Laborator 4 – Interpolare numerica. Polinoame ortogonale

Resposabil:

Ana Ion (ana.ion4@gmail.com)

Obiective:

In urma parcurgerii acestui laborator studentul va fi capabil sa inteleaga si sa utilizeze diferite metode de interpolare, precum si sa utilizeze cea mai buna varianta in functie de situatia specifica in care interpolarea este necesara.

Suport teoretic:

Exista situatii frecvente in care o functie reala $f : [a,b] \to R$ este cunoscuta doar in anumite puncte $x_0, x_1, .., x_n$. In acest caz, deducerea exacta a formei functiei nu este posibila, deci devine necesara utilizarea unei aproximari a acesteia. Astfel, functia $f : [a,b] \to R$, cunoscuta doar in $x_0, x_1, ..., x_n$ prin valorile $f(x_0), f(x_1), ..., f(x_n)$ este aproximata in afara suportului printr-un polinom de interpolare:

$$P_n(x) = a_0u_0(x) + a_1u_1(x) + ... + a_nu_n(x),$$

unde $u_0(x), u_1(x), \dots, u_n(x)$ formeaza baza interpolarii.

Determinarea coeficientilor $a_0, a_1, \dots, a_n$ presupune stabilirea unor conditii de interpolare:

$$Pn(xi) = f(xi), \qquad i = 0:n,$$

conditii ce conduc la sistemul de ecuatii:

$$\sum_{k=0}^{n} a_k u_k(x_i) = f(x_i), i = 0:n$$

Interpolare polinomiala

In cazul interpolarii polinomiale, baza folosita este $u_k(x) = x^k$, k=0:n, unde polinomul de interpolare se calculeaza folosind formula Lagrange:

$$P_n(x) = \sum_{k=0}^{n} f(x_k) \frac{\prod_{i=0, i\neq k}^{n} (x - x_i)}{\prod_{i=0, i\neq k}^{n} (x_k - x_i)} = \pi(x) \sum_{k=0}^{n} \frac{f(x_k)}{(x - x_k)\pi'(x_k)}$$

unde:

$$\pi(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$$

In raport cu suportul $x_0, \ldots, x_n$, se definesc diferentele divizate ale unei functii *f* astfel:

$$F_0[x_i] = f(x_i), i = 0:n$$

$$F_k[x_0, x_1, \ldots, x_k] = \frac{F_{k-1}[x_0, \ldots, x_{k-1}] - F_{k-1}[x_1, \ldots x_k]}{x_0 - x_k}, k = 1:n$$

$$F_k[x_0, x_1, \ldots, x_k] = \sum_{j=0}^{k} \frac{f(x_j)}{\pi'(x_j)} = \sum_{j=0}^{k} \frac{f(x_j)}{\prod_{i=0, i \neq j} (x_j - x_i)}$$

Polinomul de interpolare se poate exprima folosind diferentele divizate(*formula lui Newton*) astfel:

$P_n(x) = f(x_0)+(x-x_0)F_1[x_0,x_1]+(x-x_0)(x-x_1)F2[x_0,x_1,x_2]+ \ldots+(x-x_0)\ldots(x-x_{n-1})F_n[x_0,x_1,\ldots,x_n]$.

Pentru aceasta metoda eroare interpolarii se calculeaza utilizand urmatoarea formula:

$$E_n(X) = \pi(x)F_{n+1}[x, x_0, \ldots, x_n] \le |\pi(x)| \frac{M_{n+1}(f)}{(n+1)!}$$

<u>Polinoame ortogonale</u>

Interpolare cu polinoame Cebasev

Baza interpolarii:

$$\frac{1}{\sqrt{2}}, T_1(x), T_2(x), \ldots, T_n(x)$$

Polinomul generalizat de interpolare:

$$P_n(x) = \frac{a_0}{\sqrt{2}} + a_1 T_1(x) + \ldots + a_n T_n(x)$$

Suportul interpolarii este reprezentat de radcinile polinomului Cebasev de grad n+1:

$$T_{n+1}(x_k) = 0, x_k = \cos\frac{2k+1}{2n+2}\pi, k = 0:n$$

Utilizand sistemul de ecuatii precizat la inceputul documentului si proprietatea de ortogonalitate a polinomului se obtine solutia:

$$a_0 = \frac{\sqrt{2}}{n+1}\sum_{k=0}^{n} f(x_k) = \frac{\sqrt{2}}{n+1}\sum_{k=0}^{n} f(\cos\frac{2k+1}{2n+2}\pi)$$

$$a_j = \frac{2}{n+1}\sum_{k=0}^{n} f(x_k)Tj(x_k) = \frac{2}{n+1}\sum_{k=0}^{n} f(x_k)T_j(\cos\frac{2k+1}{2n+2}\pi)$$

Interpolare trigonometrica

Considerand functia f *periodica* si avand perioada 2Π, suportul interpolarii este reprezentat de *2n+1 puncte* care impart intervalul [0,2Π] in *n* intervale egale:

$$\theta_k = \frac{2k\pi}{2n+1}, k = 0:2n$$

Baza interpolarii:

$$\frac{1}{\sqrt{2}}, \sin\theta, \cos\theta, \sin 2\theta, \cos 2\theta, ..., \sin n\theta, \cos n\theta.$$

Polinomul generalizat de interpolare:

$$P_n(\theta) = \frac{a_0}{\sqrt{2}} + b_1\sin\theta + a_1\cos\theta + ... + b_n\sin n\theta + a_n\cos n\theta$$

Solutia sistemului de ecuatii obtinut din conditiile de interpolare:

$$a_0 = \frac{\sqrt{2}}{2n+1}\sum_{k=0}^{2n} f(\frac{2k\pi}{2n+1})$$

$$b_j = \frac{2}{2n+1}\sum_{k=1}^{2n} f(\theta_k)\sin j\theta_k = \frac{2}{2n+1}\sum_{k=1}^{2n} f(\frac{2k\pi}{2n+1})\sin\frac{2k\pi}{2n+1}, j = 1:n$$

$$a_j = \frac{2}{2n+1}\sum_{k=0}^{2n} f(\theta_k)\cos j\theta_k = \frac{2}{2n+1}\sum_{k=0}^{2n} f(\frac{2k\pi}{2n+1})\cos\frac{2k\pi}{2n+1}, j = 1:n$$

Functii spline de interpolare

Interpolarea polinomiala nu este stabila numeric, limitand gradul polinomului de interpolare. Aceasta situatie conduce la o aproximare locala, pe intervale prin polinoame de grad mic(uzual 3).

Un spline cubic de interpolare este definit ca:

$$S_i:[x_i,x_{i+1})\to R, S_i(x)=a_i+b_i(x-x_i)+c_i(x-x_i)^2+d_i(x-x_i)^3$$

In general, in calculul splineurilor se foloseste baza Bernstein, care are urmatorul suport:

$$(1-t)^3, 3t(1-t)^2, 3t^2(1-t), t^3$$

<u>Splineuri de clasa C1</u>:

Conditii de interpolare Hermite:

$$\begin{aligned}
&S_i(x_i)=f(x_i), i=0:n-1\\
&S_i'(x_i)=f'(x_i),\\
&S_{n-1}(x_n)=f(x_n),\\
&S_{n-1}'(x_n)=f'(x_n).
\end{aligned}$$

Conditii de continututitate si derivabilitate in nodurile interne:

$$\begin{aligned}
&S_i(x_{i+1})=S_{i+1}(x_{i+1}), i=0:n-2\\
&S_i'(x_{i+1})=S_{i+1}'(x_{i+1}),
\end{aligned}$$

<u>Splineuri de clasa C2:</u>

Conditii de interpolare de tip Lagrange:

$$\begin{aligned}
&S_i(x_i)=f(x_i), i=0:n-1\\
&S_{n-1}(x_n)=f(x_n).
\end{aligned}$$

Conditii de continuitate, derivabilitate si curbura in nodurile interne:

$$\begin{aligned}
&S_i(x_{i+1})=S_{i+1}(x_{i+1}), i=0:n-2\\
&S_i'(x_{i+1})=S_{i+1}'(x_{i+1}),\\
&S_i''(x_{i+1})=S_{i+1}''(x_{i+1}).
\end{aligned}$$

- splineuri naturale:

$$S_0''(x_0)=S_{n-1}''(x_n)=0$$

– splineuri tensionate:

$$S_0'(x_0) = f'(x_0),$$
$$S_{n-1}'(x_n) = f'(x_n).$$

Exercitii:

1. Scrieti o functie ce calculeaza valoarea polinomului de interpolare *Lagrange* in punctul a, cunoscand vectorii x si y care determina suportul de interpolare.(**2p**)

function l=Lagrange(a,x,y)

2. Se considera f o functie periodica impara, cu perioada 2Π, cunoscuta prin valorile:

x = Π/4 => F(x) = -1
x = Π/2 => F(x) = 0
x = 3Π/4 => F(x) = 1

a) Scrieti o functie pentru calculul *coeficientilor polinomului de interpolare* sub forma Newton. (**1p**)

function [z] = DiferenteDivizate(x,y)

b) Scrieti o functie pentru calculul *valorii polinomului de interpolare* sub forma *Newton* intr-un punct dat.(**2p**)

function NewtonVal = Newton(a,x,y)

3. 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.(**3p**)

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.(**2p**)

function s = SplineC2(a,x,y)

4. **BONUS**: Realizati un grafic al conturului mainii voastre:). Porniti de la urmatoarele instructiuni:

```
figure('position', get(0,'screensize'));
axes('position',[0 0 1 1]);
[x,y] = ginput;
```

Asezati-va palma pe ecran (sau un contur al palmei pe o foaie de hartie). Folosind mouse-ul selectati suficient de multe puncte pentru a determina conturul palmei. Salvati punctele selectate si folositi-le ca suport al interpolarii. Reprezentati grafic rezultatele obtinute.