8. APROXIMAREA NUMERICĂ A FUNCŢIILOR

Obiectivele lucrării:

- familiarizarea cu câteva din metodele de aproximare numerică bazate pe utilizarea polinoamelor,
- fixarea de cunoştinţe privitoare la rezolvarea problemei de aproximare numerică a funcţiilor prin diferite metode, utilizând mediul Matlab,

prin studierea unor exemple şi prin rezolvarea unor probleme.

Se recomandă parcurgerea anexei M8 înaintea studierii paragrafelor 8.1 şi 8.2.

8.1. ELEMENTE DESPRE APROXIMAREA NUMERICĂ A FUNCŢIILOR ÎN MATLAB

Aproximarea numerică a funcţiilor

Fie `f` o funcţie precizată printr-un set de `n` puncte (x_i, y_i=f(x_i)), i=1,2,...,n.

a. Interpolare liniară. Interpolare cu polinom Hermite. Interpolare cu funcţii spline

Pentru aproximarea numerică a funcţiei `f` de o variabilă reală prin ***interpolare liniară***, ***interpolare cu polinom Hermite*** sau ***interpolare cu funcţii spline***, Matlab pune la dispoziţia utilizatorului funcţia `interp1` cu următoarea sintaxă de apel:

```
vy=interp1(x,y,vx,'metoda')
```

unde:

- `x` este vectorul punctelor {x_i};
- `y` este vectorul punctelor {y_i};
- `vx` este vectorul punctelor în care se doreşte aproximarea funcţiei `f`;
- `vy` este vectorul obţinut prin aproximarea funcţiei `f` în punctele `vx`;
- `metoda` reprezintă un şir de caractere prin care se precizează metoda de interpolare dorită:
  - *`linear`* – pentru interpolare liniară (este metoda implicită);
  - *`cubic`* sau *`pchip`* – pentru interpolare cu polinom Hermite cubic pe porţiuni;
  - *`spline`* – pentru interpolare spline cubică.

Interpolarea cu funcții spline se mai poate realiza şi cu funcția Matlab `spline`. Sintaxa de apel a funcției `spline` este:

```
vy=spline(x,y,vx)
```

în care parametrii `x,y,vx` şi `vy` au aceleaşi semnificații ca şi în cazul funcției `interp1`.

Comentariu: De fapt, methoda '`spline`' a funcției `interp1` apelează fără implicarea utilizatorului funcția Matlab `spline` pentru realizarea interpolării.

b. Aproximare cu metoda celor mai mici pătrate

Aproximarea funcției `f` cu metoda celor mai mici pătrate se realizează în Matlab prin parcurgerea următoarelor două etape:

1. Determinarea polinomului de interpolare folosind funcția Matlab `polyfit`, care are sintaxa:

```
P=polyfit(x,y,m)
```

unde:

- `x` este vectorul punctelor $\{x_i\}$;
- `y` este vectorul punctelor $\{y_i\}$;
- `m` este gradul polinomului de aproximare, $0 \leq m \leq n-1$;
- `P` este vectorul coeficienților polinomului de interpolare, coeficienții fiind în ordinea descrescătoare a puterilor nedeterminatei polinomului.

2. Calculul valorilor polinomului de aproximare în punctele dorite, utilizând funcția Matlab `polyval`, care are sintaxa:

```
vy=polyval(P,vx)
```

unde:

- `vx` este vectorul punctelor în care se doreşte aproximarea funcției `f`;
- `P` este vectorul coeficienților polinomului de interpolare, obținut în prima etapă;
- `vy` este vectorul obținut prin aproximarea funcției `f` în punctele `vx` cu valorile polinomului `P`.

8.2. Exemple de studiat

<u>Exemplul 8.1</u>: Fie *f* o funcție reală de o variabilă reală, precizată prin tabelul de valori:

Tabelul 8.1. *Tabel de date pentru exemplul 8.1.*

x_i	0	1.2	1.6	2	2.7	3
$y_i=f(x_i)$	-2.5	0	2	1.7	-4	1

Să se determine valorile de aproximare ale funcției *f* în punctele 0.7, 1.3, 1.7, 2.5, 2.9, folosind:

a) metoda de interpolare liniară;

b) metoda de interpolare cu polinom Hermite cubic pe porțiuni.

Soluție: a) Se execută următoarea secvență de instrucțiuni Matlab:

```
% vectorul punctelor xi
x=[0 1.2 1.6 2 2.7 3];
% vectorul valorilor yi ale functiei in punctele xi
y=[-2.5 0 2 1.7 -4 1];
% vectorul punctelor in care se face aproximarea
vx=[0.7 1.3 1.7 2.5 2.9];
% aproximarea prin interpolare liniara
disp('Valorile lui f obtinute prin interpolare liniara:')
vy=interp1(x,y,vx,'linear')
```

Executând secvența de mai sus se obțin rezultatele:

```
Valorile lui f obtinute prin interpolare liniara:
vy =   -1.0417    0.5000    1.9250   -2.3714   -0.6667
```

b) Se execută următoarea secvență Matlab, care diferă de cea de la punctul a) numai prin metoda de interpolare precizată:

```
% vectorul punctelor xi
x=[0 1.2 1.6 2 2.7 3];
% vectorul valorilor yi ale functiei in punctele xi
y=[-2.5 0 2 1.7 -4 1];
% vectorul punctelor in care se face aproximarea
vx=[0.7 1.3 1.7 2.5 2.9];
% aproximarea prin interpolare Hermite cubica
disp(['Valorile lui f obtinute prin interpolare '...
    'Hermite cubica: '])
vy=interp1(x,y,vx,'cubic')
```

obținând valorile:

```
Valorile lui f obtinute prin interpolare Hermite cubica:

vy =   -1.4777    0.4901    1.9771   -2.9221   -1.3678
```

Exemplul 8.2: Să se reprezinte grafic în aceeaşi fereastră grafică funcțiile de interpolare liniară, respectiv spline cubică, pentru funcția *f* definită prin tabelul de valori:

Tabelul 8.2. *Tabel de date pentru exemplul 8.2.*

x_i	-1.5	0	1	3
$y_i=f(x_i)$	7.8	5	6.3	6.8

precum şi punctele tabelului.

Soluție: Intervalul de reprezentare este dat de cea mai mică valoare x_i şi cea mai mare valoare x_i: [-1.5,3]. Se execută următoarea secvență Matlab:

```
% vectorul punctelor xi
x=[-1.5 0 1 3];
```

```
% vectorul valorilor yi ale functiei in punctele xi
y=[7.8 5 6.3 6.8];
% vectorul punctelor intervalului de aproximare
vx=-1.5:0.1:3;
% aproximarea prin interpolare liniara
vy_liniar=interp1(x,y,vx,'linear');
% aproximarea prin interpolare spline cubica
vy_spline=spline(x,y,vx);
% reprezentarile grafice
plot(x,y,'ro',vx,vy_liniar,'g',vx,vy_spline,'b--')
legend('puncte','interpolare liniara','spline')
```

Graficele rezultate sunt prezentate în figura 8.1.:

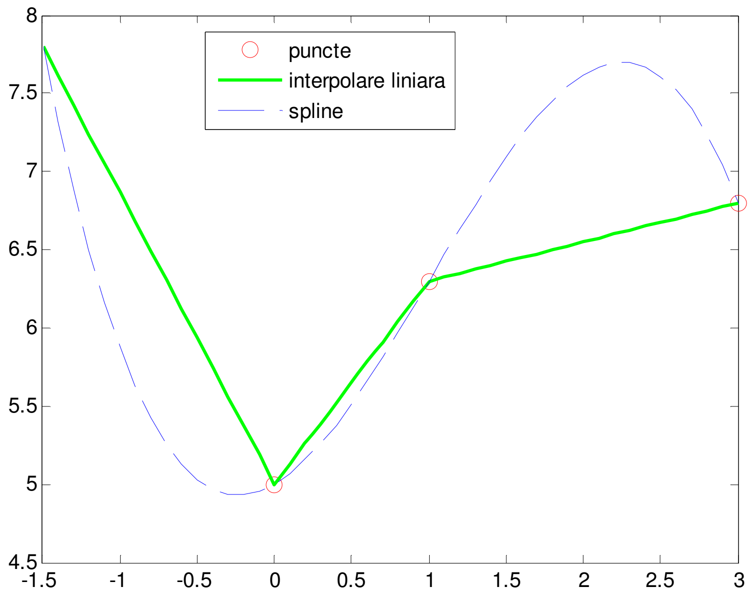

Fig.8.1. *Graficele funcțiilor de aproximare din exemplul 8.2.*

Exemplul 8.3: Fie funcția f dată prin tabelul de valori:

Tabelul 8.3. *Tabel de date pentru exemplul 8.3.*

x_i	-8	-6	-4	-2	0	2	4
$y_i=f(x_i)$	30	10	9	6	5	4	4

a) Să se aproximeze, în sensul celor mai mici pătrate, funcția f în punctele -7, -4.2, -0.75, 1, 2.15, 3, folosind regresia parabolică, regresia cubică şi regresia cu polinom de gradul 6.

b) Să se reprezinte grafic în aceeaşi fereastră grafică polinoamele de aproximare de la punctul a) şi punctele din tabel.

Soluție: a) Pentru aproximarea funcției date prin metoda celor mai mici pătrate (regresie polinomială), se parcurg cele două etape precizate în paragraful 8.1. Se execută secevnța Matlab:

```
% Etapa 1: determinarea polinoamelor de aproximare
% vectorul punctelor xi
x=-8:2:4;
% vectorul valorilor yi ale functiei in punctele xi
y=[30 10 9 6 5 4 4];
% vectorii coeficientilor polinoamelor de interpolare
P2=polyfit(x,y,2);   % regresie parabolica
P3=polyfit(x,y,3);   % regresie cubica
P6=polyfit(x,y,6);   % regresie de gradul 6

% Etapa 2: aproximarea
% vectorul punctelor in care se face aproximarea
vx=[-7 -4.2 -0.75 1 2.15 3];
% aproximarea prin regresie parabolica
disp('cmmp: regresie parabolica:')
vy2=polyval(P2,vx)
% aproximarea prin regresie cubica
disp('cmmp: regresie cubica:')
vy3=polyval(P3,vx)
% aproximarea prin regresie de gradul 6
disp('cmmp: regresie de gradul 6:')
vy6=polyval(P6,vx)
```

obținând următoarele rezultate:

```
cmmp: regresie parabolica:
vy2 =
   20.8929    9.9529    3.1473    2.5119    3.1266    4.1071
cmmp: regresie cubica:
vy3 =
   20.0595    7.1222    4.9833    5.6786    5.6114    4.9405
cmmp: regresie de gradul 6:
vy6 =
   13.6680    9.2095    5.0584    4.8867    3.7901    2.6289
```

b) Şi pentru reprezentarea grafică a funcțiilor de aproximare este necesară parcurgerea celor două etape precizate în paragraful 8.1. Prima etapă este însă identică cu prima etapa de la punctul a). Mai jos este redată secvența de instrucțiuni corespunzătoare celei de a doua etape, iar graficele obținute sunt redate în figura 8.2.:

```
% Etapa 2: graficele polinoamelor de aproximare
% intervalul de reprezentare grafica
vxg=min(x):(max(x)-min(x))/100:max(x);
% aproximarea prin regresie parabolica
vy2g=polyval(P2,vxg);
% aproximarea prin regresie cubica
vy3g=polyval(P3,vxg);
% aproximarea prin regresie de gradul 6
vy6g=polyval(P6,vxg);
% grafice
% punctele tabelului
plot(x,y,'bd')
hold on
% polinoamele de aproximare
plot(vxg,vy2g,'r',vxg,vy3g,'--g',vxg,vy6g,'k:')
axis([min(x)-1 max(x)+1 ...
    min([y vy2g vy3g vy6g])-5 max([y vy2g vy3g vy6g])+5])
```

```
legend('puncte','regresie parabolica', ...
    'regresie cubica','regresie de gradul 6')
hold off
```

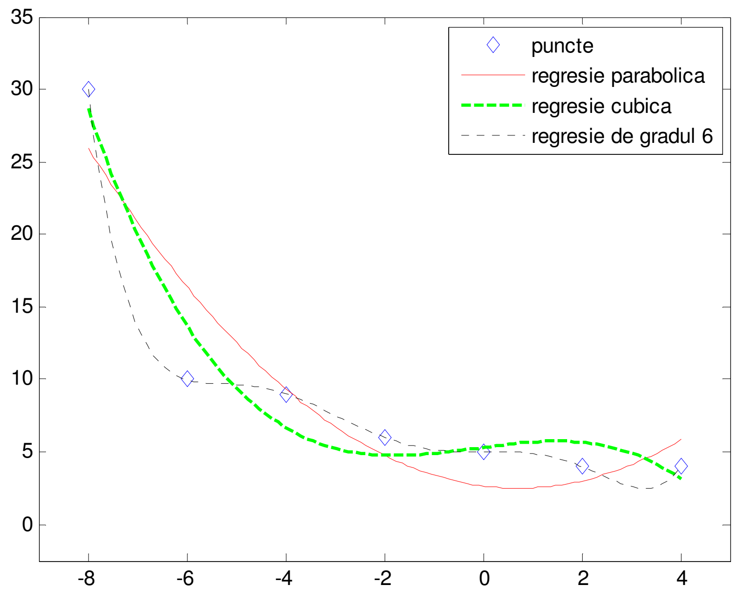

Fig.8.2. *Graficele funcțiilor de aproximare din exemplul 8.3.*

8.3. Probleme de rezolvat

P8.1. Să se scrie o funcție Matlab *langrange* care realizează interpolarea polinomială Lagrange. Argumentele funcției Matlab vor fi: vectorul punctelor $\{x_i\}$ şi vectorul punctelor $\{y_i\}$ ale unei funcții f cunoscută prin puncte şi vectorul *vx* al punctelor în care se doreşte realizarea aproximării. Funcția Matlab va returna vectorul *vy* al valorilor obținute prin aproximarea funcției f cu polinomul de interpolare Lagrange în punctele *vx*.

P8.2. Se consideră tabelul următor care reprezintă valorile vitezei unui mobil în mişcare citite la diverse momente de timp:

timp [sec]	0	1	2	3	4	5	6
viteza [m·sec^{-1}]	15	30	75	60	60	40	55

Să se estimeze valorile vitezei la momentele $t1$=0.5 sec, $t2$=3.2 sec şi $t3$=5.7 sec, utilizând:

a) interpolarea liniară;

b) interpolarea polinomială Lagrange (se va folosi funcția de la P8.1.);
c) interpolarea cu polinom Hermite cubic pe porțiuni;
d) interpolarea spline cubică;
e) regresia parabolică;
f) aproximarea cu metoda celor mai mici pătrate cu polinom de grad 5.

P8.3. În ipotezele problemei P8.2., să se reprezinte grafic în aceeaşi fereastră grafică punctele din tabel şi funcțiile de aproximare corespunzătoare punctelor a)-f).

P8.4. Să se scrie o funcție Matlab care primeşte ca argumente vectorul punctelor $\{x_i\}$, vectorul punctelor $\{y_i\}$ ale unei funcții f cunoscută prin puncte, şi vectorul vx al punctelor în care se doreşte realizarea aproximării. Funcția Matlab returnează vectorii obținuți prin interpolare liniară şi prin regresie cubică a funcției f în punctele vx.

P8.5. Să se scrie o funcție Matlab care primeşte ca argumente vectorul punctelor $\{x_i\}$ şi vectorul punctelor $\{y_i\}$ ale unei funcții f cunoscută prin puncte. Funcția Matlab reprezintă grafic în aceeaşi fereastră grafică, punctele de coordonate (x_i, y_i) şi funcțiile de aproximare corespunzătoare obținute prin interpolarea spline cubică şi interpolare cu polinom Hermite cubic pe porțiuni.

8.4. Întrebări recapitulative

Î8.1. Definiți *problema de aproximare a unei funcții reale de o variabilă reală.*

Î8.2. În ce situații se pune, de obicei, problema aproximării unei funcții?

Î8.3. Definiți noțiunile de *interpolare*, respectiv *extrapolare.*

Î8.4. Precizați formula polinomului de interpolare Lagrange, pentru o funcție f, cunoscută prin puncte $(x_i, y_i=f(x_i))$, $i=1,2,\ldots,n$.

Î8.5. Enumerați funcțiile Matlab (pe care le cunoaşteți) destinate aproximării funcțiilor de o variabilă reală (denumire, metoda de aproximare).

Î8.6. Care din următoarele metode de aproximare a funcțiilor impun ca funcția de aproximare g să aibă în punctele $\{x_i\}$ aceleaşi valori ca funcția care este aproximată f (adică, $g(x_i)=f(x_i)$): interpolare cu polinom Hermite, regresie parabolică, aproximare cu funcții spline, aproximare polinomială cubică prin metoda celor mai mici pătrate, interpolare cu polinom Lagrange?

ANEXA M8. ELEMENTE DESPRE APROXIMAREA NUMERICĂ A FUNCŢIILOR

M8.1. Aproximarea numerică a funcţiilor

a. Problema aproximării unei funcţii

Fie o funcție $f: I \subseteq \boldsymbol{R} \to \boldsymbol{R}$ şi $[a,b]$ un subinterval al domeniului de definiție I. Se pune problema determinării unei alte funcții $g : I \to \boldsymbol{R}$, de expresie relativ simplă, care să aproximeze cât mai bine funcția f pe intervalul $[a,b]$, adică $g(x) \approx f(x)$, $\forall x \in [a,b]$.

Problema aproximării unei funcții se pune în următoarele două situații:

a) expresia funcției f este cunoscută, dar suficient de complicată, astfel încât utilizarea ei în calcule este incomodă sau duce la erori mari de calcul;

b) expresia funcției f nu este cunoscută, funcția fiind precizată doar printr-un set de n puncte $\{(x_i, y_i)\}$, $y_i = f(x_i)$, $i = \overline{1,n}$, cu $x_1 = a$, $x_n = b$, $x_i \in (a,b)$, $i = \overline{2,n-1}$.

Situația cea mai întâlnită este a doua, caz în care, valorile funcției f sunt date de obicei sub formă tabelară:

x_i	$x_1=a$	x_2	...	$x_n=b$
$y_i = f(x_i)$	y_1	y_2	...	y_n

În majoritatea aplicațiilor, valorile $\{x_i\}$ sunt echidistante, cu pasul de discretizare $h = \dfrac{b-a}{n}$, adică $x_{i+1} = x_i + h$, $i = \overline{1,n-1}$.

În multe situații practice, nu este necesară determinarea expresiei funcției de aproximare g, ci doar a valorilor de aproximare $g(x)$ pentru orice x din intervalul $[a,b]$.

În cazul b), dacă pentru funcția de aproximare g se impune condiția $g(x_i) = y_i$, $i = \overline{1,n}$, problema de aproximare este denumită şi problemă de ***interpolare***. Dacă problema se extinde şi în afara intervalului $[a,b]$, adică se doreşte aproximarea funcției f într-un punct $x \in I - [a,b]$, atunci se utilizează termenul de ***extrapolare***.

b. Câteva metode numerice de aproximare a funcţiilor

În continuare se presupune că funcția f este precizată printr-un set de n puncte, $\{(x_i, y_i)\}$, $y_i = f(x_i)$, $i = \overline{1,n}$, cu $x_1 = a$ şi $x_n = b$.

Interpolare liniară

Se cere să se determine funcția de aproximare g care să fie afină pe fiecare subinterval $[x_i, x_{i+1}]$, $i = \overline{1,n-1}$, şi să treacă prin punctele date, deci să verifice condițiile:

$g(x_i) = y_i,\ i = \overline{1,n}$.

Funcţia g cu proprietăţile cerute există şi este unică. Ea este dată de expresia:

$$g(x) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \cdot x + \frac{x_{i+1} y_i - x_i y_{i+1}}{x_{i+1} - x_i},\ x \in [x_i, x_{i+1}],\ i = \overline{1,n-1}$$

Interpolare polinomială Lagrange

Se cere să se determine polinomul P_{n-1} (funcţia g este, deci, o funcţie polinomială, $g = P_{n-1}$), grad $P_{n-1} = n-1$, de forma:

$$P_{n-1}(x) = a_1 + a_2 x + a_3 x^2 + ... + a_n x^{n-1},\ \forall x \in [a,b],\ a_i \in \boldsymbol{R},\ i = \overline{1,n},$$

care să treacă prin punctele date, deci să verifice condiţiile:

$$P_{n-1}(x_i) = y_i,\ i = \overline{1,n}.$$

Se obţine astfel următorul sistem de ecuaţii liniare, în necunoscutele $a_1, a_2, ..., a_n$:

$$\begin{cases} a_1 + a_2 x_1 + a_3 x_1^2 + ... + a_n x_1^{n-1} = y_1 \\ a_1 + a_2 x_2 + a_3 x_2^2 + ... + a_n x_2^{n-1} = y_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_1 + a_2 x_n + a_3 x_n^2 + ... + a_n x_n^{n-1} = y_n \end{cases}$$

Deoarece determinantul sistemului este un determinant Vandermonde şi punctele x_i, $i = \overline{1,n}$, sunt distincte două cîte două, sistemul are soluţie unică. Prin urmare, *există un polinom unic de gradul $n-1$ care să treacă prin punctele date.* Acest polinom se numeşte ***polinom de interpolare al punctelor (x_i,y_i), $i = \overline{1,n}$***.

