METODE NUMERICE: Laborator #9
Metoda Neville. Metode de interpolare cu functii spline cubice. Curebe Bezier. Algoritmul de Casteljau

Titulari curs: *Florin Pop, George-Pantelimon Popescu*

Responsabil Laborator: **Bogdan Marchiș**

Obiective Laborator

In urma parcurgerii acestui laborator studentul va fi capabil sa:

- calculeze valoarea unui punct intr-o functie prin metoda Neville
- aproximeze traiectoria unei functii prin functii spline cubice
- traseze o curba Bezier cu algoritmul deCasteljau

Noțiuni teoretice

Metoda Neville

Teorema: Avand n+1 puncte $P_0, ..., P_n$ si parametri $t_0, ...t_n$, exista un polinom $P_{0...n}(t)$ de grad n care interpoleaza pnctele in parametrii specificati

$$P_{0...n}(t_k) = P_k, k = 0...n \tag{1}$$

$$\begin{array}{lllll}
p_{0,0}(x) = y_0 & & & & \\
 & p_{0,1}(x) & & & \\
p_{1,1}(x) = y_1 & & p_{0,2}(x) & & \\
 & p_{1,2}(x) & & p_{0,3}(x) & \\
p_{2,2}(x) = y_2 & & p_{1,3}(x) & & \boxed{p_{0,4}(x)} \\
 & p_{2,3}(x) & & p_{1,4}(x) & \\
p_{3,3}(x) = y_3 & & p_{2,4}(x) & & \\
 & p_{3,4}(x) & & & \\
p_{4,4}(x) = y_4 & & & &
\end{array}$$

Figure 1: Fiecare element depinde de doua elemente precedente

Metoda Neville este o metodă de intrepolare care în primă fază asociază valorile $y_k, k = 1, ..., n$ lui $P_k(x)$: $P_k(x) = y_k$. A doua interaţie i-a perechi P_i si P_{i+1} şi le combină in valori: $P_{12}, P_{23}, ...$. Procedura se repete generând o piramida cu rezultate pana cand rezultatul final este obţinut:

$$P_{i(i+1)...(i+m)}(x) = \frac{(x-x_{i+m})*P_{i(i+1)...(i+m-1)}(x)-(z-x_i)*P_{(i+1)(i+2)...(i+m)}(x)}{x_i-x_{i+m}} \tag{2}$$

Exemplu

Pornim de la functia:

$$f(x) = \frac{1}{x} \tag{3}$$

Si vrem sa evaluam functia in punctul f(3). In prima faza evaluam functia in trei puncte

$$\begin{array}{ccc} i & x_i & f(x_i) \\ 0 & 2 & 0.5 \\ 1 & 2.5 & 0.4 \\ 2 & 4 & 0.25 \end{array} \tag{4}$$

Putem sa facem trei aproximari de ordin zero:

$$f(3) \simeq P_0(3) = f(x_0) = 0.5 \tag{5}$$

$$f(3) \simeq P_1(3) = f(x_1) = 0.4 \tag{6}$$

$$f(3) \simeq P_2(3) = f(x_2) = 0.25 \tag{7}$$

Din ecuatiile de mai sus calculam $P_{0,1}$ si $P_{1,2}$ folosind formula lui Neville:

$$f(3) \simeq P_{0,1}(3) = \frac{(3-x_1)P_0(3)-(3-x_0)P_1(3)}{x_0-x_1} = \frac{(3-2.5)0.5-(3-2)0.4}{2-2.5} = 0.3 \tag{8}$$

$$f(3) \simeq P_{1,2}(3) = \frac{(3-x_2)P_1(3)-(3-x_1)P_2(3)}{x_1-x_2} = \frac{(3-4)0.4-(3-2.5)0.25}{2.5-4} = 0.35 \tag{9}$$

Acum putem sa completam tabelul cu valorile de pe a treia linie:

$$\begin{array}{cccc} i & x_i & f(x_i) & P_{i,i-1} \\ 0 & 2 & 0.5 & \\ 1 & 2.5 & 0.4 & 0.3 \\ 2 & 4 & 0.25 & 0.35 \end{array} \tag{10}$$

Si acum putem sa calculam $P_{0,1,2}$ folosind $P_{0,1}$ si $P_{1,2}$:

$$f(3) \simeq P_{0,1,2}(3) = \frac{(3-x_2)P_{0,1}(3)-(3-x_0)P_{1,2}(3)}{x_0-x_2} = \frac{(3-4)0.3-(3-2)0.35}{4-2} = 0.325 \tag{11}$$

Si tabelul final este:

$$\begin{array}{ccccc} i & x_i & f(x_i) & P_{i,i-1} & P_{i,i-1,i-2} \\ 0 & 2 & 0.5 & & \\ 1 & 2.5 & 0.4 & 0.3 & \\ 2 & 4 & 0.25 & 0.35 & 0.325 \end{array} \tag{12}$$

Funcţii spline cubice de interpolare

Scopul spline-ului cubic de interpolare este de a gasi o formula care are prima si a doua derivata continua atat in interval cat si la capetele acestuia. Astfel vom obtine o functie de interpolare mai neteda. In general daca functia pe care vrem sa o aproximam este neteda, spline-urile cubice vor da un rezultat mai bun decat interpolarea liniara.

Un spline cubic de interpolare este definit ca:

$$S_i : [x_i, x_{i+1}] \rightarrow \Re, S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{13}$$

In general in calculul spline-urilor se foloseste baza Bernstein, care are urmatorul suport:

$$(1-t)^3, 3t(1-t)^2, 3t^2(1-t), t^3 \tag{14}$$

Conditii de interpolare Hermite (clasa C^1):

$$S_i(x_i) = f(x_i), i = 0 : n-1 \tag{15}$$
$$S_i^{'}(x_i) = f^{'}(x_i) \tag{16}$$
$$S_{n-1}(x_n) = f(x_n) \tag{17}$$
$$S_{n-1}{}^{'}(x_n) = f^{'}(x_n) \tag{18}$$

Conditii de continuitate si derivabilitate in nodurile interne:

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}), i = 0 : n-2 \tag{19}$$
$$S_i^{'}(x_{i+1}) = S_{i+1}{}^{'}(x_{i+1}) \tag{20}$$

Conditii de interpolare Lagrange (clasa C^2):

$$S_i(x_i) = f(x_i), i = 0 : n-1 \tag{21}$$
$$S_{n-1}(x_n) = f(x_n) \tag{22}$$

Conditii de continuitate, derivabilitate si curbura in nodurile interne:

$$S_i(x_{i+1}) = S_{i+1}(x_{i+1}), i = 0 : n-2 \tag{23}$$
$$S_i^{'}(x_{i+1}) = S_{i+1}{}^{'}(x_{i+1}) \tag{24}$$
$$S_i^{''}(x_{i+1}) = S_{i+1}{}^{''}(x_{i+1}) \tag{25}$$

- spline-uri naturale:

$$S_0{}^{''}(x_0) = S_{n-1}{}^{''}(x_n) \tag{26}$$

- spline-uri tensionate:

$$S_0{}^{'}(x_0) = f^{'}(x_0) \tag{27}$$
$$S_{n-1}{}^{'}(x_n) = f^{'}(x_n) \tag{28}$$

Curbe Bezier

Spre deosebire de spline-uri, curaba Bezier nu trece prin fiecare punct de control. Fiecare punct trage curba inspre el, ceea ce inseamna ca fiecare punct de control afecteaza curba in mod particular si daca modificam pozitia unui punct de control si curba va fi afectata.

Avand un set de $n+1$ puncte de control $P_0, P_1, ..., P_n$ curba Bezier corespunzatoare este data de ecuatia:

$$B(t) = \sum_{i=0}^{n} P_i B_i^n(t), t \in [0,1] \tag{29}$$

unde functiile de imbinare sunt polinoame Bernstein.

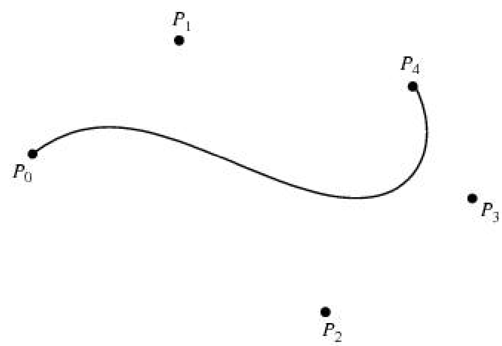

Figure 2: A simple caption

Proprietati:

- Curbele Bezier incep in P_0 si se termina in P_n:
  $B(0) = P_0$
  $B(1) = P_n$
- Curba Bezier este tangenta segmentelor P_0P_1 si $P_{n-1}P_n$:
  $B^{'}(0) = n(B_1 - B_0)$
  $B^{'}(1) = n(B_n - B_{n-1})$
- O curba Bezier este continuta complet in infasurarea complexa a punctelor de control

Curba Bezier cubica are forma:

$$Q(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, t \in [0,1] \tag{30}$$

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{31}$$

Spline-urile cubice Bezier folosesc in locul derivatelor doua puncte suplimentare de control. Punctele suplimentare de control determina derivatele, si asigura un control mai eficient decat cel prin derivate. Derivatele din capete devin:

$P_1^{'} = 3(P_1 - P_0)$

$P_2^{'} = 3(P_3 - P_2)$

ALgoritmul de Casteljau

Metoda evidenta pentru calularea unui punct $B(t)$ intr-o curaba Bezier este prin folosirea ecuatiei: $B(t) = \sum_{i=0}^{n} P_i B_i^n(t), t \in [0,1]$. Acesta metoda este foarte ineficienta intrucat ridica numere mici la puteri mari, ganerand erori mari.

O modalitate mult mai buna este folosirea algiritmului de Casteljau. Acesta este putin mai lent dar este numerc stabil si cu ajutorul sau putem obtine detalii despre curba Bezier:

- Calculul derivatelor (cu derivata curbei obtinem vectorul tangent intr-un punct)
- Subdivizarea curbei. Uneori este necesar sa luam o curba Bezier si sa o impartim in doua curbe care impreuna sunt dau initiala.

Algoritmul de Casteljau poate fi considerat o interpolare liniara repetata.

Algoritmul utilizeaza relatia de recurenta:

$$P_i^{(0)} = P_i, i = 0 : n \tag{32}$$

$$P_i^{(j)} = P_i^{(j-1)}(1 - t_0) + P_{i+1}^{(j-1)} t_0, j = 1 : n, i = 0 : n - j \tag{33}$$

cu $B(t_0) = P_0^{(n)}$

Fie $P_i^{(0)}, i = 0 : n$ poligonul punctelor de control $P_i^{(0)}$ si $P_{i+1}^{(0)}$ doua puncte succesive si $P_i^{(1)}$ un punct care imparte segmentul $P_i^{(0)} P_{i+1}^{(0)}$ in raportul $t/(1 - t)$:

$$P_i^1 = P_i^0 + t(P_{i+1}^0 - P_i^0) = (1 - t)P_i^0 + tP_{i+1}^0 \tag{34}$$

- Se foremaza in acest fel poligonul $P_i^1, i = 0 : n - 1$
- Se aplică algoritmul noului poligon acelaşi algoritm obţinându-se poligonul $P_i^2, i = 0 : n-2$. Repetand procesul de n ori se obtine un singur punct P_0^n. Acest punct este punctul cerut.

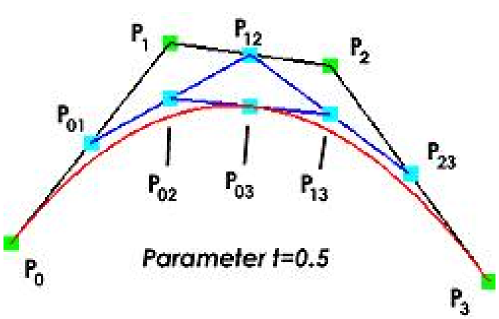

Figure 3: A simple caption

Problema 1

Scrieti o functie OCTAVE care sa foloseasca metoda Neville de interpolare pentru a afla valoarea functiei f in punctul y. Rulati programul pentru exemplul din laborator.

$functione\ rez = Neville(x, f, y)$

Problema 2

a) Scrieti o functie ce calculeaza valoarea unei functii intr-un punct a ca rezultat al interpolarii obtinute folosind splineuri de clasa C1, folosind suportul de interpolare x, y, precum si derivatele dx.

$function\ s = splineC1(a, x, y, dx)$

b)Scrieti o functie ce calculeaza valoarea unei functii intr-un punct a ca rezultat al interpolarii obtinute folosind splineuri de clasa C2 naturale, folosind suportul de interpolare x, y.

$function\ s = splineC2(a, x, y)$

Problema 3

Scrieti o functie care sa deseneze o curba Bezier cu ajutorul algoritumului de Casteljau

$function\ s = deCasteljau(x, y)$