton:

Input: BTH, BTT, f, x, y, n

Output: f, BTH, BTT

$BTH \leftarrow BuildBTH(BTH, x, y, n)$

$BTT \leftarrow BuildBTT(BTT, x, n - 1)$

Tạo thêm 1 cột cho f

$f \leftarrow fx(BTH, BTT, n, f)$

return f, BTH, BTT

3. Đa thức nội suy Newton với mốc cách đều

- Ý tưởng phương pháp:

Trường hợp các mốc cách đều: $x_k = x_0 + kh = x_n - (n-k)h$

$h = x_{i+1} - x_{\mathrm{i}} \forall i = \overline{0, n-2}$. h được gọi là bước lưới

Xuất phát từ x_0 đến x_k là bước tiến k bước

Xuất phát từ x_k đến x_0 là bước lùi k bước

Khi đó trong Bảng tỷ hiệu, trên mỗi 1 cột mẫu số sẽ giống nhau. Nhưng chúng ta chỉ cần dùng giá trị đầu hoặc cuối bảng để tìm đa thức.

Do vậy, để giảm số lượng tính toán, phép chia sẽ được thực hiện sau cùng.

- Sai phân:

Sai phân tiến cấp 1 xuất phát từ x_k: $\Delta y_k = y_{k+1} - y_k$

Sai phân lùi cấp 1 xuất phát từ x_k: $\nabla y_k = y_k - y_{k-1} = \Delta y_{k-1}$

Sai phân tiến cấp 2 xuất phát từ x_k:

$$\Delta^2 y_k = \Delta(\Delta y_k) = \Delta y_{k+1} - \Delta y_k = (y_{k+2} - y_{k+1}) - (y_{k+1} - y_k) = y_{k+2} - 2y_{k+1} + y_k$$

Sai phân lùi cấp 2 xuất phát từ x_k:

$$\nabla^2 y_k = \nabla(\nabla y_k) = \nabla y_k - \nabla y_{k-1} = y_k - 2y_{k-1} + y_{k-2} = \Delta^2 y_{k-2}$$

- Sai phân:

Sai phân tiến cấp s xuất phát từ x_k:

$$\Delta^s y_k = \Delta(\Delta^{s-1} y_k)$$

Sai phân lùi cấp s xuất phát từ x_k:

$$\nabla^s y_k = \nabla(\nabla^{s-1} y_k) = \Delta^s y_{k-s}$$

- Bảng sai phân tiến:

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\dots$	$\Delta^n y$
x_0	y_0				$\dots$	
x_1	y_1	Δy_0			$\dots$	
x_2	y_2	Δy_1	$\Delta^2 y_0$		$\dots$	
x_3	y_3	Δy_2	$\Delta^2 y_1$	$\Delta^3 y_0$	$\dots$	
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
x_n	y_n	Δy_{n-1}	$\Delta^2 y_{n-2}$	$\Delta^3 y_{n-3}$	$\dots$	$\Delta^n y_0$

3. Đa thức nội suy Newton với mốc cách đều

- Bảng sai phân lùi:

x	y	∇y	$\nabla^2 y$	$\nabla^3 y$	$\dots$	$\nabla^{n-1} y$	$\nabla^n y$
x_0	y_0	∇y_1	$\nabla^2 y_2$	$\nabla^3 y_3$	$\dots$	$\nabla^{n-1} y_{n-1}$	$\nabla^n y_n$
x_1	y_1	∇y_2	$\nabla^2 y_3$	$\nabla^3 y_4$	$\dots$	$\nabla^{n-1} y_n$	
x_2	y_2	∇y_3	$\nabla^2 y_4$	$\dots$	$\dots$		
x_3	y_3	∇y_4	$\dots$	$\nabla^3 y_n$			
$\dots$	$\dots$	$\dots$	$\nabla^2 y_n$				
x_{n-1}	y_{n-1}	∇y_n					
x_n	y_n						

