

Laboratorul 1. INTRODUCERE ÎN MATLAB

Bibliografie:

1. M. Ghinea, V. Firețeanu, *Matlab: Calcul numeric- Grafică-Aplicații*, ed. Teora, București, 1998.
2. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Aplicarea Matlab-ului in calcule matematice fundamentale.
- 2) Calcul numeric în Matlab cu aplicații în Algebră.
- 3) Rezolvarea in Matlab a ecuațiilor si sistemelor de ecuatii diferențiale.
- 4) Notiunea fisier in Matlab: creare, scriere, adaugare de date, citire.
- 5) Grafica in Matlab: reprezentari grafice 2D, reprezentarea grafica a histogramelor, reprezentari grafice 3D.

MATLAB-ul este un pachet de programe de o performanță remarcabilă care are o vastă aplicabilitate atât în domeniul științei cât al ingineriei.

Pentru lansarea în execuție a programului se acționează dublu click pe pictograma Matlab de pe Desktop sau se selectează Start → All Programs → Matlab.

APLICAȚII ÎN CALCULE MATEMATICE FUNDAMENTALE

1) Să se calculeze expresiile următoare:

a) $\sqrt[3]{17+5\sqrt{2}} + \sqrt[3]{17-5\sqrt{2}}$

```
>> (17+5*sqrt(2))^(1/3)+(17-5*sqrt(2))^(1/3)
```

```
ans =
```

```
5.0367
```

b) $\left(\frac{1+i}{1-i}\right)^3 - \left(\frac{1-i}{1+i}\right)^3$

```
>> ((1+i)/(1-i))^3-((1-i)/(1+i))^3
```

```
ans =
```

```
0 - 2.0000i
```

c) $B = \log_3 \sqrt[5]{x}$, pentru $x = 7$;

Folosim formula de schimbare a bazei logaritmice

2

$$\log_a x = \frac{\log_b x}{\log_b a}.$$

```
>> x=7;
```

```
>> log(x^(1/5))/log(3)
```

```
ans =
```

```
0.3542
```

d) $C = \cos^4 x + \cos^4 3x + \cos^4 5x + \cos^4 7x$, pentru $x = \frac{\pi}{8}$;

```
>> x=pi/8;
```

```
>> C=cos(x)^4+cos(3*x)^4+cos(5*x)^4+cos(7*x)^4
```

```
C =
```

```
1.5000
```

2) Să se sorteze în ordine descrescătoare elementele vectorului

$x = (-0.76 \ -1 \ 20 \ 8 \ -7)$, cu precizarea indicelui fiecărui element.

```
>> x=[-0.76 -1 20 8 -7]
```

```
x =
```

```
-0.7600 -1.0000 20.0000 8.0000 -7.0000
```

```
>> [y,I]=sort(x,'descend')
```

```
y =
```

```
20.0000 8.0000 -0.7600 -1.0000 -7.0000
```

```
I =
```

```
3 4 1 2 5
```

3) Se consideră matricea $A = \begin{pmatrix} 1 & -7 & 99 & 3 \\ 4.7 & 0 & 0.9 & -7 \\ 5.7 & 4 & 5 & 78 \\ -78 & 12 & -7.4 & 3 \end{pmatrix}$.

Se cere:

a) Transformați matricea A într-un vector coloană b ;

```
>> A=[1 -7 99 3;4.7 0 0.9 -7;5.7 4 5 78;-78 12 -7.4 3];
```

```
>> b=A(:);
```

b) Să se extragă submatricea D de dimensiune 3×2 , ce constă din elementele situate pe ultimele trei linii și primele două coloane ale matricei A .

```
>> D=A(2:4,1:2)
```

```
D =
```

```
4.7000    0
5.7000   4.0000
-78.0000  12.0000
```

CALCUL NUMERIC ÎN MATLAB CU APLICAȚII ÎN ALGEBRĂ

4) Rezolvați ecuațiile neliniare:

a) $1 + x = \arctg x$

```
>> f=@(x) 1+x-atan(x);
```

```
>> x=fzero(f,[-3 3])
```

```
x =
```

```
-2.1323
```

```
>> f(x)
```

```
ans =
```

```
2.2204e-016
```

b) $x^2 - \cos \pi x = 0$

```
>> f=@(x) x.^2-cos(pi*x);
```

```
>> s=fsolve(f,[-0.5,0.5])
```

```
s =
```

```
-0.4384    0.4384
```

```
>> f(s)
```

```
ans =
```

```
1.0e-011 *
```

```
0.962    0.1001
```

5) Rezolvați sistemele neliniare:

a)
$$\begin{cases} x^3 + y^3 - 6x + 3 = 0 \\ x^3 - y^3 - 6y + 2 = 0 \end{cases} \quad (\text{considerând ca punct inițial } (0.5, 0.5))$$

Pasul 1. Definim funcția vectorială f .