Există mai multe modalităţi de determinare a polinomului de interpolare, care duc, de fapt, la exprimări echivalente ale acestuia: de tip Newton, de tip Gauss, de tip Stirling, de tip Bessel, de tip Lagrange ş.a. În continuare se face referire doar la ultimul.

Polinomul de interpolare Lagrange se determină în modul următor: se alege polinomul de aproximare de forma:

$$L_{n-1}(X) = \sum_{k=1}^{n} y_k p_k(X),$$

unde polinoamele p_k sunt de grad $n-1$ şi au proprietatea $p_k(x_i) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$.

Expresia acestor polinoame se poate deduce uşor:

$$p_k(X) = \prod_{\substack{i=1 \\ i \neq k}}^{n} \frac{X - x_i}{x_k - x_i}$$

obţinând, în final, expresia polinomului de interpolare Lagrange:

$$L_{n-1}(X)=\sum_{k=1}^{n}\left(y_k\prod_{\substack{i=1\\ i\neq k}}^{n}\frac{X-x_i}{x_k-x_i}\right)$$

Interpolare cu polinom Hermite

Se presupune că se cunosc pentru funcția f şi valorile derivatelor până la un anumit ordin în anumite puncte $\{x_i\}$:

$$f^{(r)}(x_i)=y_i^r,\quad r=\overline{0,r_i},\quad r_i\in \boldsymbol{N},\quad i=\overline{1,n}$$

Se cere să se determine polinomul P de grad minim care să îndeplinească următoarele condiții:

$$P^{(r)}(x_i)=y_i^r,\quad r=\overline{0,r_i},\ i=\overline{1,n}$$

Un astfel de polinom există, este unic şi are gradul $m=n+\sum_{i=1}^{n}r_i$. Se numeşte ***polinom de interpolare Hermite***.

Aproximarea cu funcții spline

Se cere să se aproximeze funcția f cu o funcție g spline polinomială de grad $m<<n$, astfel încât:

$$g(x_i)=y_i,\ i=\overline{1,n}$$

Prin ***funcție spline polinomială de grad m*** se înțelege o funcție de clasă $C_{[a,b]}^{m-1}$, ale cărei restricții g_i pe fiecare subinterval $[x_{i-1}, x_i]$ sunt polinoame de grad $m<<n$:

$$g_i(x)=P_m^i(x),\ \ \forall x\in[x_{i-1},x_i],\ i=\overline{2,n},\ gradP_m^i=m\ .$$

Dacă m=3 funcția g este numită ***funcție spline cubică***.

Aproximarea cu metoda celor mai mici pătrate

Se cere să se determine polinomul $P_m,\ grad\ P_m=m,\ \ m<n$, de forma:

$$P_m(x)=a_1+a_2x+a_3x^2+...+a_{m+1}x^m,\ \forall x\in[a,b]\ ,$$

care să aproximeze funcția f astfel încât să fie minimizată suma pătratelor diferențelor dintre valorile aproximate şi cele exacte în punctele $\{x_i\}$.

Problema enunțată este o problemă de optimizare:

$$\hat{P}_m=\{P_m \mid \min_{a_1,...,a_{m+1}}\sum_{i=1}^{n}[P_m(x_i)-y_i]^2\}\ .$$

Metoda de calcul rezultată se numeşte ***metoda celor mai mici pătrate (CMMP)***. Această metodă se foloseşte atunci când fie perechile $(x_i,\ y_i = f(x_i))$, $i=\overline{1,n}$, nu sunt cunoscute cu exactitate fie n este foarte mare.

Aproximarea funcției f printr-un polinom de forma de mai sus prin metoda CMMP este numită în general şi ***regresie polinomială***, în particular ***regresie***

liniară dacă m=1, ***regresie parabolică*** dacă m=2, respectiv ***regresie cubică*** dacă m=3.

Aproximarea prin metoda CMMP poate fi aplicată însă şi altor funcţii de aproximare g, diferite de cele polinomiale.