- Tính chất của sai phân:

Tính chất 1 (liên hệ giữa sai phân và tỷ sai phân):

$$f[x_0,\dots,x_k] = \frac{\Delta^k y_0}{h^k k!} = \frac{\nabla^k y_k}{h^k k!}$$

Chứng minh:

$$f[x_0,x_1] = \frac{f(x_1)-f(x_0)}{x_1-x_0} = \frac{\Delta y_0}{h}$$

$$f[x_0,x_1,x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0} = \frac{\Delta y_1-\Delta y_0}{2h^2} = \frac{1}{2!\,h^2}\Delta^2 y_0$$

….

$$\Rightarrow f[x_0,x_1,\dots,x_k] = \frac{1}{k!\,h^2}\Delta^k y_0$$

3. Đa thức nội suy Newton với mốc cách đều

- Tính chất của sai phân:

Tính chất 2: $\mathit{\Delta}$ là toán tử tuyến tính:

$$\mathit{\Delta}(\alpha y + \beta z) = \alpha \mathit{\Delta} y + \beta \mathit{\Delta} z$$

$$\Delta C = 0 \; C - const$$

$$\mathit{\Delta}^n(x^n) = n!\,h^n$$

$$\mathit{\Delta}^m(x^n) = 0 \; khi \; m > n$$

$$\mathit{\Delta}^m(\mathit{\Delta}^k f) = \mathit{\Delta}^{k+m} f$$

$$\mathit{\Delta}^0 f = f$$

- Tính chất của sai phân:

Tính chất 3: Giá trị của hàm $f(x)$ được biểu diễn qua sai phân các cấp của nó

$$y_m = f(x_m) = f(x + mh) = \sum_{k=0}^{m} C_m^k \Delta^k f(x) \qquad C_m^k = \frac{m!}{k!\,(m-k)!}$$

Chứng minh:

$$\Delta f(x) = f(x+h) - f(x) \Rightarrow f(x+h) = (1+\Delta) f(x)$$

$$f(x+2h) = f(x+h+h) = (1+\Delta) f(x+h) = (1+\Delta)^2 f(x)$$

....

$f(x+mh) = (1+\Delta)^m f(x) = \sum_{k=0}^{m} C_m^k \Delta^k f(x)$ (Theo khai triển Newton)

- Tính chất của sai phân:

Tính chất 4: Sai phân cấp n của $f(x)$ được biểu diễn qua các giá trị liên tiếp của nó

$$\Delta^n f(x) = \sum_{i=0}^{n} (-1)^i C_n^i f(x + (n-i)h)$$

Chứng minh:

$$\Delta^n f(x) = [(1+\Delta) - 1]^n f(x) = \sum_{i=0}^{n} (-1)^i C_n^i (1+\Delta)^{n-i} f(x)$$

$$= \sum_{i=0}^{n} (-1)^i C_n^i f(x + (n-i)h)$$

- Tính chất của sai phân:

Tính chất 5: Giá trị $f(x)$ có đạo hàm liên tục đến cấp n trên đoạn $[x, x+nh]$ thì ta có

$$\Delta^n f(x) = h^n f^{(n)}(x+\theta nh) \qquad 0<\theta<1$$

Chứng minh:

- Khi $n=1$:

$$\Delta f(x) = f(x+h) - f(x) = hf'(x+h\theta') \text{ (công thức số gia giới nội)}$$

Giả sử đúng với $n=k$:

$$\Delta^k f(x) = h^k f^{(k)}(x+\theta' kh)$$

$$\Delta^n f(x) = h^n f^{(n)}(x + \theta n h) \qquad 0 < \theta < 1$$

Ta cần chứng minh đúng với $n = k + 1$:

$$\Delta^{k+1} f(x) = \Delta(\Delta^k f(x)) = \Delta\left(h^k f^{(k)}(x + \theta' k h)\right)$$

$$= h^k\left[f^{(k)}(x + kh\theta' + h) - f^{(k)}(x + kh\theta')\right]$$

Áp dụng công thức số gia giới nội: $\Delta^{k+1} f(x) = h^k h f^{(k+1)}(x + \theta' k h + \theta'' h)$