4

```
>>f=@(x) [x(1)^3+x(2)^3-6*x(1)+3;  
           x(1)^3-x(2)^3-6*x(2)+2];
```

Pasul 2. Rezolvăm sistemul neliniar.

```
>> s = fsolve(f,[0.5 0.5])  
  
s =  
    0.53237037226762  0.35125744755245
```

Pasul 3. Efectuăm verificarea soluției pe care am obținut-o.

```
>> f(s)  
  
ans =  
    1.0e-009 *  
    0.29623858921468  
    0.19358559200100
```

$$\text{b) } \begin{cases} x^2 + y^2 + z^2 = 9 \\ xyz = 1 \\ x + y - z^2 = 0 \end{cases} \quad (\text{considerând ca punct inițial } (2.5, 0.2, 1.6)).$$

```
>> f=@(x) [x(1)^2+x(2)^2+x(3)^2-9;x(1)*x(2)*x(3)-1;x(1)+x(2)-x(3)^2]  
  
f =
```

```
@(x) [x(1)^2+x(2)^2+x(3)^2-9;x(1)*x(2)*x(3)-1;x(1)+x(2)-x(3)^2]  
  
>> s = fsolve(f,[2.5 0.2 1.6])
```

```
s =  
    2.49137569683072  0.24274587875742  1.65351793930053
```

```
>> f(s)  
  
ans =  
    1.0e-011 *  
    0.11226575225010  
    0.13493650641294  
   -0.05244693568329
```

REZOLVAREA IN MATLAB A ECUAȚIILOR SI SISTEMELOR DE ECUATII DIFERENȚIALE

6) Să se integreze ecuația de tip Riccati

$$xy' = y^2 - (2x+1)y + x^2 + 2x, \quad x > 0$$

```
>> y=dsolve('x*Dy=y^2-(2*x+1)*y+x^2+2*x','x')
```

$y =$

$$\frac{-x-1+x^2*C1}{(-1+x*C1)}$$

7) Să se integreze ecuația diferențială Euler:

$$xy''' + y'' = 1 + x$$

`>> y=dsolve('x*D3y+D2y=1+x','x')`

$y =$

$$1/12*x^3+x*\log(x)*C1-C1*x+1/2*x^2+C2*x+C3$$

8) Să se rezolve următorul sistem de ecuații diferențiale liniare neomogen:

$$\begin{cases} y_1' = y_2 + y_3 - x - x^2 \\ y_2' = 3y_1 + y_3 - 2 - x^2 \\ y_3' = 3y_1 + y_2 + x - 3 \end{cases}$$

`>> [y1,y2,y3]=dsolve('Dy1=y2+y3-x-x^2','Dy2=3*y1+y3-2-x^2','Dy3=3*y1+y2+x-3','x')`

$y1 =$

$$1+2/3*C2*\exp(3*x)-C3*\exp(-2*x)$$

$y2 =$

$$x+\exp(-x)*C1+C2*\exp(3*x)+C3*\exp(-2*x)$$

$y3 =$

$$-\exp(-x)*C1+C2*\exp(3*x)+C3*\exp(-2*x)+x^2$$

NOTIUNEA FISIER IN MATLAB

9) Construiti un program in Matlab, cu ajutorul caruia:

- sa se creeze un fisier in care vor fi salvate doua matrice, una de tipul (2,2) si cealalta (2,3) ;
- sa se citeasca continutul fisierului in alte doua matrice.

Se va construi un script in Matlab cu numele fisier.m, avand continutul:

`W=[1 6; 7 8]; w=[-2 3 5; 6 7 8];`

`fid=fopen('pond1fb','w');`

`fprintf(fid,'%f\t',W);`

`fprintf(fid,'%f\t',w);`

`fclose(fid);`

`fid=fopen('pond1fb','r');`

6

```
a=fscanf(fid,'%f\t',[2,3]);  
b=fscanf(fid,'%f\t',[2,2]);  
fclose(fid);
```

In linia de comanda se scrie

```
>> fisier
```

GRAFICA IN MATLAB

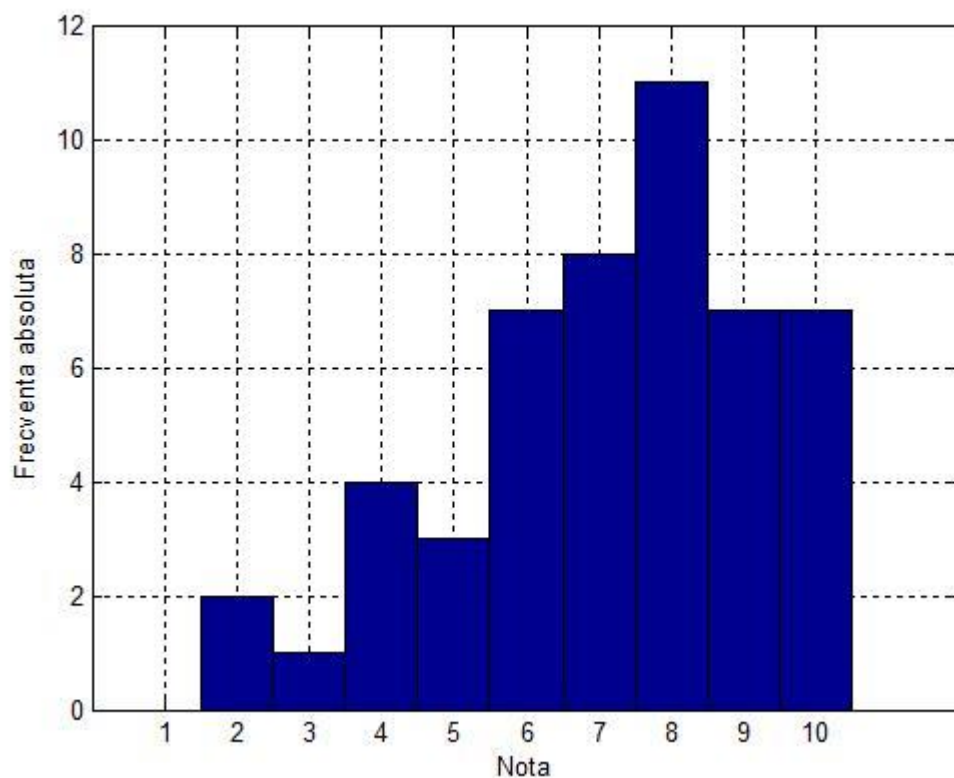
10) Reprezentati grafic in Matlab printr-o histograma, notele obtinute de cei 50 de studenti inscrisi la cursul de Statistica. Notele se vor citi dintr-un fisier text.

Se construiesc un script in Matlab, cu denumirea fis.m, ce contine:

```
fid=fopen('pond.txt','r');  
u=fscanf(fid,'%f\t',[1,50]);  
fclose(fid);
```

In linia de comanda scriem:

```
>> fis  
>> for j=1:max(u)  
v=find(u(1:length(u))==j);  
n(j)=length(v);  
end  
>> sum(n)  
ans =  
50  
>> v=1:10;  
>> hist(u,v)  
>> grid on
```



11) Reprezentați grafic următoarele funcții în plan:

$$a) f(x) = \arcsin \frac{2x}{1+x^2}, \quad x \in [-5, 5]$$

Pasul 1. Se fixează intervalul pe care va fi reprezentată funcția și un anumit pas.

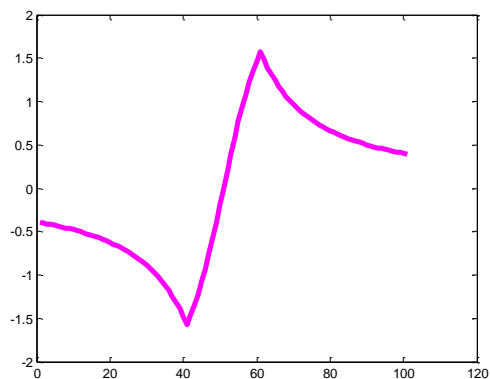
`>>x=-5:0.1:5;`

Pasul 2. Definim funcția ce urmează să fie reprezentată.

`>> f=@(x) asin(2*x./(1+x.^2));`

Pasul 3. Realizăm reprezentarea grafică.

`>> plot(f(x),'m','LineWidth',4)`



8

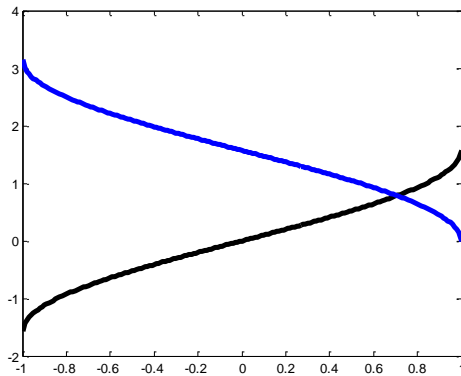
b) $f(x) = \arcsin x$, $g(x) = \arccos x$, $x \in [-1, 1]$

```
>>x=-1:0.01:1;
```

```
>> f=@(x) asin(x);
```

```
>> g=@(x) acos(x);
```

```
>> plot(x,f(x),'r',x,g(x),'b','LineWidth',3)
```



