UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
AUTOMATICĂ ȘI INFORMATICĂ APLICATĂ

Proiect identificarea sistemelor-Regresii liniare

Alinei-Poiana Tudor, Ciurezu Gheorghe-Dragos, Kocis Bianca Laura

8/8

2021-2022

Cuprins

1 Descrierea problemei **3**

2 Structura aproximatorului **3**
2.1 Observație 3
2.2 Alte tipuri de regresori 4

3 Media erorilor pătratice și grafice reprezentative. **6**
3.1 Discuție pentru media erorilor pătratice. 6
3.2 Grafice reprezentative pentru aproximare. 8

4 Comparație între algoritmi **11**

5 Concluzii **13**

6 Anexă **14**
6.1 Funcția "regresie" 14
6.2 Script-ul care apeleaza funcția "regresie" 16
6.3 Funcția "aproximare" 16
6.4 Script-ul care apeleaza funcția "aproximator" 19
6.5 Regresie cu funcții de bază Gaussiene 20

1 Descrierea problemei

Pornind de la un set de date de tip intrare-ieșire, ieșire ce este generată de o funcție necunoscută, neliniară și statică, trebuie sa determinăm un model matematic ce să poată aproxima cât mai precis modelul matematic din spatele datelor exeperimentale oferite.

Funcția pe care o vom căuta va avea două variabile de intrare și una de ieșire, iar modelul căutat este cel al unui aproximator polinomial, obținut prin metoda "Regresiei liniare".

Odată găsit modelul ce aproximează setul de date de identificare, acesta trebuie să fie probat pe setul de date de validare pentru a asigura corectitudinea modelului obținut.

k poate fi o variabila de timp în funcție de care modelăm seria temporală y(k) sau, în cazul nostru, k este doar un index de eșantion, și $\varphi(k) = \phi(x(k))$, unde x este intrarea unei funcții necunoscute g.

Pentru aproximarea funcției, regresorii $\phi_i(k)$ din: $\phi(x(k)) = [\phi_1(x(k)), \phi_2(x(k)), ..., \phi_n(x(k))]^T$ se numesc funcții de bază.

2 Structura aproximatorului

Aproximatorul polinomial de grad m are următoarele forme:

Pentru m=1: $\hat{g}(x) = [1, x_1, x_2] \cdot \theta = \theta_1 + \theta_2 x_1 + \theta_3 x_2$

Pentru m=2: $\hat{g}(x) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2] \cdot \theta = \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \theta_6 x_1 x_2$

Pentru m=3: $\hat{g}(x) = [1, x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1 x_2, x_1^2 x_2, x_1 x_2^2] \cdot \theta$

unde θ este matricea parametrilor, iar [1,$x_1, x_2, x_1^2, x_2^2, ..., x_1^{m-k} \cdot x_2^k, ...$] este matricea regresorilor.

Rescriind datele problemei (pentru un aproximator de grad m) sub forma unui sistem în care considerăm toate combinațiile de tip $x_1(i) \cdot x_2(j), i = \overline{1,a}, j = \overline{1,b}$ vom obține următorul sistem supradeterminat cu necunoscutele $\theta = [\theta_1, \theta_2 ... \theta_{\frac{(m+1)(m+2)}{2}}]$:

$$\begin{cases} \hat{g}(x_1(1), x_2(1)) & = \theta_1 + \theta_2 x_1(1) + \theta_3 x_2(1) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(1)^{m-k} x_2^k(1) = y_{1,1} \\ \hat{g}(x_1(1), x_2(2)) & = \theta_1 + \theta_2 x_1(1) + \theta_3 x_2(2) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(1)^{m-k} x_2^k(2) = y_{1,2} \\ \hat{g}(x_1(1), x_2(3)) & = \theta_1 + \theta_2 x_1(1) + \theta_3 x_2(3) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(1)^{m-k} x_2^k(3) = y_{1,3} \\ & \vdots \\ \hat{g}(x_1(1), x_2(b)) & = \theta_1 + \theta_2 x_1(1) + \theta_3 x_2(b) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(1)^{m-k} x_2^k(b) = y_{1,b} \\ \hat{g}(x_1(2), x_2(1)) & = \theta_1 + \theta_2 x_1(2) + \theta_3 x_2(1) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(2)^{m-k} x_2^k(1) = y_{2,1} \\ \hat{g}(x_1(2), x_2(2)) & = \theta_1 + \theta_2 x_1(2) + \theta_3 x_2(2) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(2)^{m-k} x_2^k(2) = y_{2,2} \\ & \vdots \\ \hat{g}(x_1(a), x_2(b-1)) & = \theta_1 + \theta_2 x_1(a) + \theta_3 x_2(b-1) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(a)^{m-k} x_2^k(b-1) = y_{a,b-1} \\ \hat{g}(x_1(a), x_2(b)) & = \theta_1 + \theta_2 x_1(a) + \theta_3 x_2(b) + ... + \theta_{\frac{(m+1)(m+2)}{2}} x_1(a)^{m-k} x_2^k(b) = y_{a,b} \end{cases}$$

2.1 Observație

În funcție de gradul m al aproximatorului vom avea $\frac{(m+1)(m+2)}{2}$ coeficienți. Acest rezultat l-am obținut ținând cont că la fiecare incrementare cu 1 a gradului unui aproximator de variabile x_1, x_2

se vor adăuga încă m+1 termeni, de exemplu pentru gradul 3, pornind de la gradul 2 se vor adăuga pe langă cei 6 regresori deja existenți $[1, x_1, x_2, {x_1}^2, {x_2}^2, x_1x_2]$ încă 3+1=4 termeni corespunzători gradului 3: $[{x_1}^3, {x_2}^3, {x_1}^2x_2, x_1{x_2}^2]$.

Astfel, pentru gradul m vom obține:
$(0+1)+(1+1)+(2+1)+...+(m+1) = \frac{m(m+1)}{2} + m + 1 = \frac{(m+1)(m+2)}{2}$, rezultat ce va fi de folos și în implementarea ulterioară a unui algoritm în Matlab.