Đặt:

$$\frac{\theta' k + \theta''}{k+1} = \theta$$

$$\Rightarrow \Delta^{k+1} f(x) = h^{k+1} f^{(k+1)}(x + \theta(k+1)h)$$

3. Đa thức nội suy Newton với mốc cách đều

- Đa thức nội suy Newton tiến với mốc cách đều

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

Đặt: $x = x_0 + th$

$f[x_0, \ldots, x_k] = \frac{\Delta^k y_0}{h^k k!}$

$x - x_0 = th$

$x - x_1 = x - x_0 + x_0 - x_1 = th - h = h(t - 1)$

$x - x_{k-1} = x - x_0 + x_0 - x_{k-1} = th - (k - 1)h = h(t - k + 1)$

$(x - x_0)(x - x_1)\ldots(x - x_{k-1}) = h^k(t - 1)(t - 2)\ldots.(t - k + 1)$

3. Đa thức nội suy Newton với mốc cách đều

- Đa thức nội suy Newton tiến với mốc cách đều

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

Đặt: $x = x_0 + th$

$f[x_0, \ldots, x_k] = \frac{\Delta^k y_0}{h^k k!}$

$\Rightarrow f[x_0, \ldots, x_k](x - x_0)(x - x_1)\ldots(x - x_{k-1}) = \frac{\Delta^k y_0}{k!}(t-1)(t-2)\ldots.(t-k+1)$

$\Rightarrow P_n(t) = y_0 + \frac{\Delta y_0}{1!}t + \frac{\Delta^2 y_0}{2!}t(t-1) + \ldots + \frac{\Delta^n y_0}{n!}t(t-1)\ldots(t-n+1)$

- Đa thức nội suy Newton lùi với mốc cách đều

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

Đặt: $x = x_n + th$

$$f[x_k, \ldots, x_0] = \frac{\nabla^k y_k}{h^k k!}$$

$$x - x_k = x - x_n + x_n - x_k = th + (n-k)h = h(t + n - k)$$

$$x - x_{k-1} = x - x_n + x_n - x_{k-1} = th + (n-k+1)h = h(t + n - k + 1)$$

$$x - x_1 = x - x_n + x_n - x_1 = th + (n-1)h = h(t + n - 1)$$

$$(x - x_k)(x - x_{k-1})\ldots(x - x_1) = h^k(t + n - k)(t + n - k + 1)\ldots(t + n - 1)$$

3. Đa thức nội suy Newton với mốc cách đều

- Đa thức nội suy Newton lùi với mốc cách đều

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

Đặt: $x = x_n + th$

$$f[x_k, \ldots, x_0] = \frac{\nabla^k y_k}{h^k k!}$$

$$\Rightarrow f[x_k, \ldots, x_0](x - x_k)\ldots(x - x_1) = \frac{\nabla^k y_k}{k!}(t + n - k)(t + n - k + 1)\ldots(t + n - 1)$$

$$\Rightarrow P_n(t) = y_n + \frac{\nabla y_n}{1!}t + \frac{\nabla^2 y_n}{2!}t(t+1) + \ldots + \frac{\nabla^n y_n}{n!}t(t+1)\ldots(t+n-1)$$

3. Đa thức nội suy Newton với mốc cách đều

- Nếu thêm mốc nội suy (x_{n+1}, y_{n+1}) đa thức $P_{n+1}(x)$ được tính như sau:

Đối với Newton tiến: $P_{n+1}(t) = P_n(t) + \dfrac{\Delta^{n+1} y_0}{(n+1)!} t(t-1)(t-2)\ldots(t-n)$