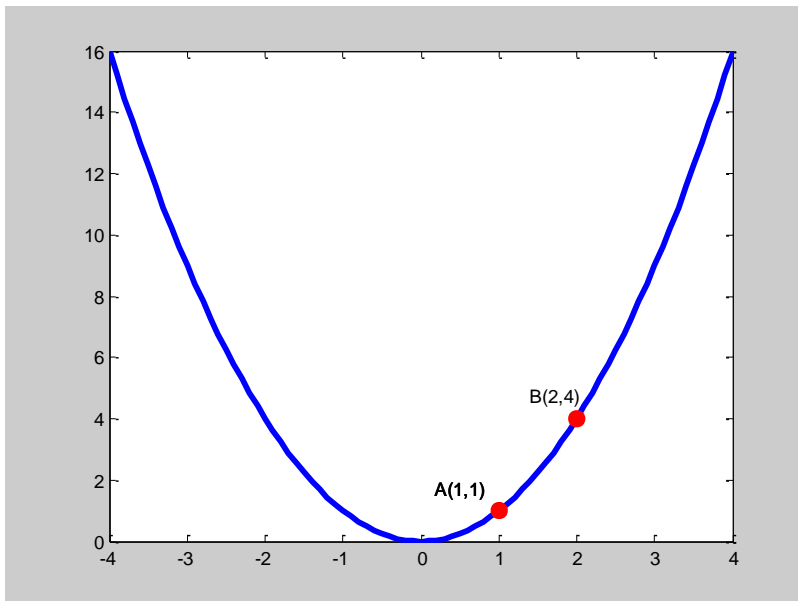
12) Reprezentați arcul de parabolă $AB : y = x^2$, care unește punctele $A(1,1)$ și $B(2,4)$.

Secvența Matlab următoare permite reprezentarea arcului AB .

```
>> x=-4:.1:4;
```

```
>> y=x.^2;
```

```
>> plot(x,y,1,1,'or',2,4,'or')
```



13) Scrieți un fișier “function” în Matlab pentru a reprezenta grafic funcția

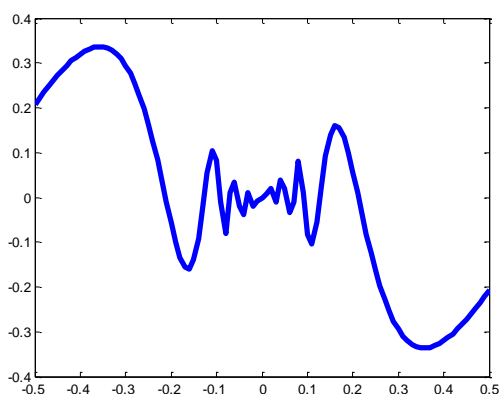
$$f(x) = \begin{cases} x \cos \frac{1}{x}, & x \neq 0 \\ 0, & x = 0 \end{cases}, x \in [-0.5, 0.5], h = 0.01$$

Se selectează succesiv File->New->M-file și se scriu următoarele instrucțiuni

```
function r=f(x)
if x~=0
    r=x*cos(1/x);
elseif x==0
    r=0;
end
end
```

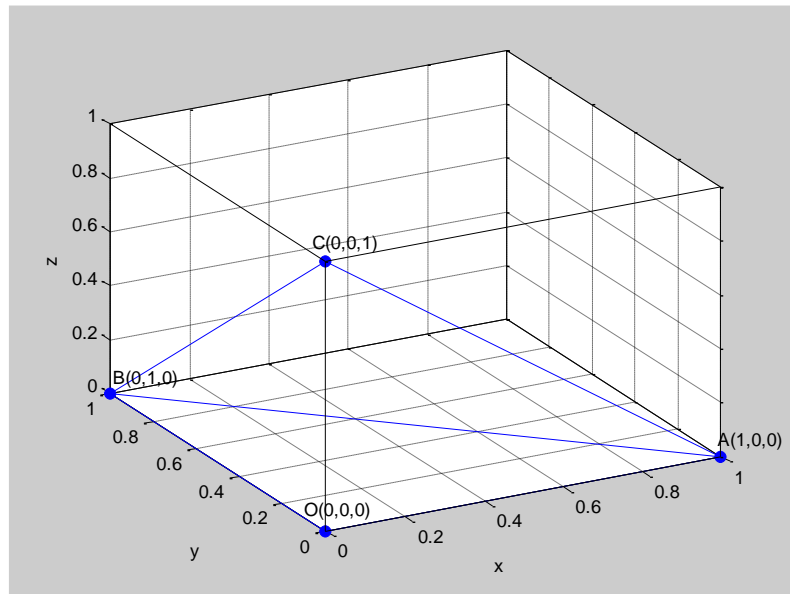
Se salvează fișierul cu f.m apoi în linia de comanda se scrie:

```
>> x=-0.5:0.01:0.5;
>> for k=1:length(x)
y(k)=f(x(k));
end
>> plot(x,y)
```



14) Reprezentați grafic în Matlab corpul solid, de forma tetraedrului din primul octant, marginit de planele $x + y + z = 1$, $x = 0$, $y = 0$, $z = 0$.

```
>> x=[1 0 0 0 1 0]; y=[0 0 1 0 0 1]; z=[0 0 0 1 0 0];
>> plot3(x,y,z,1,0,0,'ob',0,1,0,'ob',0,0,1,'ob',0,0,0,'ob')
```



15) Realizati in Matlab graficul corpului, limitat de suprafetele: $y^2 + 2z^2 = 4x$ si $x = 2$.

Secventa Matlab urmatoare permite reprezentarea corpului omogen.

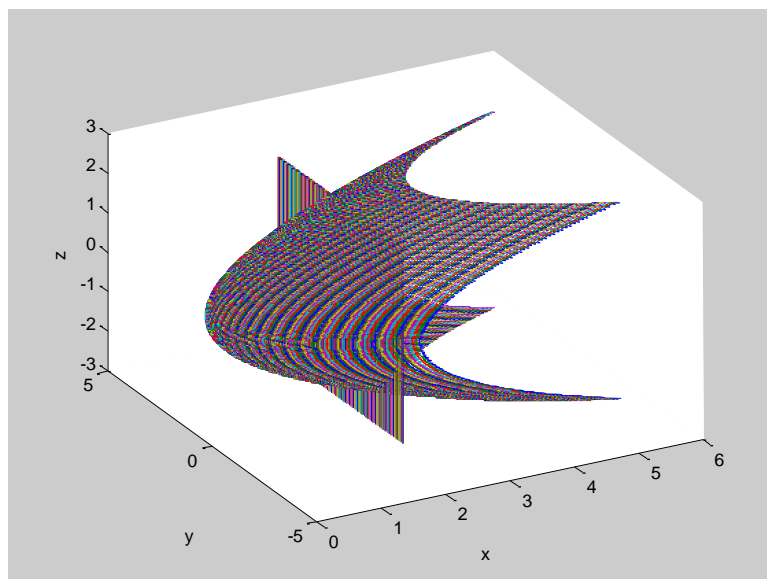
```
>> [y,z]=meshgrid(-3:.03:3,-2.5:.03:2.5);
```

```
>> x=y.^2/4+z.^2/2;
```

```
>> [m,n]=size(x);
```

```
>> x1=2*ones(m,n);
```

```
>> plot3(x,y,z,x1,y,z)
```



Laboratorul 2. Aplicații ale testelor Chauvenet și Young

Bibliografie:

1. Văduva, I. *Modele de simulare*, Editura Universității din București, 2004.
2. Trandafir, R. *Note de curs*, Facultatea de Hidrotehnică, an III, an AIA, 2011-2012.
3. M. Marusteri, *Notiuni fundamentale de biostatistică. Note de curs*. University Press, Târgu Mureș, 2006, http://www.umftgm.ro/info/Curs_Notiuni_fundamentale.pdf.
4. M. Ghinea, V. Fireșanu, *Matlab: Calcul numeric- Grafică-Aplicații*, ed. Teora, București, 1998.
5. I. Tutunea, *Algoritmi, logica și programare*, Reprografia Universității Craiova, 1993.
6. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Aplicarea testului Chauvenet privind eliminarea datelor afectate de erori aberante;
- 2) Aplicarea testului Young pentru identificarea erorilor sistematice.

TESTUL CHAUVENET

Fiind date valorile, (x_1, \dots, x_n) rezultate din măsurători, se consideră că valoarea x_i este afectată de erori aberante dacă verifică relația:

$$|x_i - \bar{x}| > z\sigma, \quad (1)$$

unde:

- $\bar{x} = \frac{\sum_{j=1}^n x_j}{n}$ este media aritmetică a valorilor observate,
- $\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ semnifică abaterea standard a valorilor observate,
- z se ia din tabele (în funcție de numărul n de valori din sir, ce reprezintă dimensiunea sirului sau volumul esantionului) sau se aproximează conform relației următoare: $z = \frac{0,435 - 0,862a}{1 - 3,604a + 3,213a^2}$, cu $a = \frac{2n-1}{4n}$.

Observatie. Este suficient ca verificarea relației (1) să fie efectuată doar pentru valorile extreme (minima și maxima) din cadrul esantionului.

Dacă, în urma aplicării testului, rezulta că una dintre valorile testate este afectată de erori aberante, valoarea respectivă este eliminată din cadrul esantionului, se recalculează valorile mediei și abaterii standard pentru valorile rămase și se reia verificarea condiției (1),

algoritmul aplicându-se pâna când conditia respectiva nu mai este verificata pentru nici una dintre cele doua valori extreme ale esantionului.

Exemplul 1. În simularea utilizării unei imprimante conectate la o rețea se urmărește repartiția numărului maxim de fișiere care sunt in lista de așteptare pentru listare pe o perioada de 30 de zile, înregistrându-se valorile:

săpt	L	Ma	Mi	J	V	săpt	L	Ma	Mi	J	V
I	15	19	13	18	20	IV	19	21	13	23	17
II	12	21	22	19	21	V	18	20	21	13	15
III	13	13	13	14	17	VI	20	19	12	18	12

Să se verifice dacă acest eșantion are valori aberante și dacă există, să se elimine! Scrieți un program Matlab pentru testul Chauvenet. Datele de intrare se iau dintr-un fișier text.

Componentele esantionului	x_i	x_i (ordonate crescator)	Valori extreme	$ x_i - \bar{x} $	$z\sigma$
1	15	12	12	5.0333	8.3158
2	19	12	12	5.0333	
3	13	12	12	5.0333	
4	18	13	23	5.9667	
5	20	13			
6	12	13			
7	21	13			
8	22	13			
9	19	13			
10	21	14			
11	13	15			
12	13	15			
13	13	17			
14	14	17			
15	17	18			
16	19	18			
17	21	18			
18	13	19			
19	23	19			
20	17	19			
21	18	19			
22	20	20			
23	21	20			
24	13	20			
25	15	21			
26	20	21			

27	19	21
28	12	21
29	18	22
30	12	23
$\bar{x} = 17.0333, \sigma = 3.5183$		

In concluzie, nici una dintre valorile extreme **nu este afectata de erori aberante**.

Exemplul 2. A fost măsurată greutatea a 15 indivizi adulți. Rezultatele măsurătorilor sunt cele din tabelul următor. Să se verifice dacă acest eșantion are valori aberante și dacă există, să se elimine! Scrieți un program Matlab pentru testul Chauvenet. Datele de intrare se iau dintr-un fișier text.

Prima aplicare a criteriului Chauvenet

Componentele esantionului	x_i	x_i (ordonate crescator)	Valori extreme	$ x_i - \bar{x} $	$z\sigma$
1	58	35	35	38.9333	56.53065
2	60	50	160	86.06667	
3	80	55			
4	77	58			
5	83	60			
6	75	65			
7	82	70			
8	79	75			
9	50	77			
10	35	79			
11	70	80			
12	160	80			
13	80	82			
14	65	83			
15	55	160			
$\bar{x} = 73.9333, \sigma = 26.652$					

După cum rezultă din tabel, valoarea 160 va trebui să fie eliminată din datele supuse prelucrării.

A doua aplicare a criteriului Chauvenet

Componentele esantionului	x_i (ordonate crescator)	Valori extreme	$ x_i - \bar{x} $	$z\sigma$
1	35	35	32.7857	29.19275
2	50	83	15.2143	
3	55			
4	58			

5	60
6	65
7	70
8	75
9	77
10	79
11	80
12	80
13	82
14	83
$\bar{x} = 67.7857, \sigma = 13.93443$	

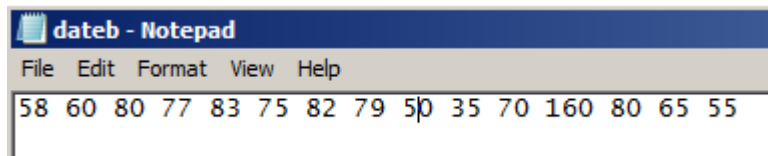
De data aceasta trebuie eliminata din tabel valoarea 35.

A treia aplicare a criteriului Chauvenet

Componentele esantionului	x_i (ordonate crescator)	Valori extreme	$ x_i - \bar{x} $	$z\sigma$
1	50	50	20.30769	22.64275
2	55	83	12.69231	
3	58			
4	60			
5	65			
6	70			
7	75			
8	77			
9	79			
10	80			
11	80			
12	82			
13	83			
$\bar{x} = 70.30769, \sigma = 10.95715$				

Utilizand Matlab 7.9, vom construi programul corespunzator testului Chauvenet:

Etapa 1. Din fisierul dateb.txt se vor citi datele initiale.



Etapa 2. Se scrie functia sterge.m, ce permite stergerea unei componente a unui vector.

```
function v=sterge(u,k,n)
for i=1:k-1
    v(i)=u(i);
end
if n~=0
    for j=k:n-1
```

```

        v(j)=u(j+1);
    end
end
end

```

Etapa 3. Se scrie programul propriu- zis.

```

fid=fopen('dateb.txt','r');
u=fscanf(fid,'%f\t',[1,15]);
fclose(fid);
m=mean(u);
sigma=std(u);
n=length(u);
a=(2*n-1)/(4*n);
z=(0.435-0.862*a)/(1-3.604*a+3.213*a^2);
it=1;
u=sort(u);
mi=min(u);
ma=max(u);
p1=length(find(u==mi));
p2=length(find(u==ma));
for i=1:p1
    w(i)=mi;
end
for i=p1+1:p1+p2
    w(i)=ma;
end
q=find(abs(w-m)>z*sigma);
while (length(q)~=0)
    it=it+1;
    for k=1:length(q)
        h=find(u==w(q(k)));
        for l=1:length(h)
            v=sterge(u,h(l),n);
            n=n-1;
            u=v;
        end
    end
end
m=mean(u);
sigma=std(u);
n=length(u);
a=(2*n-1)/(4*n);
z=(0.435-0.862*a)/(1-3.604*a+3.213*a^2);
u=sort(u);
mi=min(u);
ma=max(u);
p1=length(find(u==mi));
p2=length(find(u==ma));
for i=1:p1
    w(i)=mi;
end
for i=p1+1:p1+p2
    w(i)=ma;
end
q=find(abs(w-m)>z*sigma);
end

```

TESTUL LUI YOUNG

Problema depistării și eliminării erorilor sistematice este mai dificilă datorită multitudinii de factori care se intercondiționează. Vom prezenta testul lui Young, test care nu oferă posibilitatea eliminării erorilor sistematice, ci doar pe aceea a aprecierii influenței cauzelor sistematice asupra datelor de sondaj.

Pasul 1. Intrare: (x_1, \dots, x_n) - șir de date experimentale și α probabilitatea de acceptare (coeficient de încredere); se calculează mărimile:

$$\begin{cases} \delta^2 = \frac{1}{n-1} \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 \\ M = \frac{\delta^2}{\sigma^2}. \end{cases}$$

Pasul 2. Se compară M cu valoarea critică (VCI) inferioară și valoarea critică superioară (VCS), $VCI < M < VCS$, valori luate din tabele în funcție de n și α sau determinate cu relațiile:

a) Dacă $\alpha = 0,95$ atunci

$$\begin{cases} VCI = 0,491 + 0,081n - 0,003n^2 \\ VCS = 3,317 - 1,057e^{-8,919n^{-0,341}} \end{cases}$$

b) Dacă $\alpha = 0,99$ atunci

$$\begin{cases} VCI = \frac{192,883 + 1,269n^{2,33}}{411,427 + n^{2,33}} \\ VCS = 3,484 - 0,882e^{-33,574n^{-1,388}} \end{cases}$$

Dacă inegalitatea este satisfăcută, atunci se consideră că șirul de date experimentale are un caracter aleator (nu este afectat de erori sistematice) cu probabilitatea α .

Pasul 3. Stop!

Observatie. În tabele se dau valori pentru α corespunzătoare diferitelor valori ale lui n ($n \leq 25$)

Exemplul 3. Testati dacă esanționul de date experimentale, ce constă din valorile x_i ale timpului de latență a instalării efectului hipnotic în cazul amobarbitalului (vezi tabelul următor) are sau nu un caracter aleator cu probabilitatea $\alpha = 0.95$:

Componentele esantionului	x_i (secunde)
1	16.1
2	15.5
3	13.4
4	22.8
5	12.1
6	11.3
7	11.6
8	6.3
9	8.8
10	7.1

```
function [VCI, VCS]=valcrit(n,al)
if al==0.95
    VCI=0.491+0.081*n-0.003*n^2;
    VCS=3.317-1.057*exp(-8.919*n^(-0.341));
elseif al==0.99
    VCI=(192.883+1.269*n^2.33)/(411.427+n^2.33);
    VCS=3.484-0.882*exp(-33.574*n^(-1.388));
end
end

fid=fopen('data.txt','r');
x=fscanf(fid,'%f\t',[1,10]);
fclose(fid)
n=10;al=0.95
sigma=var(x)
delta=0;
for i=1:n-1
    delta=delta+(x(i+1)-x(i))^2;
end
delta=delta/(n-1);
M=delta/sigma
[VCI,VCS]=valcrit(10,al);
if VCI<M<VCS
    disp(['esantionul datelor experimentale are un caracter aleator cu
probabilitatea:', num2str(al)]);
else
    disp('esantionul datelor experimentale este afectat de erori
sistematice')
end
```