2.2 Alte tipuri de regresori

$$\phi_i(x) = exp\ [-\frac{(x-c_i)^2}{b_i^2}] \qquad \text{(1-dim)}$$

$$= exp[-\sum_{j=1}^{d} \frac{(x_j - c_i j)^2}{b_i j^2}] \qquad \text{(d-dim)}$$

Pentru aproximarea cu ajutorul RBF-urilor am considerat secțiuni din ieșirea Y între X1(i) și X1(i+σ) și X2(j) și X2(j+β). Pentru simplitate am considerat $\sigma = \beta$. Astfel, am aproximat fiecare dintre aceste porțiuni ale graficului ieșirii Y cu câte o exponențială de următoarea formă:

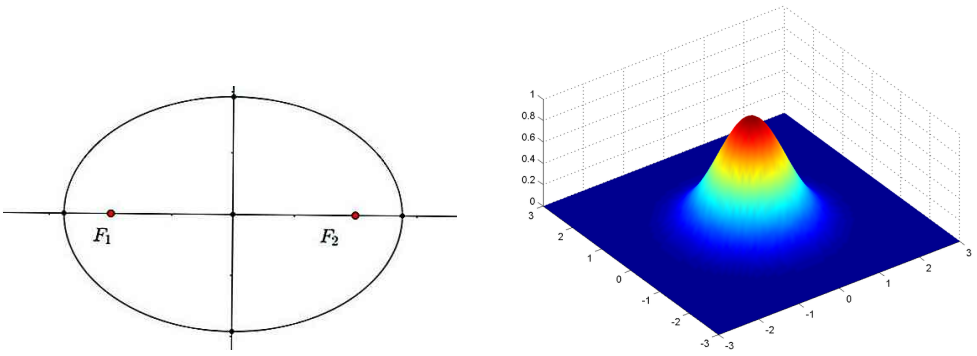

Figure 1: Elipsa și distribuția Gaussiană bidimensională

Dacă intersectăm exponențiala bidimensională reprezentată în partea dreaptă cu un plan paralel cu axa planul XOY vom obtine elipsa reprezentată în stânga. În cazul de față semiaxa mare este obținută prin diferența X1(i)-X1(i+σ) și semiaxa mică prin diferența X2(j)-X2(j+β), iar punctele din care pornesc aceste semiaxe sunt la mijlocul intervalului dintre X1(i) și X1(i+σ), respectiv X2(j) și X2(j+β).

În Figura 11 avem graficul aproximarii cu ajutorul funcțiilor de baza Gaussiene pentru datele de validare pentru un numar de RBF-uri (aproximativ 1849) unde MSE-ul este minim (MSE=8.1121).

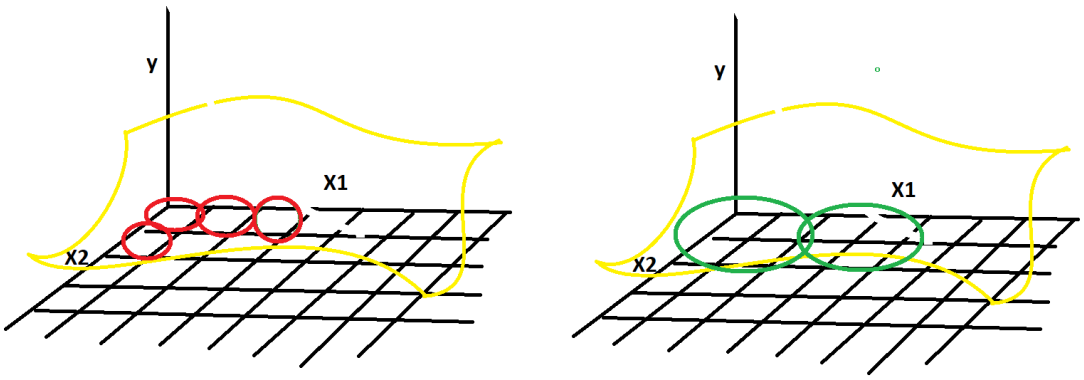

Figure 2: Abordarea rezolvarii cu RBF 1

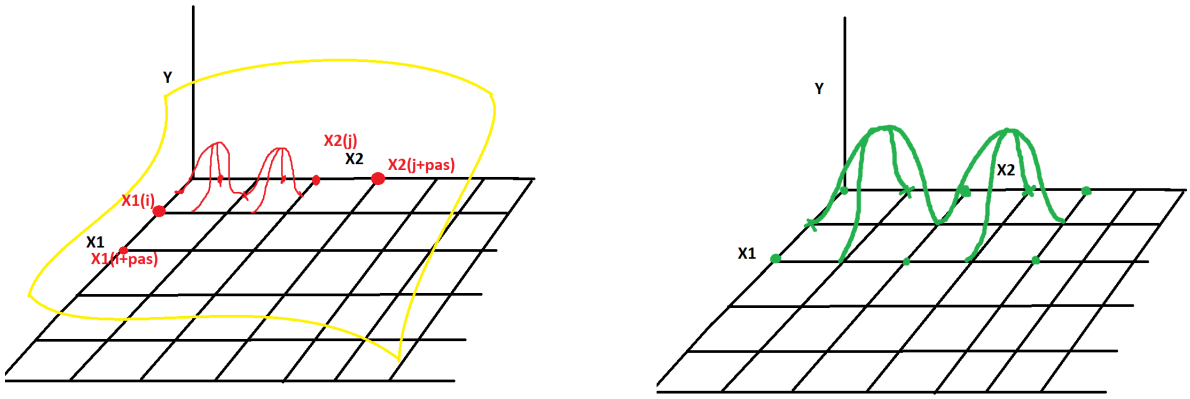

Figure 3: Abordarea rezolvarii cu RBF 2

3 Media erorilor pătratice și grafice reprezentative.

3.1 Discuție pentru media erorilor pătratice.

m	MSE identificare	MSE validare
1	57.3269	56.6524
2	57.3261	57.3261
3	47.0537	45.5769
4	47.0520	45.5769
5	11.9475	11.8810
6	11.9436	11.8851
7	6.5276	6.6247
8	6.5215	6.6325
9	9.5692	9.6489
10	11.0592	11.0991
11	9.0947	9.1647
12	22.3910	22.2659
13	12.7966	12.7399
14	40.1938	39.8625
15	20.7740	20.6547
16	61.1098	60.8516
17	33.0205	32.8518
18	82.9715	82.7824
19	50.3838	50.3005
20	104.6981	104.5875
21	125.1533	125.1129
22	125.2077	125.1672
23	144.1136	144.1772
24	144.1122	144.1812
25	161.4234	161.6113
26	161.4902	161.6769
27	177.1429	177.4822
28	177.1855	177.5373
29	191.3672	191.8731
30	191.4670	191.9860
31	204.2778	204.9708
32	204.3968	205.1230
33	215.9855	216.9208
34	216.1279	217.0740
35	226.5999	227.7889
36	226.7653	227.9667
37	236.2672	237.7425
38	236.4417	237.9239
39	245.0591	246.8493
40	245.2497	247.0544