Đối với Newton lùi: $P_{n+1}(t) = P_n(t) + \dfrac{\nabla^{n+1} y_{n+1}}{(n+1)!} t(t+1)(t+2)\ldots(t+n)$

- Khối lượng phép toán: $3n^2 + 3n - 2$

3. Đa thức nội suy Newton với mốc cách đều

- Sai số:

Công thức đánh giá sai số:

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

Đa thức nội suy Newton tiến. Từ $x = x_0 + th$ ta được:

$$\omega_{n+1}(x) = \prod_{k=0}^{n}(x - x_k) = \prod_{k=0}^{n}(t - k)$$

Và do $f^{(n+1)}(x) = \lim_{h \to 0} \frac{\Delta^{n+1} y_0}{h^{n+1}}$ nên nếu xem: $f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1} y_0}{h^{n+1}}$

Thì

$$R_n(x) \approx \frac{\Delta^{n+1} y_0}{(n+1)!}\prod_{k=0}^{n}(t - k)$$

2. Đa thức nội suy Newton với mốc cách đều

- Thuật toán: Nội suy đa thức Newton tiến mốc cách đều

Các gói sử dụng:

1. $main$: thuật toán chính
2. $Input$: nhập x, y từ file và kiểm tra điều kiện cách đều
3. $pickPoint$: chọn ra num mốc gần $x0$ nhất
4. $BuildBSP$: kết nạp mốc $x[i]$ vào bảng sai phân tiến
5. $BuildBTT$: kết nạp mốc $x[i]$ vào bảng tính tích theo lược đồ Hoocner
6. fx: kết nạp mốc $x[i]$ vào đa thức
7. $Newton\_interpolation\_Forward$: xây dựng đa thức nội suy Newton tiến cách đều
8. $fx0$: ước tính giá trị tại $x0$ theo lược đồ Hoocner
9. add_Newton: thêm mốc nội suy
10. $plot$: vẽ đồ thị đa thức f

- $main$:

Input: file các mốc nội suy, điểm cần ước lượng giá trị: $x0$,

số lượng thêm mốc: k, số lượng mốc nội suy tính: num

Output: hệ số đa thức: f, giá trị ước lượng tại $x0$: $value$, đồ thị đa thức f

- $main$:

$x, y, h \leftarrow Input()$

$h \leftarrow x[1] - x[0]$

x1, y1 ← pickPoint(x0, num, x, y)

In các mốc nội suy $x1$

$n \leftarrow$ len($x1$) – 1

Khởi tạo các ma trận không:

$BSP(n+1), BTT(n+1), f(n+1)$

$BSP[0], BTT[0], f[0] \leftarrow y1[0], 1, y1[0]$

$f, BSP, , BTT \leftarrow Newton\_interpolation\_Forward(BSP, BTT, f, x1, y1, n)$

$value \leftarrow fx0(f, x0, x1)$

In hệ số đa thức f và giá trị $value$

$plot(x1, y1, f)$

```
if k <= 0 then: Dừng chương trình
end if
for i = 1 → k do
    Nhập mốc và giá trị nội suy: {xt, yt}
    if xt not in x1 and abs(xt – x1[n] – h) < 1e-6 then
        n ← n + 1
        append {xt, yt} vào {x1, y1}
        f, BSP, BTT ← add_Newton(BSP, BTT, f, x1, y1, n)
    end if
end for
In hệ số đa thức f và giá trị value
plot(x1, y1, f)
```

2. Đa thức nội suy Newton với mốc cách đều

- $pickPoint$:

Input: $x0, num, x, y$

Output: $x1, y1$

if $num > \text{len}(x)$ then

  Báo lỗi và dừng chương trình

end if

$i \leftarrow \text{int}((x0 - x[0])/(x[1] - x[0]))$

$left \leftarrow \min(\text{len}(x) - num, \max(0, i + 1 - \text{int}(num/2)))$

$right \leftarrow left + num - 1$

Khởi tạo list $x1, y1$

$x1 \leftarrow x[left : right + 1]$

$y1 \leftarrow y[left: right + 1]$

return $x1, y1$

2. Đa thức nội suy Newton với mốc cách đều

- $BuildBSP$:

Input: BSP, x, y, i

Output: SP_row_i

Khởi tạo ma trận không $SP\_row\_i(\text{len}(x))$

$SP\_row\_i[0] \leftarrow y[i]$

for $j = 1 \rightarrow i$ **do**

  $SP\_row\_i[j] \leftarrow SP\_row\_i[j-1] - BSP[j-1]$

end for

return SP_row_i

- $BuildBTT$:

Input: BTT, x, i

Output: TT_row_i

Khởi tạo ma trận không $TT\_row\_i(\text{len}(x))$

$TT\_row\_i[i] \leftarrow 1$

for $j = i \ \rightarrow 1$ **do**

$TT\_row\_i[j] \leftarrow BTT[j - 1] - i * BTT[j]$

end for

if $i \ != \ 0$ **then**

$TT\_row\_i[0] \leftarrow -i * BTT[0]$

end if

return TT_row_i

2. Đa thức nội suy Newton với mốc cách đều

- fx:

Input: BSP, BTT, i, f

Output: f

for $j = 1 \to i$ **do**

$f[j] \leftarrow f[j] + BSP[i] * BTT[j-1]/i!$

end for

return f

- $Newton\_interpolation\_Forward$:

Input: $BSP, BTT, f, x1, y1, n$

Output: f, BSP, BTT

for $i = 1 \rightarrow n$ **do**

$BSP \leftarrow BuildBSP(BSP, x1, y1, i)$

$BTT \leftarrow BuildBTT(BTT, x1, i - 1)$

$f \leftarrow fx(BSP, BTT, i, f)$

end for

return f, BSP, BTT

2. Đa thức nội suy Newton với mốc cách đều

- $fx0$:

Input: $f, x0, x1$

Output: $value$

$t \leftarrow (x0 - x1[0])/(x1[1] - x1[0])$

$value \leftarrow f[\text{len}(x1) - 1]$

for $i = \text{len}(x1) - 2 \rightarrow 0$ **do**

  $value \leftarrow value * t + f[i]$

end for

return $value$

- add_Newton:

Input: $BSP, BTT, f, x1, y1, n$

Output: f, BSP, BTT

$BSP \leftarrow BuildBSP(BSP, x1, y1, n)$

$BTT \leftarrow BuildBTT(BTT, x1, n - 1)$

Tạo thêm 1 cột cho f

$f \leftarrow fx(BSP, BTT, n, f)$

return f, BSP, BTT

2. Đa thức nội suy Newton với mốc cách đều

- Thuật toán: Nội suy đa thức Newton lùi mốc cách đều

Các gói sử dụng:

1. $main$: thuật toán chính
2. $Input$: nhập x, y từ file và kiểm tra điều kiện cách đều
3. $pickPoint$: chọn ra num mốc gần $x0$ nhất
4. $BuildBSP$: kết nạp mốc $x[i]$ vào bảng sai phân lùi
5. $BuildBTT$: kết nạp mốc $x[i]$ vào bảng tính tích theo lược đồ Hoocner
6. fx: kết nạp mốc $x[i]$ vào đa thức
7. $Newton\_interpolation\_Backward$: xây dựng đa thức nội suy Newton tiến cách đều
8. $fx0$: ước tính giá trị tại $x0$ theo lược đồ Hoocner
9. add_Newton: thêm mốc nội suy
10. $plot$: vẽ đồ thị đa thức f

*Note: Các gói 2, 3, 6, 10 giống với Nội suy đa thức Newton tiến cách đều

2. Đa thức nội suy Newton với mốc cách đều

- $main$:

Input: file các mốc nội suy, điểm cần ước lượng giá trị: $x0$,

số lượng thêm mốc: k, số lượng mốc nội suy tính: num

Output: hệ số đa thức: f, giá trị ước lượng tại $x0$: $value$, đồ thị đa thức f

2. Đa thức nội suy Newton với mốc cách đều

- $main$:

$x, y, h \ \leftarrow Input()$

$h \ \leftarrow x[1] - x[0]$

x1, y1 ← pickPoint(x0, num, x, y)

In các mốc nội suy $x1$

$n \leftarrow \text{len}(x) - 1$

Khởi tạo các ma trận không:

$BSP(n+1), BTT(n+1), f(n+1)$

$BSP[0], BTT[0], f[0] \leftarrow y1[n], 1, y1[n]$

$f, BSP, BTT \leftarrow Newton\_interpolation\_Backward(BSP, BTT, f, x1, y1, n)$

$value \ \leftarrow fx0(f, x0, x1)$

In hệ số đa thức f và giá trị $value$

$plot(x1, y1, f)$

if $k <= 0$ **then**: Dừng chương trình

end if

for $i = 1 \rightarrow k$ **do**

Nhập mốc và giá trị nội suy: $\{xt, yt\}$

if xt **not in** $x1$ **and** abs$(x1[0] - xt - h) < 1$e-6 **then**

$n \leftarrow n + 1$

insert $\{xt, yt\}$ vào vị trí đầu $\{x1, y1\}$

$f, BSP, BTT \leftarrow add\_Newton(BSP, BTT, f, x1, y1, n)$

end if

end for

In hệ số đa thức f và giá trị $value$

$plot(x1, y1, f)$

- $BuildBSP$:

Input: BSP, x, y, i

Output: SP_row_i

Khởi tạo ma trận không $SP\_row\_i(\text{len}(x))$

$SP\_row\_i[0] \leftarrow y[i]$

for $j = 1 \rightarrow \text{len}(x) - i - 1$ **do**

  $SP\_row\_i[j] \leftarrow BSP[j - 1] - SP\_row\_i[j - 1]$

end for

return BSP

- $BuildBTT$:

Input: BTT, x, i

Output: TT_row_i

Khởi tạo ma trận không $TT\_row\_i(\text{len}(x))$

$TT\_row\_i[-i] \leftarrow 1$

for j = -i $\rightarrow 1$ **do**

$TT\_row\_i[j] \leftarrow BTT[j - 1] - i * BTT[j]$

end for

if i ! = 0 **then**

$TT\_row\_i[0] = -i * BTT[0]$

end if

return TT_row_i

- $Newton\_interpolation\_Backward$:

Input: $BSP, BTT, f, x1, y1, n$

Output: f, BSP, BTT

for $i = 1 \rightarrow n$ **do**

$BSP \leftarrow BuildBSP(BSP, x1, y1, n - i)$

$BTT \leftarrow BuildBTT(BTT, x1, 1 - i)$

$f \leftarrow fx(BSP, BTT, i, f)$

end for

return f, BSP, BTT

2. Đa thức nội suy Newton với mốc cách đều

- $fx0$:

Input: $f, x0, x1$

Output: $value$

$t \leftarrow (x0 - x1[\text{len}(x1) - 1])/(x1[1] - x1[0])$

$value \leftarrow f[\text{len}(x1) - 1]$

for $i = \text{len}(x1) - 2 \rightarrow 0$ **do**

  $value \leftarrow value * t + f[i]$

end for

return $value$

- add_Newton:

Input: $BSP, BTT, f, x1, y1, n$

Output: f, BSP, BTT

$BSP \leftarrow BuildBSP(BSP, x1, y1, 0)$

$BTT \leftarrow BuildBTT(BTT, x1, 1 - n)$

Tạo thêm 1 cột cho f

$f \leftarrow fx(BSP, BTT, n, f)$

return f, BSP, BTT

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !