

MEF-AI.1.4

APROXIMAREA FUNCȚIILOR

CUPRINS

MEF-AI.1.4.1 INTRODUCERE

MEF-AI.1.4.2 INTERPOLAREA CU POLINOAME ALGEBRICE

MEF-AI.1.4.3 INTERPOLAREA CU POLINOAME LAGRANGE

MEF-AI.1.4.4 FUNCȚII MATLAB

MEF-AI.1.4.1 INTRODUCERE

Fundamentarea teoretică riguroasă din punct de vedere matematic, a Metodei Elementelor Finite (MEF), presupune cunoștințe avansate de modelare matematică analitică și numerică, în corelație. Deoarece această lucrare nu se adresează celor implicați în cercetarea teoretică fundamentală a MEF, în acest capitol se prezintă elemente aproximarea funcțiilor necesare înțelegerii, implementării și aplicării acestei metode de către utilizatori și programatori.

Problema aproximării funcțiilor este frecvent întâlnită și este necesară în aplicațiile ingineresti, inclusiv în cazul MEF. Mărimile fizice măsurate sau calculate pot fi exprimate ca funcții discrete, prin relații tabelare între mulțimi discrete și/sau funcții analitice, prin corespondențe între elementele unor intervale.

Deseori, se impune prelucrarea numerică a acestor funcții, în scopul cunoașterii valorilor acestora pentru argumente diferite de punctele tabelate, în cazul primului tip de funcție și generarea unei noi dependențe, din considerente teoretice și/sau de simplitate, pentru cel de-al doilea tip de funcție.

Pentru simplitate, se consideră cazul funcției reale de o variabilă reală $f: [a,b] \rightarrow \mathbb{R}$, care în punctele $x_i \in [a,b]$ ia valorile $y_i = f(x_i)$, cu $i = 1, 2, \dots, n$. Aproximarea funcției f pe intervalul $[a,b]$ presupune determinarea unei funcții de aproximare $F: [a,b] \rightarrow \mathbb{R}$, suficient de simplă și de generală, care să permită studiul funcției f cu abateri controlate. Expresia funcției $F = F(x, \alpha_j)$ conține parametrii α_j , care se determină din condiția de apropiere a funcției aproximative F de funcția f . Această apropiere se cuantifică prin „distanța”

$$d(f, F) = \sqrt{\int_a^b (f(x) - F(x, \alpha_j))^2 dx}, \quad (1)$$

când se cunoaște expresia analitică a funcției f sau

$$d(f, F) = \sqrt{\sum_{i=1}^n (f(x_i) - F(x_i, \alpha_j))^2}, \quad (2)$$

pentru funcții definite sub formă de tabel.

Aproximarea pe baza acestor definiții se poate realiza cu ajutorul a două metode numerice, regresia și interpolarea.

Regresia (ajustarea) sau aproximarea prin *metoda celor mai mici pătrate* presupune determinarea parametrilor α_j ai funcției de aproximare $F(x, \alpha_j)$, astfel încât să se minimizeze „distanța” dintre F și f , dată de relația (1) sau (2). Astfel, parametrii α_j pot fi determinați prin rezolvarea sistemului de ecuații obținute din relația

$$\frac{\partial d(f, F)}{\partial \alpha_j} = 0, \quad (3)$$

cu $j = 1, 2, \dots, n$.

Funcția $y = F(x, \alpha_j)$ astfel obținută nu trece prin punctele tabelate $M(x_i, y_i)$, ci prin puncte adiacente acestora (fig. 1,a). Aproximarea prin regresie se aplică frecvent funcțiilor sub formă de tabele, care sunt cu

erori, de regulă, provenite în urma măsurărilor.

Interpolarea se aplică în cazul când valorile tabelate $y_i = f(x_i)$ sunt exacte și se impune ca funcția aproximativă (funcția de interpolare) să conțină aceste valori, adică

$$F(x_i) = y_i, \quad (4)$$

pentru $i = 1, 2, \dots, n$ (fig. 1,b). Funcția F care respectă condiția (4) este utilizată pentru aproximarea funcției $f(x)$ când argumentul $x \in [x_i, x_n]$.