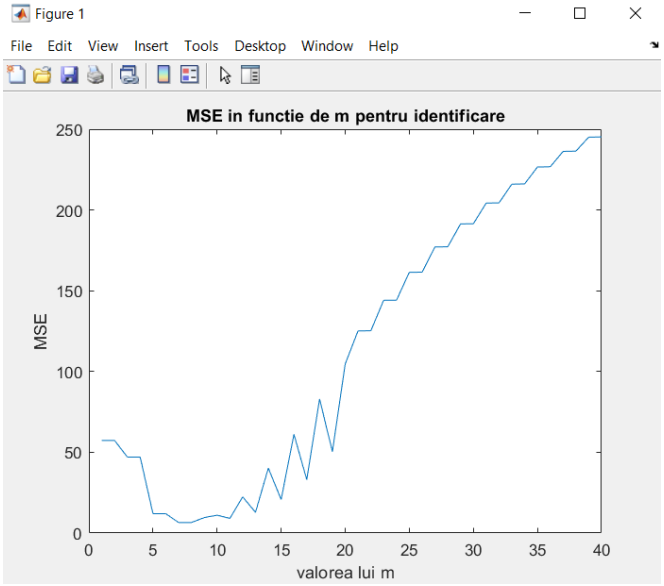

Figure 4: MSE în funcție de m pentru identificare

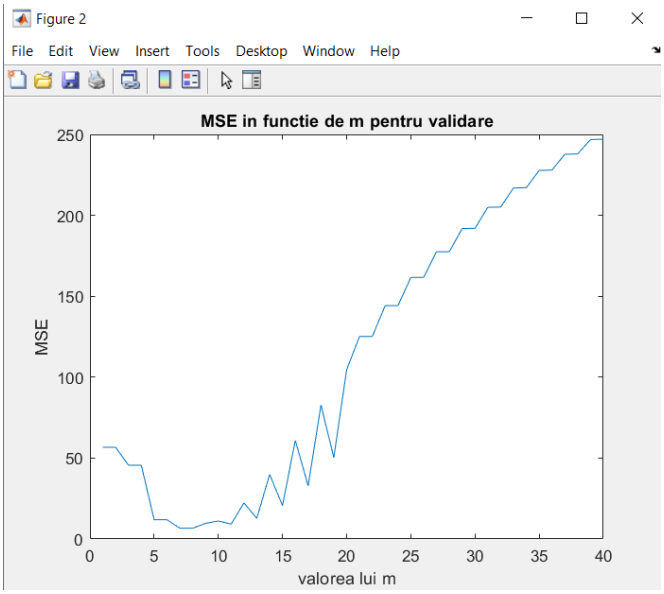

Figure 5: MSE în funcție de m pentru validare

Putem observa atât din grafice cât și din tabel că eroarea medie pătratică minimă este obținută pentru un aproximator polinomial de grad 7. De asemenea trebuie remarcat că începând de la gradul 9 apare fenomenul de supraantrenare ce duce la o creștere rapidă a erorii medii pătratice.

Număr funcții RBF	MSE
16	233.00
16	139.13
16	164.91
25	196.77
25	222.71
36	27.25
36	109.70
49	17.04
64	9.93
100	81.60
121	12.55
196	9.10
324	41.36
676	8.34
1849	8.11

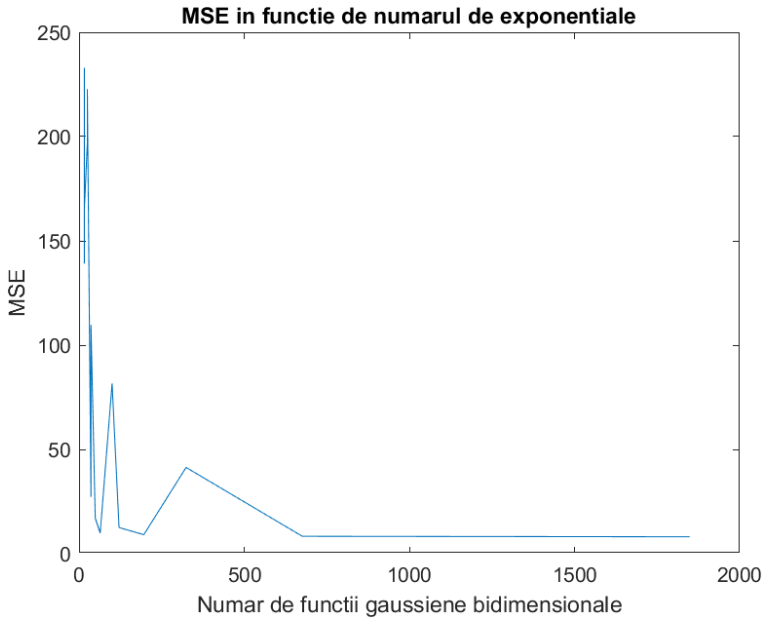

Figure 6: Reprezentarea MSE

Se poate remarca din tabel că pentru un același număr de funcții de bază (de exemplu 16) obtinem erori diferite. Acest lucru se întamplă din cauză că intervalul pe care obtinem parametrii c si b pentru regresori difera, ceea ce duce la forme diferite ale exponențialelor bidimensionale și implicit și la erori diferite.

