PROIECT PARTEA I

PIDX: **11/11**

Abrudan Lavinia-Maria
Malutan Alexandra

Introducere

Modelarea comportamentului unei functii necunoscute:

- Programarea unui aproximator polinomial de grad configurabil.
- Antrenarea aproximatoarelor de diverse grade in vederea obtinerii celui mai precis.
- Raportarea erorii mediei patratice pe ambele seturi de date.
- Reprezentarea grafica a rezultatului dat de aproximator comparat cu valorile reale ale functiei, pe cele doua seturi de date.

Rationamentul de determinare al matricei phi

Pentru m=3:

$$\phi = \begin{bmatrix} 1 & x_1(1) & x_2(1) & x_1^2(1) & x_2^2(1) & x_1(1) * x_2(1) \\ 1 & x_1(1) & x_2(2) & x_1^2(1) & x_2^2(2) & x_1(1) * x_2(2) \\ 1 & x_1(1) & x_2(3) & x_1^2(1) & x_2^2(3) & x_1(1) * x_2(3) \\ \ldots & & \ldots & & & \ldots \end{bmatrix}$$

- Pornind de la numarul de termeni pe care ii genereaza gradul polinomului, am cautat o formula care sa ne usureze creearea coloanelor pentru matricea phi.

m=1 => $1 + x_1 + x_2$

=> 3 termeni

m=2 => $1 + x_1 + x_2 + x_1^1 + x_2^2 + x_{1^*} x_2$

=> 6 termeni

m=3 => $1 + x_1 + x_2 + x_1^2 + x_2^2 + x_1^3 + x_2^3 + x_1 * x_2 + x_1^2 * x_2 + x_1 * x_2^2$

=> 10 termeni

m=4 => $1 + x_1 + x_2 + x_1^2 + x_2^2 + x_1^3 + x_2^3 + x_1^4 + x_2^4 + x_{1^*} x_2 + x_1^2 * x_2 + x_1 * x_2^2 + x_1^3 * x_2 + x_1 * x_2^3 + x_1^2 * x_2^2$

=> 15 termeni

m=5 => $1 + x_1 + x_2 + x_1^2 + x_2^2 + x_1^3 + x_2^3 + x_1^4 + x_2^4 + x_1^5 + x_2^5 + x_1 * x_2 + x_1^2 * x_2 + x_1 * x_2^2 + x_1^3 * x_2 + x_1 * x_2^3 + x_1^4 * x_2 + x_1 * x_2^4 + x_1^2 * x_2^2 + x_1^3 * x_2^2 + x_1^2 x_2^3$

=> 21 termini

...

- Am pornit de la suma primelor n numere naturale :

$$\sum_{1}^{n} k = \frac{n(n+1)}{2}$$

- Numarul total de termeni de gradul m include termenul constant 1 o singura data, prin urmare numarul de termini pentru fiecare grad l-am considerat ca fiind suma de termeni pana la gradul m+1

$$\sum_{k=1}^{m+1} k = \frac{(m+1)(m+2)}{2}$$

Crearea functiei ϕ

- Prima data am luat dimensiunile vectorilor x{1}, x{2} din setul de date;
- Am creat matricea ϕ cu ajutorul a 4 bucle for;
- Doua bucle pentru parcurgerea pozitiilor lui x1 si x2 in matrice;
- Doua bucle pentru creearea puterilor acestor doi vectori;

Creearea vectorului ϴ

- Am observat ca datele de identificare pentru matricea y are dimensiunea 51 x 51;
- Pentru a creea un vector coloana care sa contina toate elementele lui y am aplicat functia din matlab reshape, care concateneaza toate liniile intr-o coloana, obtinand o dimensiune de 2601 x 1;
- Am facut acest lucru pentru a creea vectorul ϴ folosind regresia liniara.

$$\Theta=\frac{\phi}{Y}$$

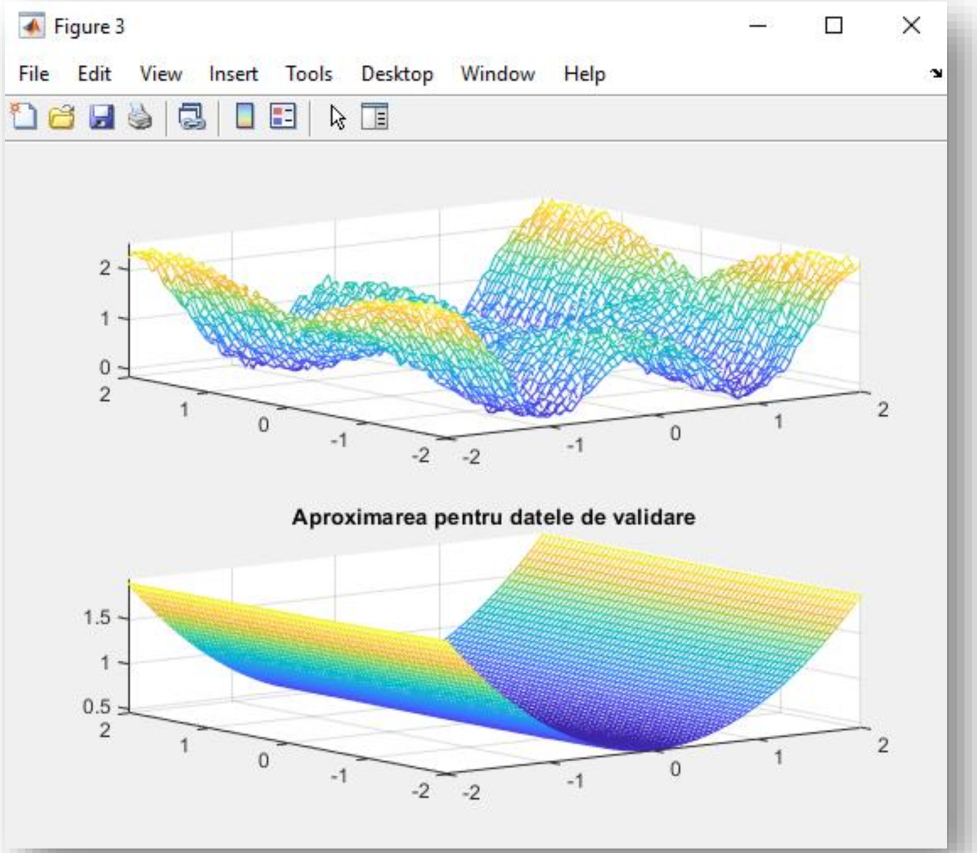
Structura aproximatorului

$$\hat{g} = \phi * \theta$$

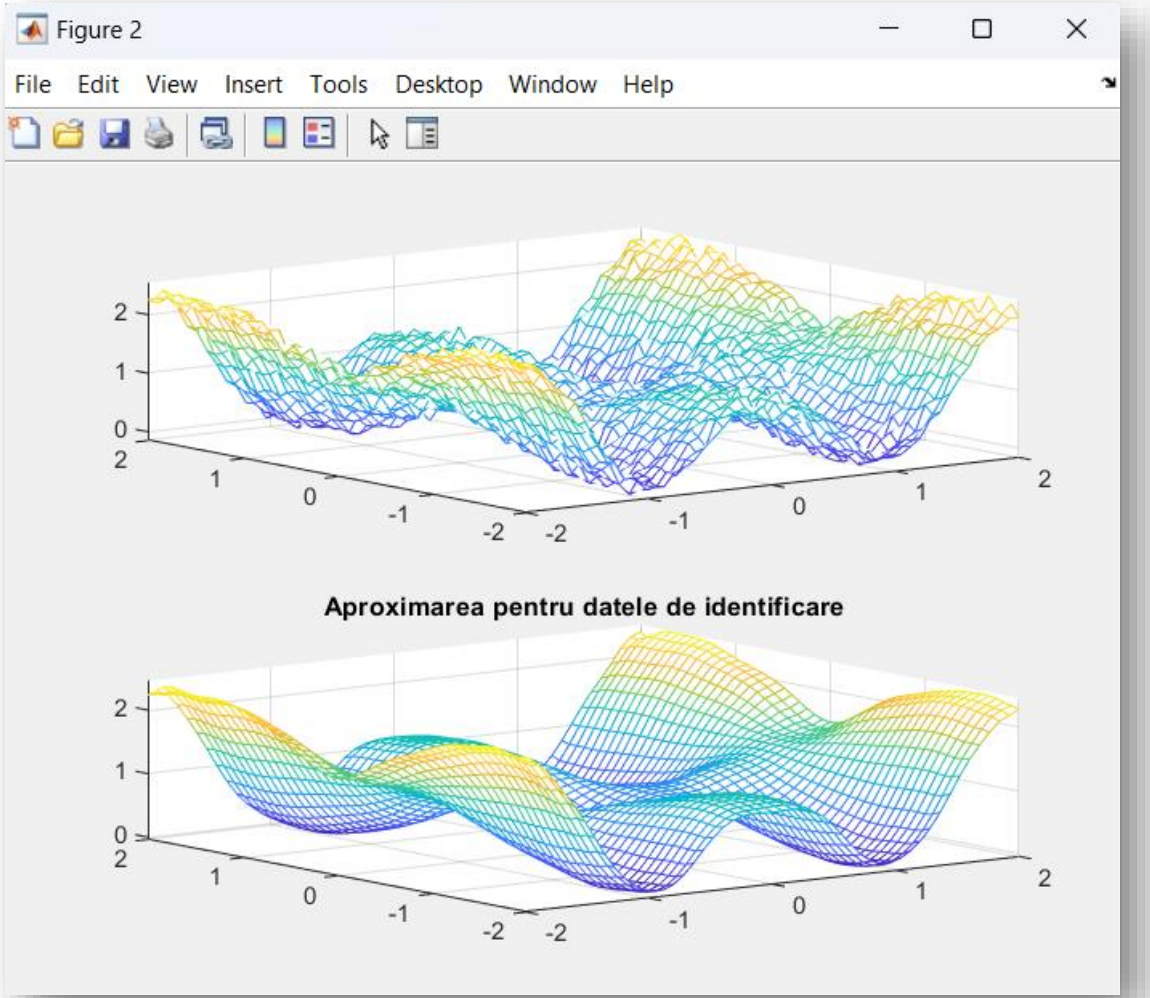
Acesta a fost realizat pe primul set de date pentru antrenarea modelului, de asemenea si pe al doilea set de date pentru validare.

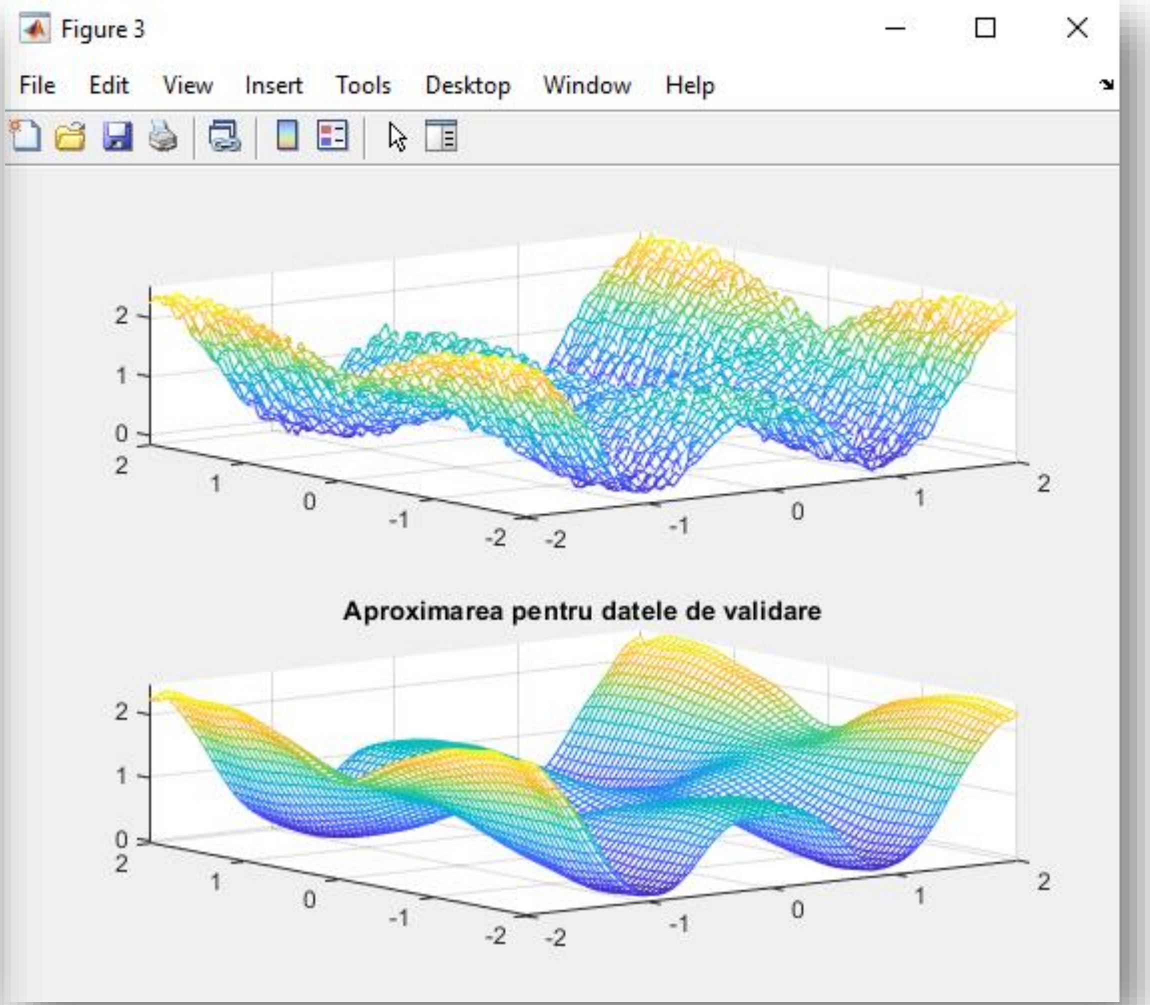
Determinarea celui mai bun grad m

- Am observant ca pentru gradul m=3, modelul determinat nu aproximeaza setul de date precis.



- Pentru gradul m=18, m=19 modelul aproximeaza cel mai corect, avand eroarea cea mai mica.

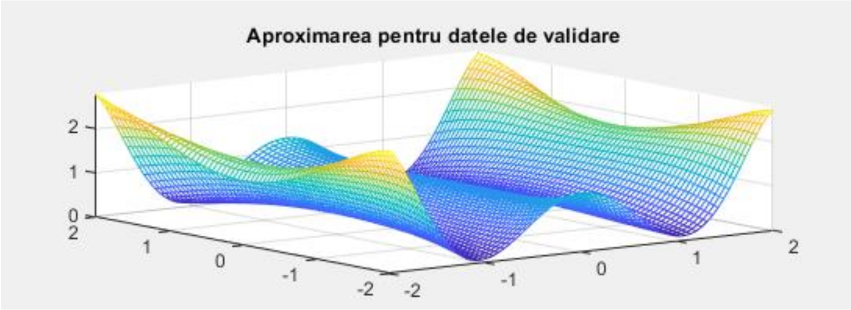




Pentru m=7;

Eroarea medie patratica pe datele de identificare: 0.0354

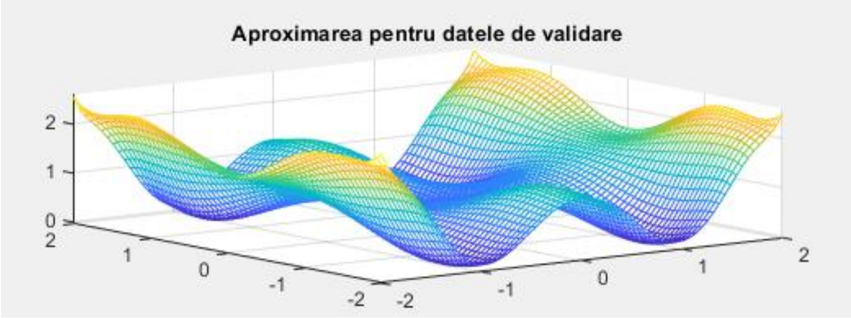
Eroarea medie patratica pe datele de validare: 0.0344



Pentru m=10;

Eroarea medie patratica pe datele de identificare: 0.0088

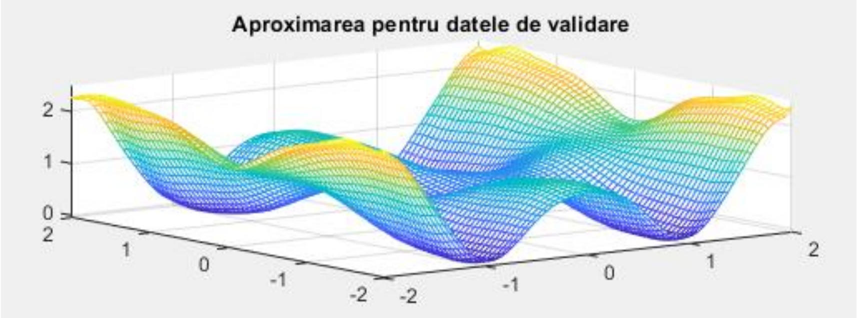
Eroarea medie patratica pe datele de validare: 0.0087



Pentru m=14;

Eroarea medie patratica pe datele de identificare: 0.0058

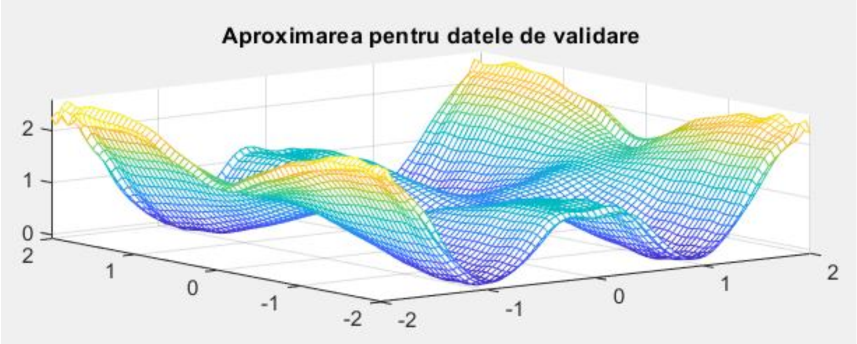
Eroarea medie patratica pe datele de validare: 0.0059



Pentru m=25;

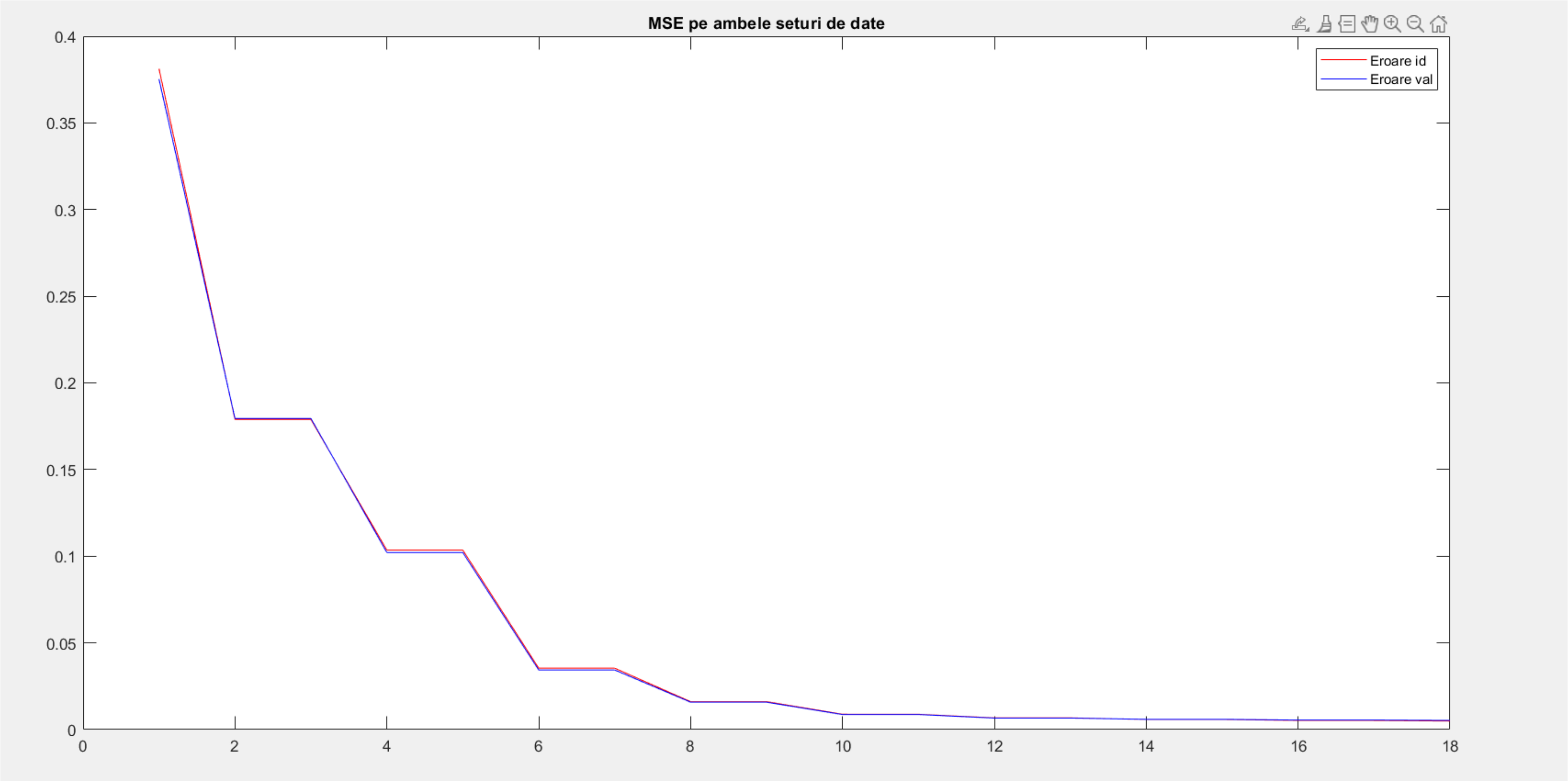
Eroarea medie patratica pe datele de identificare: 0.0041

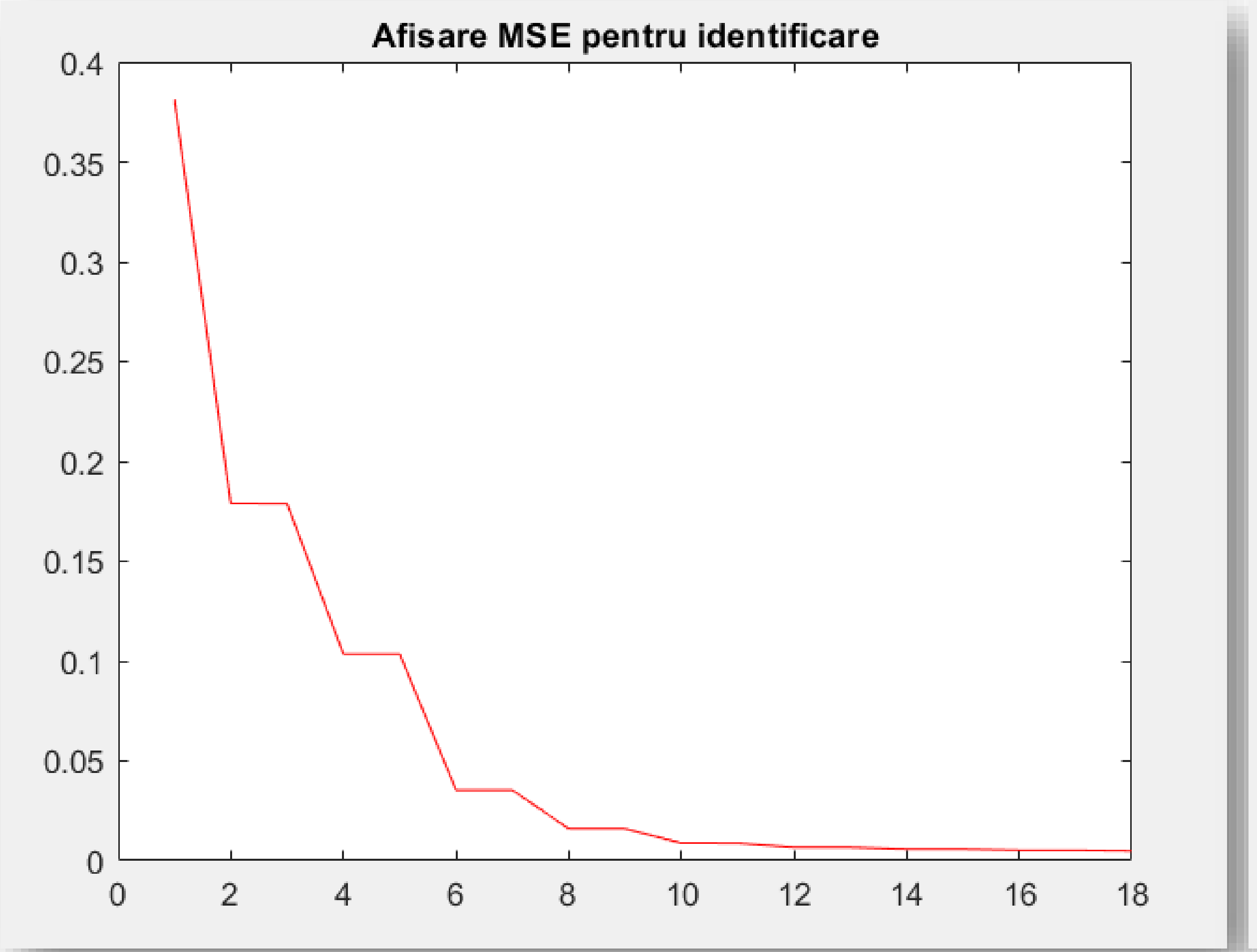
Eroarea medie patratica pe datele de validare: 0.0055

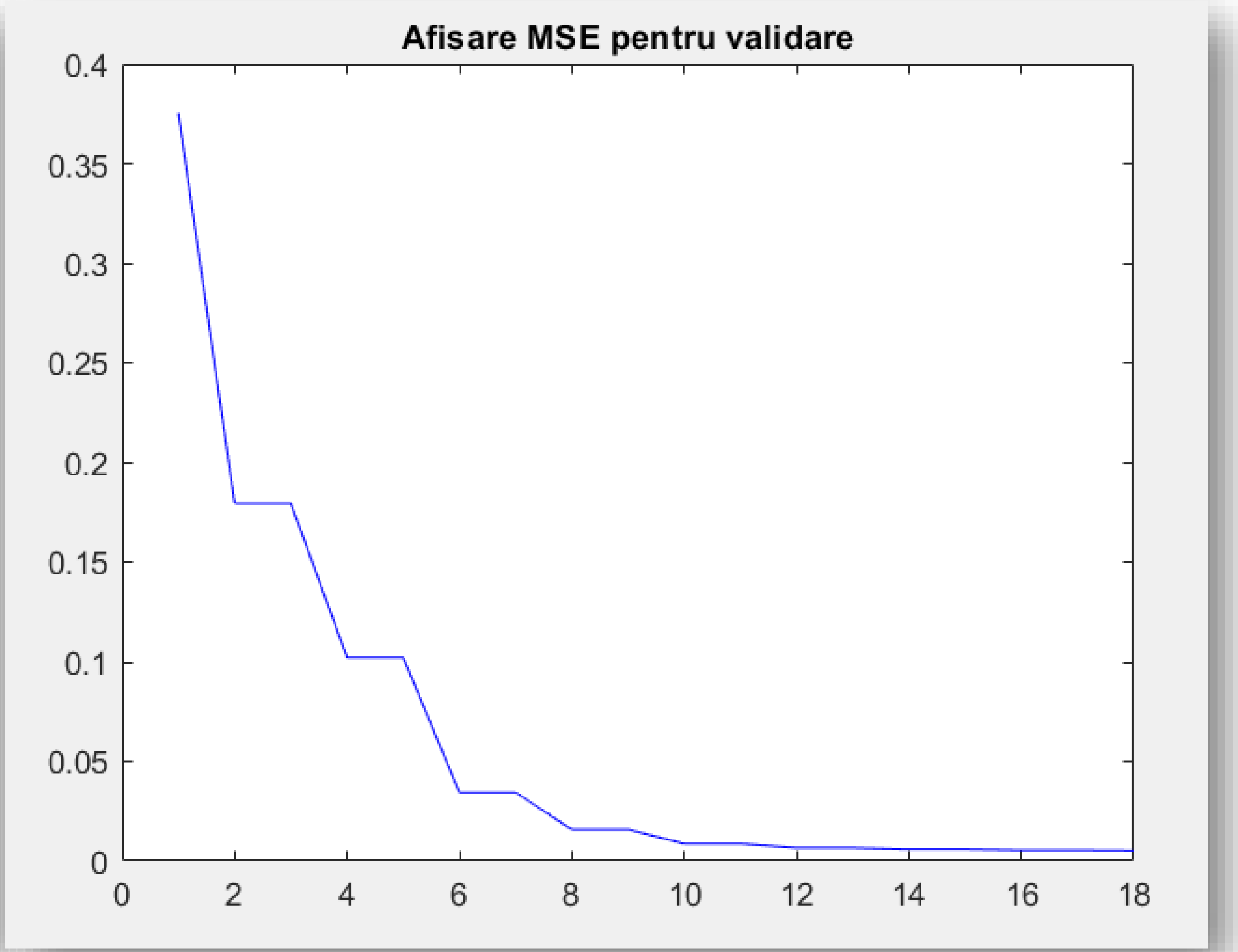


Afisarea erorii

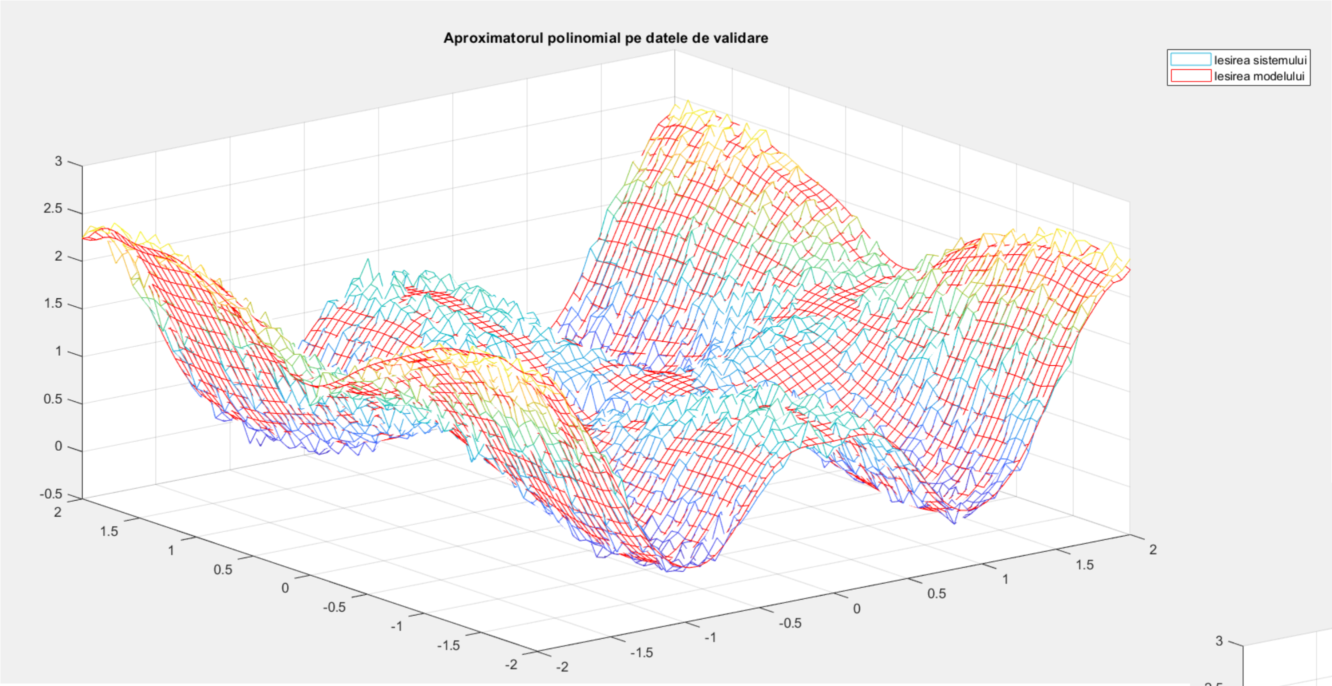
- Pentru m=18, erorile sunt apropiate, atat pentru datele de validare cat si pentru cele de identificare.

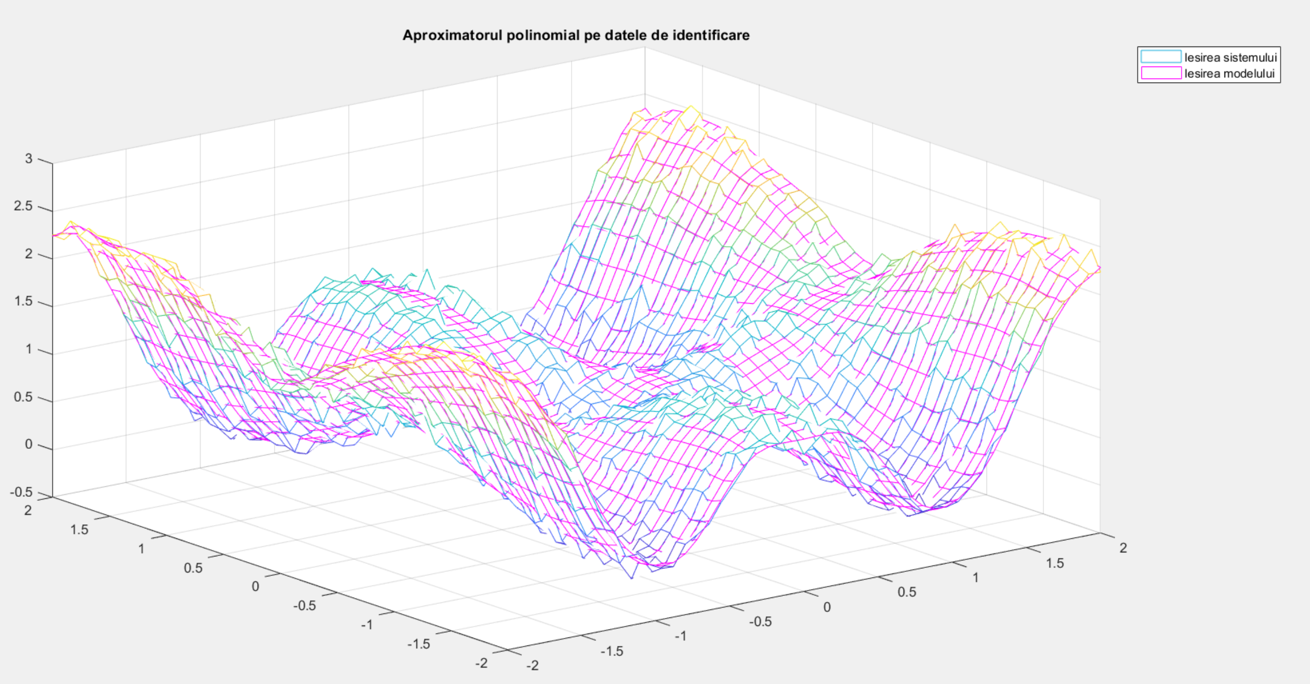



Afisare MSE pentru identificare
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 2 4 6 8 10 12 14 16 18


Afisare MSE pentru validare
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0 2 4 6 8 10 12 14 16 18

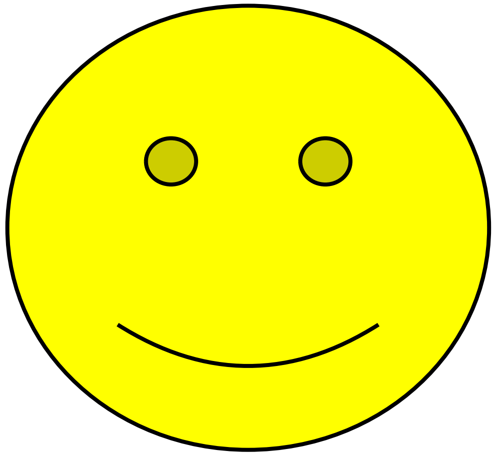
Suprapunerea aproximatorului pe datele reale





Concluzie

- Cu ajutorul regresiei liniare studiat in cursul 2 am generat fiecare regresor folosind matricea ϕ.
- Antrenarea modelului s-a bazat pe gasirea vectorului cel mai precis de parametri θ pentru ca aproximatorul polinomial sa fie cat mai apropiat de functia care genereaza iesirea.



Anexa cu listingul codului

```
%incarcarea fisierelor cu datele primite
load('iddata-11.mat')
load('proj_fit_11.mat')
% afisarea seturilor de date
figure;
mesh(id.X{1},id.X{2},id.Y)
title('Datele de identificare');
figure;
mesh(val.X{1},val.X{2},val.Y)
title('Datele de validare');
%dezvoltarea unui model folosind aproximatorul polinomial
m=18;
idN=length(id.X{1})*length(id.X{2});
valorimseid=zeros(1,m);
valN=length(val.X{1})*length(val.X{2});
valorimseval=zeros(1,m);
```

```
for l=1:m
    %creez vectorul phi pentru datele de identificare
    phi1=polinom_generat(id.X{1},id.X{2},l);
    phimatrice=phi1;
    Y=reshape(id.Y,[],1);
    theta=phimatrice\Y;
    yaproximat=phimatrice*theta;
    suma1=0;
    for i=1:idN
        suma1=suma1+(yaproximat(i)-Y(i)).^2;
    end
    id_mse=(1/idN)*suma1;
    valorimseid(l)=id_mse;
    %creez vectorul phi pentru datele de validare
    phi2=polinom_generat(val.X{1},val.X{2},l);
    validarephimatrice=phi2;
    valY=reshape(val.Y,[],1);
    validare_yaproximat=validarephimatrice*theta;
    suma2=0;
    for i=1:valN
        suma2=suma2+(validare_yaproximat(i)-valY(i)).^2;
    end
    val_mse=(1/valN)*suma2;
    valorimseval(l)=val_mse;
end
```

```
%afisarea valorilor MSE
disp('Eroarea medie patratica pe datele de identificare:');
disp(id_mse);
disp('Eroarea medie patratica pe datele de validare:');
disp(val_mse);


%afisarea graficelor ce contin valorile MSE pe ambele seturi de date
figure;
plot(1:m, valorimseid, 'r')
hold on;
plot(1:m,valorimseval,'b')
legend('Eroare id', 'Eroare val');
title('MSE pe ambele seturi de date’);

%suprapunerea rezultatului dat de aproximatorul polinomial pentru datele de identificare
figure;
mesh(id.X{1},id.X{2},id.Y);
hold on;
mesh(id.X{1},id.X{2},reshape(yaproximat,length(id.X{1}),length(id.X{2})),'EdgeColor','magenta');
legend('Iesirea sistemului','Iesirea modelului');
title("Aproximatorul polinomial pe datele de identificare");
```

```
%suprapunerea rezultatului dat de aproximatorul polinomial pentru datele de validare
figure;
mesh(val.X{1},val.X{2},val.Y);
hold on;
mesh(val.X{1}, val.X{2}, reshape(validare_yaproximat,length(val.X{1}),length(val.X{2})),'EdgeColor','red');
legend('Iesirea sistemului','Iesirea modelului');
title('Aproximatorul polinomial pe datele de validare');

%afisarea rezultatului dat de aproximator pe datele de identificare
figure;
subplot(211)
mesh(id.X{1},id.X{2},id.Y);
subplot(212)
mesh(id.X{1},id.X{2},reshape(yaproximat,length(id.X{1}),length(id.X{2})));
title("Aproximarea pentru datele de identificare");

%afisarea rezultatului dat de aproximator pe datele de validare
figure;
subplot(211)
mesh(val.X{1},val.X{2},val.Y);
subplot(212)
mesh(val.X{1}, val.X{2}, reshape(validare_yaproximat,length(val.X{1}),length(val.X{2})));
title('Aproximarea pentru datele de validare');
```

```
function phi=polinom_generat(x1,x2,m)
    dim1=length(x1);
    dim2=length(x2);
    if m==1
        phi=zeros(dim1*dim2,m+2);
    else
        phi=zeros(dim1*dim2,((m+1)*(m+2))/2);
    end
    linie=1;
    for i=1:dim1
        for j=1:dim2
        coloana=2;
        phi(linie,1)=1;
        for px1=1:m
            for px2=0:px1
                phi(linie,coloana)=(x1(i).^(px1-px2)).*(x2(j).^px2);
                coloana=coloana+1;
            end
        end
        linie=linie+1;
        end
    end
end
```