Deoarece, cu precădere, MEF utilizează aproximarea funcțiilor prin interpolare, în continuare, se vor prezenta câteva metode frecvent folosite.

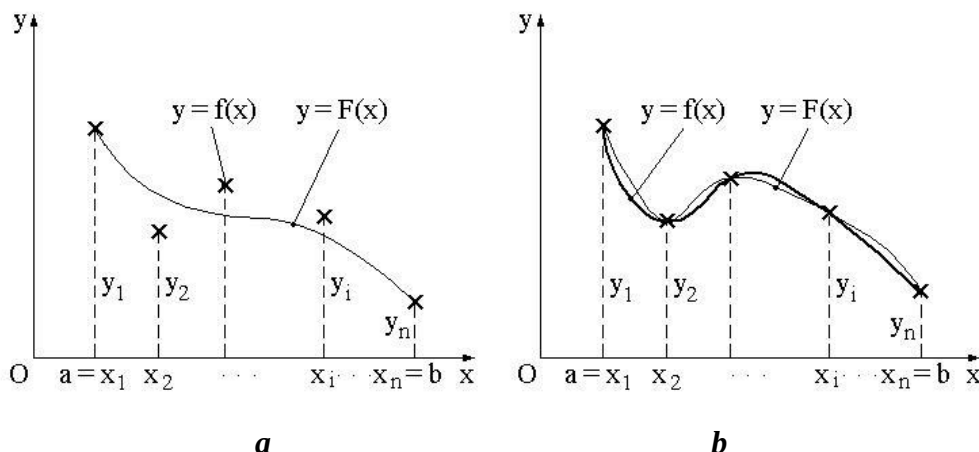


Fig. 1

MEF-AI.1.4.2 INTERPOLAREA CU POLINOAME ALGEBRICE

Foarte frecvent, datorită simplității, în procesul de aproximare oarecare se folosesc funcții polinomiale ale căror expresii se obțin prin combinații liniare ale următoarelor monoame independente: $1, x, x^2, x^3, \dots, x^n$. Pentru cazul unei funcții monovariabile cu n puncte cunoscute (fig. 1.1,b), se poate considera ca funcție de aproximare funcția polinomială

$$y(x) = c_0 + c_1 x + \dots + c_{n-1} x^{n-1}, \quad (5)$$

având coeficienții c_i , cu $i = 0, 1, \dots, n-1$, necunoscuți.

În urma particularizării polinomului propus pentru punctele de interpolare $M_i(x_i, y_i)$, cu $i = 1, 2, \dots, n$, rezultă sistemul liniar,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}, \quad (6)$$

care poate fi rezolvat în raport cu necunoscutele c_i , cu una din metodele prezentate în cap MEF-AI.1.2.

Aproximarea folosind polinoamele algebrice, cu algoritmul prezentat, se aplică pentru cazul funcțiilor cu un număr redus de puncte, când și volumul de calcule este redus.

MEF-AI.1.4.3 INTERPOLAREA CU POLINOAME LAGRANGE

Pentru obținerea polinomului de interpolare a unei funcții cu n puncte distincte, cunoscute, Lagrange a propus următoarea formă:

$$y(x) = L_1(x)y_1 + L_2(x)y_2 + \dots + L_n(x)y_n, \quad (7)$$

în care

$$\begin{aligned} L_1(x) &= \frac{(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)}, \\ L_2(x) &= \frac{(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)}, \\ &\dots \\ L_i(x) &= \frac{(x-x_1)(x-x_2)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_1)(x_i-x_2)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \\ &\dots \\ L_n(x) &= \frac{(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\dots(x_n-x_{n-1})}. \end{aligned} \quad (8)$$