3.2 Grafice reprezentative pentru aproximare.

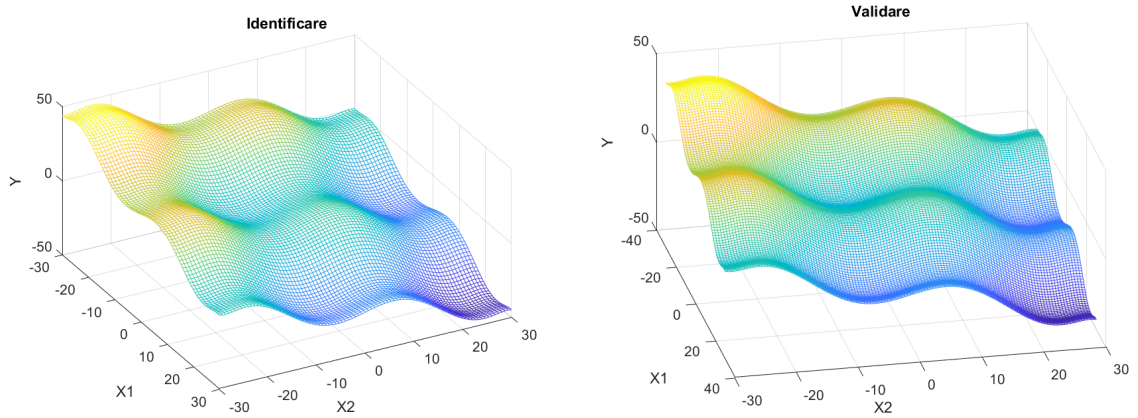

Figure 7: Graficele aproximarilor obtinute pentru m=7

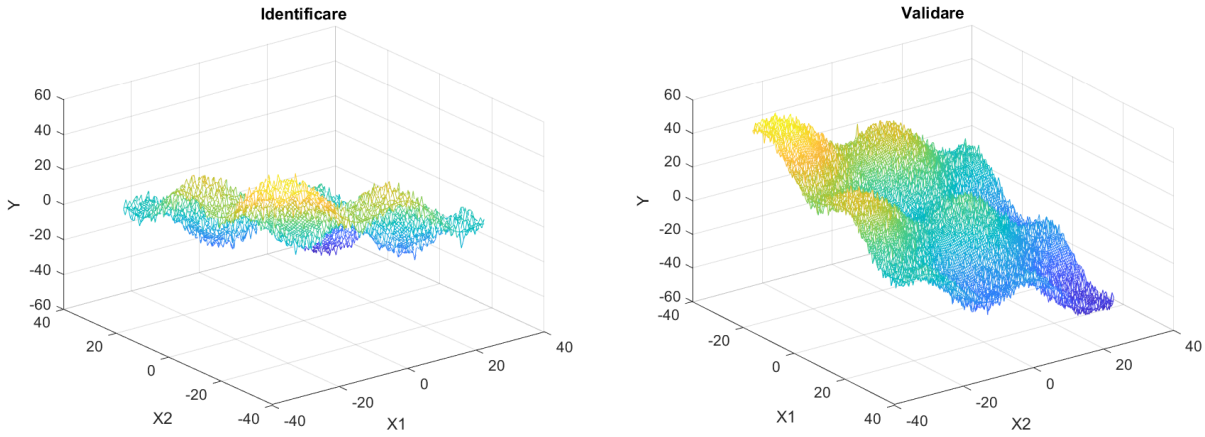

Figure 8: Suprapunerea aproximarilor peste ieșirea inițială(m=7)

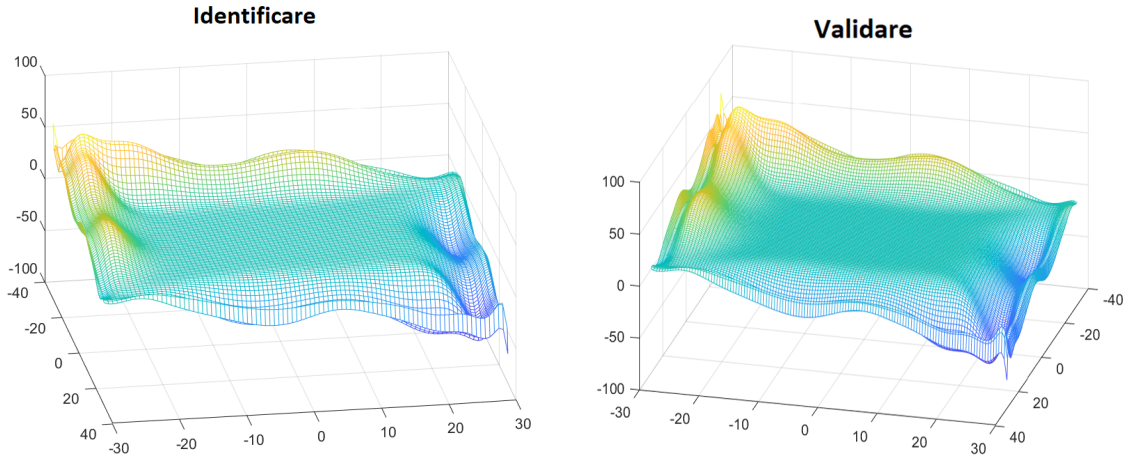

Figure 9: Fenomenul de supraantrenare pentru m=20 - fără suprapunere

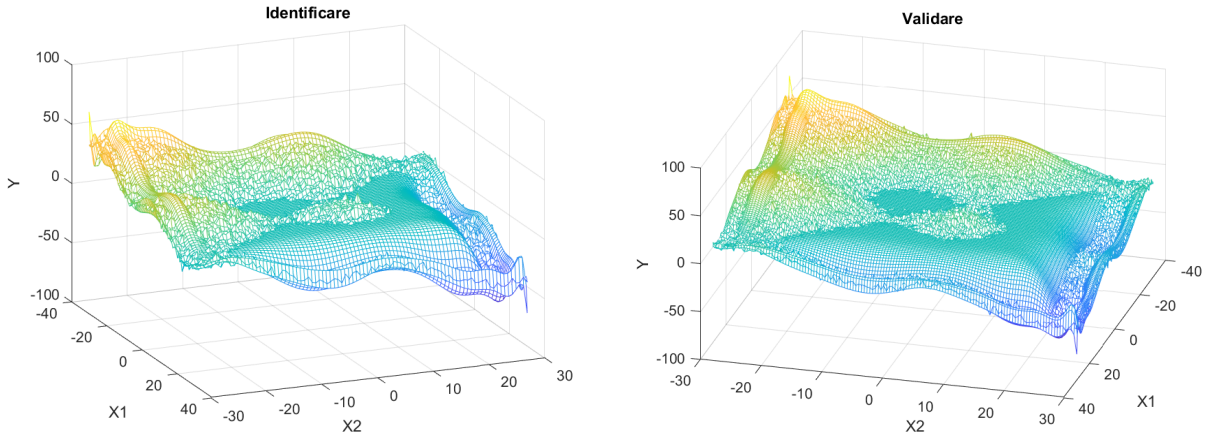

Figure 10: Fenomenul de supraantrenare pentru m=20 - cu suprapunere peste ieșirea inițială

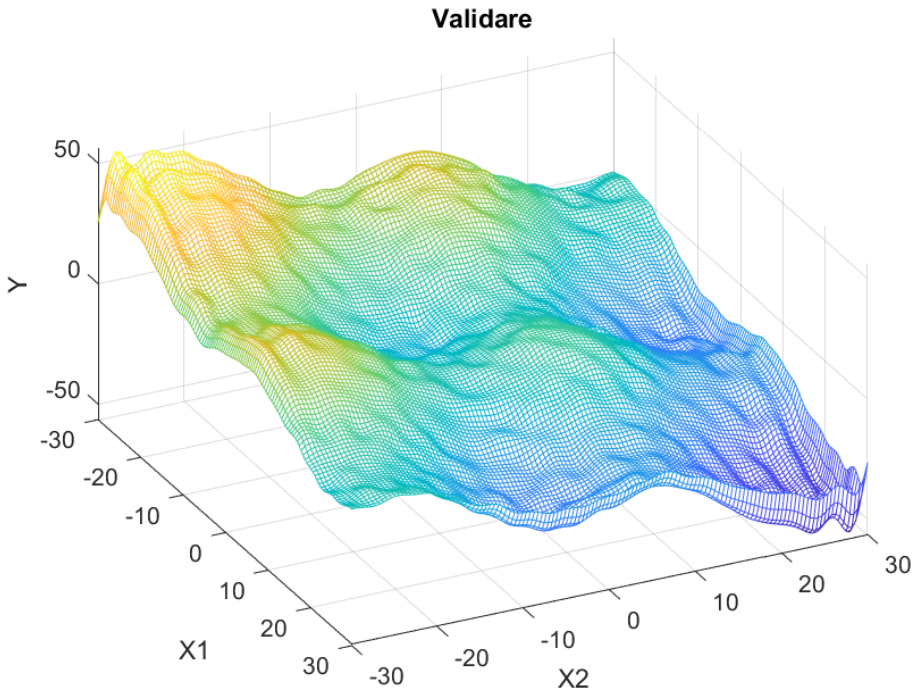

Figure 11: Aproximarea RBF - fără suprapunere

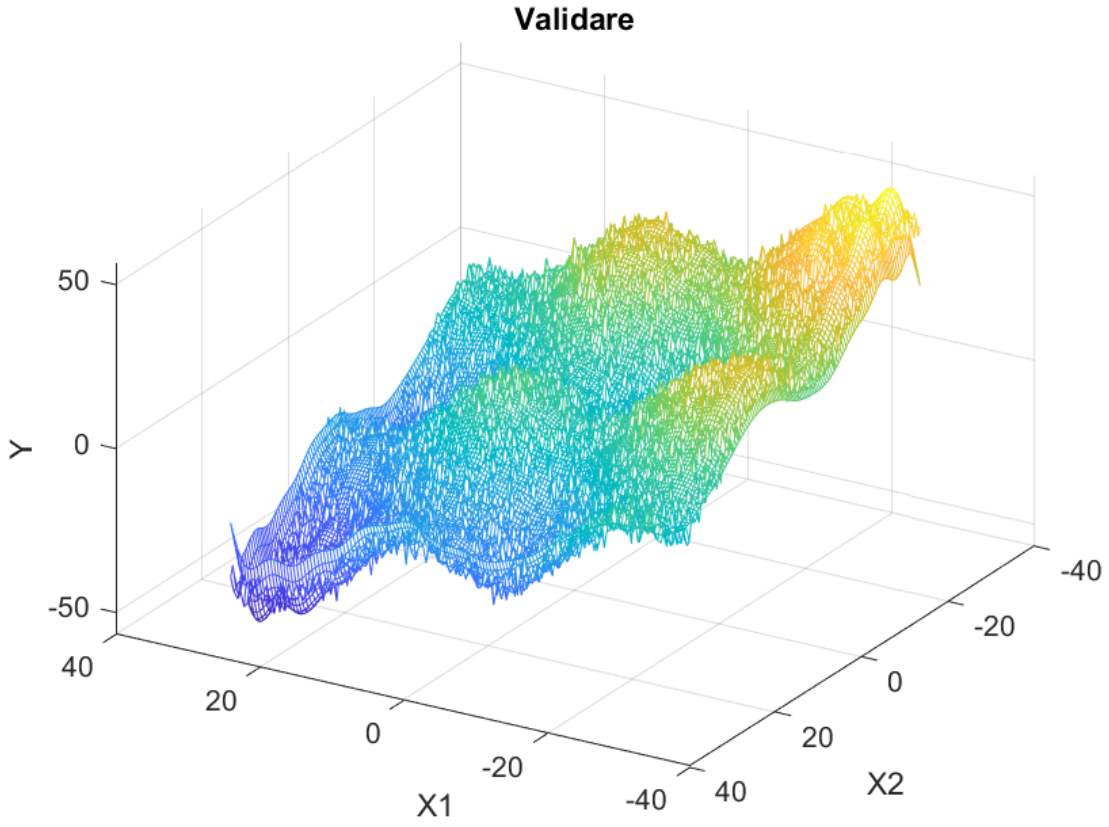

Figure 12: Aproximarea RBF - cu suprapunere

4 Comparație între algoritmi

Mai jos avem compararea timpilor de execuție și a erorilor medii pătratice între cei doi algoritmi de regresie polinomiala, respectiv între regresia polinomială și aproximarea cu funcții de bază gaussiene.

Între cei doi algoritmi de regresie ce folosesc polinoame nu există diferențe legate de erorile medii pătratice, deoarece principiile din spatele lor sunt identice, diferită fiind doar implementarea ce duce de asemenea la diferențe legate de timpii de execuție ale celor doua abordări.

De asemenea, diferențele între primul algoritm și cel care se folosește de funcțiile de bază Gaussiene sunt atât în timpul de execuție cât și în erorile medii pătratice obținute.

m	timp de executie functia aproximare	timp de executie funcția regresie
1	0.33 secunde	0.23 secunde
2	0.68 secunde	0.15 secunde
3	3.38 secunde	0.24 secunde
4	7.72 secunde	0.35 secunde
5	13.31 secunde	0.50 secunde
6	19.70 secunde	0.61 secunde
7	26.98 secunde	0.94 secunde
8	34.83 secunde	1.19 secunde
9	43.60 secunde	1.42 secunde
10	54.46 secunde	1.67 secunde
11	72.72 secunde	2.28 secunde
12	86.28 secunde	2.86 secunde
13	89.14 secunde	4.71 secunde
14	102.44 secunde	7.54 secunde
15	132.13 secunde	8.03 secunde
16	150.23 secunde	9.57 secunde
17	170.39 secunde	9.84 secunde
18	185.25 secunde	11.94 secunde
19	201.30 secunde	12.36 secunde
20	201.31 secunde	13.82 secunde

Număr funcții RBF	Timp de execuție
16	0.322 secunde
16	0.224 secunde
16	0.215 secunde
25	0.218 secunde
25	0.299 secunde
36	0.408 secunde
36	0.377 secunde
49	0.592 secunde
64	0.895 secunde
100	0.856 secunde
121	1.505 secunde
196	2.243 secunde
324	3.245 secunde
676	7.858 secunde
1849	22.123 secunde

5 Concluzii

-Identificarea prin regresie liniară este o metodă simplă și utilă în cazul modelelor de tip cutie neagră, putând să ajungem la un model parametric unde numărul de parametri îl putem regla în funcție de eroarea obținută prin aproximare (vom păstra mereu parametrii care oferă cea mai mică eroare la aproximare);

-Indiferent de algoritmul folosit pentru determinarea regresorilor, parametrii optimi vor trebui să fie de fiecare dată la fel, respectiv și erorile corespunzătoare obținute de acei algoritmi vor trebui sa fie la fel;

-Tipul regresorilor aleși influențează calitatea aproximării și de aceea trebuie să îi considerăm cu atenție înainte de a face o aproximare și să verificăm ce fel de regresori s-ar preta cel mai bine;

-Chiar dacă erorile introduse de fiecare funcție de bază radială sunt locale (și teoretic pot fi minimizate prin alegerea parametrilor c si b potriviți), la finalul aproximării sunt cazuri în care ieșirea nu poate fi aproximată la fel de bine că atunci când am folosi un aproximator polinomial.

6 Anexă

6.1 Funcția "regresie"

```
function [theta,y_aprox_id,y_aprox_val] = regresie(data,val,m)
tic

%in matricea regresorilor vom avea toate perechile de forma
%X{1}^p1*X{2}^p2, cu p1+p2<=m
p1=m;%p1 va fi puterea la care il vom calcula pe X{1}
p2=0;%p2 este puterea la care il vom ridica pe X{2}
S=zeros(length(data.X{1})*length(data.X{2}),(m+1)*(m+2)/2);%alocam
    spatiu petru matricea "regresorilor"

%k va reprezenta indicele coloanei matricei S=Sigma de regresori,
    coloana
%maxima fiind (m+1)*(m+2)/2
for k=1:(m+1)*(m+2)/2
    ct=1;%contorul ct va itera pe liniile matricei S pana la N1*N2, N1=
        lungimea lui X{1}, iar N2=lungimea lui X{2}

    %iteram prin cele doua matrice pentru a obtine toate combinatiile
    %(X{1}(i),X{2}(j)).
    %Aceste doua for-uri vor calcula practic cate o coloana din matricea
        S
    %corespunzatoare regresorului X{1}^p1*X{2}^p2, cu p1+p2<=m
    for i=1:length(data.X{1})
        for j=1:length(data.X{2})
            b=data.X{1}(i)^p1.*data.X{2}(j)^p2;
            S(ct,k)=b;
            ct=ct+1;
        end
    end
    %Aici redeterminam forma regresorului, pe care il calculam dupa
    %obtinerea fiecarei coloane.
    %p1 va merge de la m la 0, si va fi decrementat dupa obtinerea
        fiecarei
    %coloane, in timp ce p2 va fi incrementat cu cate o unitate atunci
        cand
    %s-au obtinut toate combinatiile X{1}^p1*X{2}^p2, cu p2 fixat.
    %Astfel, regresorii determinati vor fi in urmatoarea ordine in
        matricea
    %simbolica Sigma:
    %S=[x1^m, x1^(m-1),..., x1, 1, x1^(m-1)*x2, x1^(m-2)*x2,..., x1^2*x2
        ,
    %   x1*x2, x2,.....,x1*x2^(m-1), x2^(m-1), x2^m].
    p1=p1-1;
    if p1==-1
```

```
37          p2=p2+1;
38          p1=m-p2;
39      end
40  end
41  %folosim operatorul \ pentru a rezolva sistemul de ecuatii
        supradeterminat
42  %cu necunoscutele theta(i), cu i de la 1 la (m+1)(m+2)/2, cu
        specificatia
43  %ca matricea patratica Y ce contine datele de iesire trebuie sa fie
44  %transformata intr-o matrice coloana: Y->Y(:).
45  theta=S\data.Y(:);
46  y_aprox_id=S*theta;%reconstruim matricea ce ne aproximeaza iesirea Y
        folosind regresorii
47                  %deja calculati si matricea S de parametri, obtinuta mai
                        sus
48  y_aprox_id=reshape(y_aprox_id,length(data.X{1}),length(data.X{2}));%
        redimensionam matricea
49  %ce aproximeaza iesirea Y, transformand-o dintr-o matrice coloana intr-o
         matrice patratica
50  %pentru a putea sa o reprezentam grafic sub forma urmatoare: fiecarei
51  %intrari de tipul (X{1}(i),X{2}(j)) ii corespunde o iesire y_aprox(i,j)
52
53  %algoritmul de mai jos foloseste matricea de parametri theta pentru a
54  %aproxima setul de date de validare, nefiind diferit cu ceva fata de
        partea
55  %descrisa mai sus.
56      p1=m;
57      p2=0;
58      S=zeros(length(val.X{1})*length(val.X{2}),(m+1)*(m+2)/2);
59      for k=1:(m+1)*(m+2)/2
60          ct=1;
61          for i=1:length(val.X{1})
62              for j=1:length(val.X{2})
63              b=val.X{1}(i)^p1.*val.X{2}(j)^p2;
64              S(ct,k)=b;
65              ct=ct+1;
66              end
67          end
68          p1=p1-1;
69          if p1==-1
70              p2=p2+1;
71              p1=m-p2;
72          end
73      end
74          y_aprox_val=S*theta;
75          y_aprox_val=reshape(y_aprox_val,length(val.X{1}),length(val.X{2})
                );
76  toc
```

6.2 Script-ul care apeleaza funcția "regresie"

```
1
2  clear
3  load('proj_fit_08.mat')
4  MSE=[]
5  for i=1:40
6  [theta,y_aprox_id,y_aprox_val]=regresie(id,val,i);
7  MSE=[MSE, mean(mean((y_aprox_val-val.Y).^2))];
8  end
9  plot(1:40,MSE);
10 [eroare_min,m_min]=min(MSE)
11 [theta,y_aprox_id,y_aprox_val]=regresie(id,val,m_min);
12 figure(1)
13 mesh(id.X{1},id.X{2},id.Y),hold on;
14 mesh(id.X{1},id.X{2},y_aprox_id),title('Identificare'),xlabel('X1'),
       ylabel('X2'),zlabel('Y');
15
16 figure(2)
17 mesh(val.X{1},val.X{2},val.Y),hold on;
18 mesh(val.X{1},val.X{2},y_aprox_val),title('Validare'),xlabel('X1'),
       ylabel('X2'),zlabel('Y');
19
20 [theta,y_aprox_id,y_aprox_val]=regresie(id,val,20);
21 figure(3)
22 mesh(id.X{1},id.X{2},id.Y),hold on;
23 mesh(id.X{1},id.X{2},y_aprox_id),title('Identificare'),xlabel('X1'),
       ylabel('X2'),zlabel('Y');
24
25 figure(4)
26 mesh(val.X{1},val.X{2},val.Y),hold on;
27 mesh(val.X{1},val.X{2},y_aprox_val),title('Validare'),xlabel('X1'),
       ylabel('X2'),zlabel('Y');
```

6.3 Funcția "aproximare"

```
1  function [Theta,y_aproximat_id,y_aproximat_val] = aproximator(date_id,
       date_val,m)
2      tic
3      Regresori=[]; % In variabila Regresori,
4      %vom retine matricea cu regresori
5
6
7      for i=1:length(date_id.X{1})
8          % Este indexul curent pentru valorile
9          % din intrarea X1
10         for j=1:length(date_id.X{2})
11             % Este indexul curent pentru valorile
12             % din intrarea X2
```

```
13
14              %astfel se vor parcurge pe rand combinatiie
15              % x1(1),x2(1)...
16              % x1(1),x2(ultimul_index)...
17              % ....x1(ultimul_index),x2(ultimul_index)
18
19               linie = [1];
20               % Este o varibabila in care vom adauga
21               % toti termenii ce ar trebui sa existe pe o linie in
                     matricea
22               % de regresori, primul element din linie intotdeauna
23               % va fi 1
24
25                    for h=1:m
26                        % Se vor adauga toate puterile
27                        % lui X1, respectiv X2, pana la gradul m,
28                        % in variabila "linie"
29                        elem1 = date_id.X{1}(i)^h;
30                        elem2 = date_id.X{2}(j)^h;
31                        linie = [linie,elem1,elem2];
32                    end
33
34
35                    for putere_x1 =1 : (m-1)
36                        for putere_x2 = 1 :(m-putere_x1)
37                            % Se vor adauga toate
38                            % combinatiile de puteri, x1,x2,
39                            % astfel incat sa se respecte conditia,
40                            % putere_x1 + putere_x2 sa fie mai mic
41                            % sau egal cu m
42                            elem = (date_id.X{1}(i)^putere_x1) * (date_id.X
                                  {2}(j)^putere_x2);
43                            linie = [linie,elem];
44                        end
45                    end
46               % Linia de regresori a fost completata cu toate
                     combinatiile
47               % posibile, urmeaza sa o adaugam in matricea corespondenta.
48                Regresori = [Regresori;linie];
49
50          end
51      end
52
53      %Calculare Theta
54      Theta = Regresori\date_id.Y(:);
55
56      % Calculare Y aproximat pentru identificare
57      y_aproximat_id = Regresori*Theta;
```

```
58
59      % Reconstruim y astfel incat sa fie bidimensional, ca
60      % sa poata fi reprezentat in acelasi format, precum
61      % iesirea data
62      y_aproximat_id = reshape(y_aproximat_id,length(date_id.X{1}),length(
            date_id.X{2}));
63
64
65
66      Regresori=[]; % In variabila Regresori,
67      %vom retine matricea cu regresori
68      %de data aceasta pentru validare
69
70
71      for i=1:length(date_val.X{1})
72          % Este indexul curent pentru valorile
73          % din intrarea X1
74          for j=1:length(date_val.X{2})
75              % Este indexul curent pentru valorile
76              % din intrarea X2
77
78             %astfel se vor parcurge pe rand combinatiie
79             % x1(1),x2(1)...
80             % x1(1),x2(ultimul_index)...
81             % ....x1(ultimul_index),x2(ultimul_index)
82
83              linie = [1];
84              % Este o varibabila in care vom adauga
85              % toti termenii ce ar trebui sa existe pe o linie in
                    matricea
86              % de regresori, primul element din linie intotdeauna
87              % va fi 1
88
89                  for h=1:m
90                      % Se vor adauga toate puterile
91                      % lui X1, respectiv X2, pana la gradul m,
92                      % in variabila "linie"
93                      elem1 = date_val.X{1}(i)^h;
94                      elem2 = date_val.X{2}(j)^h;
95                      linie = [linie,elem1,elem2];
96                  end
97
98
99                  for putere_x1 =1 : (m-1)
100                     for putere_x2 = 1 :(m-putere_x1)
101                         % Se vor adauga toate
102                         % combinatiile de puteri, x1,x2,
103                         % astfel incat sa se respecte conditia,
```

```
104                             % putere_x1 + putere_x2 sa fie mai mic
105                             % sau egal cu m
106                             elem = (date_val.X{1}(i)^putere_x1) * (date_val.
                                    X{2}(j)^putere_x2);
107                             linie = [linie,elem];
108                         end
109                     end
110                 % Linia de regresori a fost completata cu toate
                        combinatiile
111                 % posibile, urmeaza sa o adaugam in matricea corespondenta.
112                 Regresori = [Regresori;linie];
113 
114             end
115         end
116 
117         % Calculare Y aproximat pentru validare
118         y_aproximat_val = Regresori*Theta;
119 
120         % Reconstruim y astfel incat sa fie bidimensional, ca
121         % sa poata fi reprezentat in acelasi format, precum
122         % iesirea data
123         y_aproximat_val = reshape(y_aproximat_val,length(date_val.X{1}),
                length(date_val.X{2}));
124         toc
125 end
```

6.4 Script-ul care apeleaza funcția ”aproximator”

```
1  clc
2  clear
3  tic
4  load('proj_fit_08.mat')
5 
6  erori =[];
7  erori_val = [];
8  k=1;
9   for m = 1:20;
10          [Theta,y_aprox_id,y_aprox_val] = aproximator(id,val,m);
11          MSE_id = mean(mean((id.Y - y_aprox_id).^2));
12          MSE_val = mean(mean((val.Y - y_aprox_val).^2));
13          figure(k);
14          mesh(id.X{1},id.X{2},id.Y),title(strcat('Identificare m=',
                num2str(m),' ',' MSE=',num2str(MSE_id)));
15          xlabel('X1'),ylabel('X2'),zlabel('Y'),hold on;
16 
17          figure(k);
18          mesh(id.X{1},id.X{2},y_aprox_id),title(strcat('Identificare m=',
                num2str(m),' ',' MSE=',num2str(MSE_id)));
```

```
19          xlabel('X1'),ylabel('X2'),zlabel('Y');
20
21          k=k+1;
22          figure(k)
23          mesh(val.X{1},val.X{2},val.Y),title(strcat('Validare m=',num2str
                (m),' ',' MSE=',num2str(MSE_val)));
24          xlabel('X1'),ylabel('X2'),zlabel('Y'),hold on;
25
26          figure(k);
27          mesh(val.X{1},val.X{2},y_aprox_val),title(strcat('Validare m=',
                num2str(m),' ',' MSE=',num2str(MSE_val)));
28          xlabel('X1'),ylabel('X2'),zlabel('Y');
29
30          MSE_id = mean(mean((id.Y - y_aprox_id).^2));
31          MSE_val = mean(mean((val.Y - y_aprox_val).^2));
32          erori = [erori;
33                  MSE_id];
34          erori_val = [erori_val;
35                  MSE_val];
36   end
37
38  toc
39
40  figure;
41  plot(erori);
42  title('MSE in functie de m pentru identificare');
43  xlabel('valorea lui m');
44  ylabel('MSE');
45
46  figure;
47  plot(erori_val);
48  title('MSE in functie de m pentru validare');
49  xlabel('valorea lui m');
50  ylabel('MSE');
```

6.5 Regresie cu funcții de bază Gaussiene

```
1   clear
2   load('proj_fit_08.mat');
3   MSE=[];
4   %crestem numarul de exponentiale cu fiecare iteratie prin acest for
5   %cu cat pasul este mai mare, cu atat intervalul pe care se face
        aproximarea
6   %este mai mare si deci numarul de exponentiale considerat este mai mic.
7   for pas=31:-2:2
8   tic
9   k=1;
10
```

```
%pentru fiecare sectiune cuprinsa intre X1(i)-X1(i+pas) si X2(j)-X2(j+pas)
%cautam sa calculam valorile c si b ce aproximeaza prin exponentiale acele sectiuni
%considerate
for i=1:pas:length(id.X{1})
    for j=1:pas:length(id.X{2})
        %verificam daca nu cumva am depasit lungimea datelor de intrare
          if (i+pas<=length(id.X{1}) & j+pas<=length(id.X{2}))
              %coeficientii c ai exponentialei i-am considerat ca fiind
              %mediile capetelor intervalelor de pe axele X1 si X2. c(i,1)
              %si c(i,2) sunt la mijlocul acestor intervale.
            c(k,1)=mean([id.X{1}(i),id.X{1}(i+pas)]);
            c(k,2)=mean([id.X{2}(j),id.X{2}(j+pas)]);
            %coeficientii b sunt latimile pe axele X1 si X2 ale exponentialelor
            %ce aproximeaza iesirea Y
            b(k,1)=abs(id.X{1}(i)-id.X{1}(i+pas));
            b(k,2)=abs(id.X{2}(j)-id.X{2}(j+pas));
            k=k+1;
          end
    end
end

%identificare
%calculam valorile regresorilor pentru fiecare pereche X1(i),X2(j).
k=1;
nr=length(b(:,1));
for i=1:length(id.X{1})
    for j=1:length(id.X{2})
        for n=1:nr
            S(k,n)=exp(-(id.X{1}(i)-c(n,1))^2/b(n,1)^2)*exp(-(id.X{2}(j)-c(n,2))^2/b(n,2)^2);
        end
        k=k+1;
    end
end

T=S\id.Y(:);
y_aprox=S*T;
y_aprox=reshape(y_aprox,length(id.X{1}),length(id.X{2}));

mesh(id.X{1},id.X{2},y_aprox),title('Identificare'),xlabel('X1'),ylabel('X2'),zlabel('Y');

%validare
k=1;
```

```
for i=1:length(val.X{1})
    for j=1:length(val.X{2})
        for n=1:nr
            S2(k,n)=exp(-(val.X{1}(i)-c(n,1))^2/b(n,1)^2)*exp(-(val.X
                {2}(j)-c(n,2))^2/b(n,2)^2);
        end
        k=k+1;
    end
end
y_aprox_val=S2*T;
y_aprox_val=reshape(y_aprox_val,length(val.X{1}),length(val.X{2}));
MSE=[MSE,mean( mean( (y_aprox_val-val.Y).^2) )];

end
pas=31:-2:2;
dim=size(val.Y);
figure(2)
%Pentru a afla cate RBF-uri calculam cate functii radiale avem de-a
    lungul
%axei X1, cate functii de baza radiale avem de-a lungul lui X2,
%preluam partea intreaga a celor doua impartiri si inmultim
%cele doua rezultate obtinute.
plot(floor((dim(1)./pas)).*floor((dim(2)./pas)),MSE),xlabel('Numar de
    functii gaussiene bidimensionale'),ylabel('MSE'),title('MSE in
    functie de numarul de exponentiale');
figure(1)
mesh(val.X{1},val.X{2},val.Y),xlabel('x'),ylabel('y'),zlabel('z'),hold
    on,
mesh(val.X{1},val.X{2},y_aprox_val),title('Validare'),xlabel('X1'),
    ylabel('X2'),zlabel('Y');
```