Coeficienții $L_i(x)$ sunt polinoame de gradul $n-1$ și au următoarele proprietăți caracteristice:

$$L_i(x_j) = \begin{cases} 1, & \text{pentru } j = i; \\ 0, & \text{pentru } j \neq i; \end{cases} \quad (9)$$

$$\sum_{i=1}^n L_i(x) = 1. \quad (10)$$

Subprogramul LAGRANGE, fig. 2, permite interpolarea unei funcții monovariabile, reprezentată tabelar prin n puncte distincte.

```

Procedure LAGRANGE(var n:integer;xi,yi:vect;x,y:real);
{type vect=array[1..n] of real;}
var i,j:integer;p:real;
begin
  y:=0.0;
  for i:= 1 to n do
    begin
      p:=1.0;
      for j:=1 to n do
        if j <> i then p:=p*(x-xi[j])/(xi[i]-xi[j]);
      y:=y+p*yi[i];
    end;
  end;
end;

```

Fig. 2

În vederea determinării punctelor suprafeței S (fig. 3,a), cu funcția necunoscută, când se cunosc valorile acestora z_i în punctele $P_i(x,y)$, $i=1, 2, \dots, 9$, se folosește următorul algoritm.

- Se consideră variabila y constantă, de valoare y_1, y_2 , și y_3 , se aplică succesiv formula de interpolare (1.98), utilizând valorile cunoscute ale funcției și rezultă (fig. 3,b):

$$z_I(x, y_1) = L_1(x)z_1 + L_2(x)z_2 + L_3(x)z_3, \quad (11)$$

cu,

$$\begin{aligned} L_1(x) &= \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}, \quad L_2(x) = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}, \quad L_3(x) = \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}; \\ z_{II}(x, y_2) &= L_4(x)z_4 + L_5(x)z_5 + L_6(x)z_6, \end{aligned} \quad (12)$$

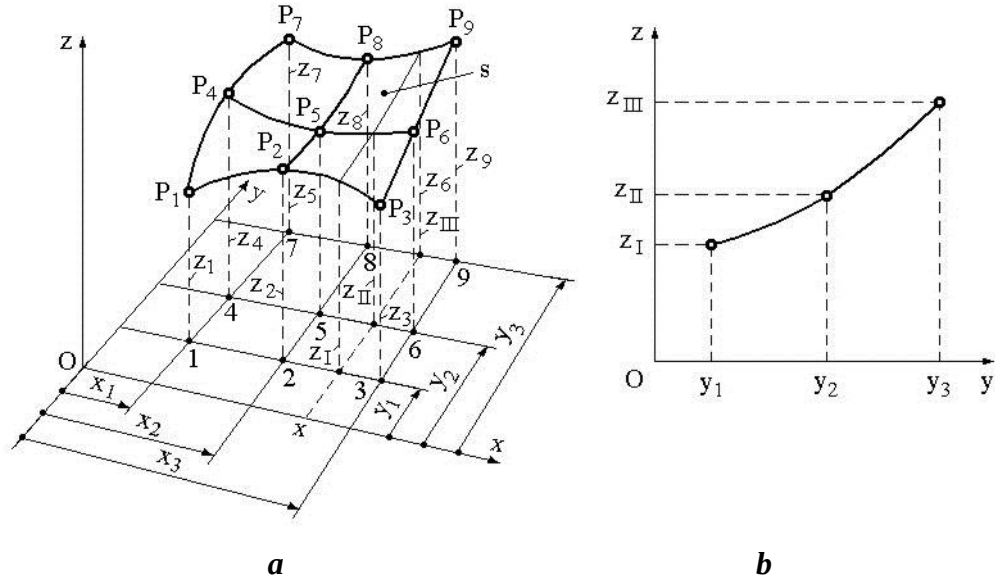


Fig. 3

cu,

$$L_4(x) = \frac{(x - x_5)(x - x_6)}{(x_4 - x_5)(x_4 - x_6)}, \quad L_5(x) = \frac{(x - x_4)(x - x_6)}{(x_5 - x_4)(x_5 - x_6)}, \quad L_6(x) = \frac{(x - x_4)(x - x_5)}{(x_6 - x_4)(x_6 - x_5)};$$

$$z_{III}(x, y_3) = L_7(x)z_7 + L_8(x)z_8 + L_9(x)z_9, \quad (13)$$

cu,

$$L_7(x) = \frac{(x - x_8)(x - x_9)}{(x_7 - x_8)(x_7 - x_9)}, \quad L_8(x) = \frac{(x - x_7)(x - x_9)}{(x_8 - x_7)(x_8 - x_9)}, \quad L_9(x) = \frac{(x - x_7)(x - x_8)}{(x_9 - x_7)(x_9 - x_8)}.$$

- Se continuă interpolarea în raport cu variabila y, cu relația

$$z(x, y) = L_1(y)z_1(x, y_1) + L_2(y)z_2(x, y_2) + L_3(y)z_3(x, y_3), \quad (14)$$

în care,

$$L_1(y) = \frac{(y - y_2)(y - y_3)}{(y_1 - y_2)(y_1 - y_3)}, \quad L_2(y) = \frac{(y - y_1)(y - y_3)}{(y_2 - y_1)(y_2 - y_3)}, \quad L_3(y) = \frac{(y - y_1)(y - y_2)}{(y_3 - y_1)(y_3 - y_2)}.$$

- Înlocuind relațiile (11), (12) și (13) în (14) și ținând cont că pentru $x_1 = x_4 = x_7$, $x_2 = x_5 = x_8$ și $x_3 = x_6 = x_9$, $L_1(x) = L_4(x) = L_7(x)$, $L_2(x) = L_5(x) = L_8(x)$ și, respectiv, $L_3(x) = L_6(x) = L_9(x)$, rezultă

$$z(x, y) = +L_1(y)L_1(x)z_1 + L_1(y)L_2(x)z_2 + L_1(y)L_3(x)z_3 + \\ + L_2(y)L_1(x)z_4 + L_2(y)L_2(x)z_5 + L_2(y)L_3(x)z_6 + \\ + L_3(y)L_1(x)z_7 + L_3(y)L_2(x)z_8 + L_3(y)L_3(x)z_9. \quad (15)$$

unde,

$$L_1(s) = \frac{(s - s_2)(s - s_3)}{(s_1 - s_2)(s_1 - s_3)}, \quad L_2(s) = \frac{(s - s_1)(s - s_3)}{(s_2 - s_1)(s_2 - s_3)}, \quad L_3(s) = \frac{(s - s_1)(s - s_2)}{(s_3 - s_1)(s_3 - s_2)}.$$

Cu, $s = x$ sau y .

Această formulă, de interpolare a funcțiilor bivariabile, ce folosește nouă puncte cunoscute, poate fi

generalizată, pe de-o parte, pentru cazul mai multor puncte necunoscute și, pe de altă parte, pentru interpolarea funcțiilor invariabile.

Folosirea polinoamelor Lagrange pentru interpolare permite obținerea directă a funcției de aproximare fără rezolvarea sistemului de ecuații de tipul (2).

MEF-AI.1.4.4 FUNCȚII MATLAB

În tab. 1 s prezintă funcția *interp1* cu câteva exemple de utilizare.

Tab. 1

Nr. crt.	Funcția	Semnificație rezultat	Exemple	Rezultat
1	interp1	Matrice cu valorile funcției de interpolare	<pre>x=0:pi/4:pi; y=sin(x); xi=0:pi/100:pi; yi=interp1(x,y,xi,'nearest'); plot(xi,yi)</pre>	
2			<pre>x=0:pi/4:pi; y=sin(x); xi=0:pi/100:pi; yi=interp1(x,y,xi,'linear'); plot(xi,yi)</pre>	
3			<pre>x=0:pi/4:pi; y=sin(x); xi=0:pi/100:pi; yi=interp1(x,y,xi,'cubic'); plot(xi,yi)</pre>	
4	interp1		<pre>[x, y] = meshgrid(-1:0.5:1); z = sin(x.^2+y.^2); [xi, yi] = meshgrid(-1:.05:1); zi = interp2(x, y, z, xi, yi, 'linear'); surf(xi, yi, zi)</pre>	
5			<pre>[x, y] = meshgrid(-1:0.5:1); z = sin(x.^2+y.^2); [xi, yi] = meshgrid(-1:.05:1); zi = interp2(x, y, z, xi, yi, 'cubic'); surf(xi, yi, zi)</pre>	