Laboratorul 3. Aplicatii ale testelor Massey si χ^2

Bibliografie:

1. G. Ciucu, V. Craiu, A. Stefanescu. *Statistica matematica si cercetari operationale*, ed. Didactica si pedagogica, Bucuresti, 1974.
2. I. Văduva. *Modele de simulare*, Editura Universitatii din București, 2004.
3. I. Vladimirescu, *Probabilitati si statistica*, Note de curs, Facultatea de Matematica si Informatica, Universitatea din Craiova, an III, 1995-1996.
4. R. Trandafir. *Modele de simulare*, Note de curs, Facultatea de Hidrotehnica, an III, AIA, 2011-2012.
5. I. Armeanu, V. Petrehus, *Probabilitati si statistica aplicate in biologie*, MatrixRom, Bucuresti, 2006.
6. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Listarea unor functii predefinite in Matlab, ce vor fi utilizate in cadrul acestei lucrari de laborator.
- 2) Aplicarea testelor Massey si χ^2 pentru verificarea normalitatii unei serii de date si implementarea in Matlab 7.9 a algoritmilor corespunzatori acestor teste.

In cadrul acestei lucrari de laborator vom utiliza urmatoarele functii Matlab 7.9:

Nume functie	Semnificatie
sqrt(x)	Calculeaza \sqrt{x}
exp(x)	Calculeaza e^x
int(f(x),a,b)	Calculează $\int_a^b f(x)dx$
sort(x)	Sortează în ordine crescătoare elementele vectorului x
length(x)	Returneaza numarul de component ale vectorului x
abs(x)	Determina valoarea absoluta a lui x
u=find(x<=a)	Returnează in u indicii elementelor din vectorul x , care au valoarea mai mica sau egala cu a
m=max(v)	Determina cea mai mare component a vectorului v
mean(x)	Calculeaza valoarea medie a elementelor lui x
std(x)	Calculeaza abaterea standard pentru valorile individuale ale unei selectii x
sum(x)	Calculeaza suma componentelor lui x
eval	Evalueaza sirurile de caractere cu expresii Matlab

vpa	Afiseaza un rezultat cu o anumita precizie
fopen	Realizeaza deschiderea unui fisier
fclose	Realizeaza inchiderea unui fisier
fscanf	Realizeaza citirea dintr-un fisier

TESTUL MASSEY

Acest test se utilizeaza pentru verificarea normalității unei selectii de volum mai mic decât 32.

Pentru aplicarea acestui test se procedeaza astfel:

Pasul 1. Se ordoneaza valorile experimentale (x_1, \dots, x_n) rezultate din măsurători in ordine crescatoare.

Pasul 2. Se calculeaza: \bar{x} = media aritmetică a valorilor observate si σ = abaterea standard a valorilor selectiei.

Pasul 3. Se normalizeaza valorile rezultate din măsurători: $z_i = \frac{x_i - \bar{x}}{\sigma}$, $(\forall) i = \overline{1, n}$.

Pasul 4. Se determina valorile $\Phi(z_i)$, adica valorile functiei de repartitie teoretica, unde

$$\Phi: \mathbb{R} \rightarrow \mathbb{R}, \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-t^2/2} dt,$$

numita functia lui Laplace.

Pasul 5. Se calculeaza frecvențele relative cumulate

$$F_i = \frac{n_i}{n}, (\forall) i = \overline{1, n},$$

adica valorile functiei de repartitie empirica (experimentală), unde n_i reprezinta numarul de valori z mai mici sau egale cu valoarea z_i .

Pasul 6. Se calculeaza diferentele

$$d_i = |F_i - \Phi(z_i)|, (\forall) i = \overline{1, n},$$

și se determină

$$d_{\max} = \max_{i=1}^n d_i.$$

Pasul 7. Dacă $d_{\max} < d_{\text{critic}}$ (d_{critic} se extrage din tabele în funcție de coeficientul de încredere α adoptat și de volumul selecției n) atunci se acceptă ipoteza că eșantionul are repartiția normală; altfel se respinge ipoteza conform căreia eșantionul de valori experimentale are o distribuție normală.

Observație. Valorile pentru d_{critic} din tabele pot fi approximate prin calcul utilizând

$$\text{expresiile: } d_{\text{critic}} = \begin{cases} 0,1408 + 0,00714n - 0,000769n^2, & \text{pentru } \alpha = 0.90 \\ 0,1851 - 0,01064n + 0,000785n^2, & \text{pentru } \alpha = 0.95. \end{cases}$$

Exemplul 1. Să se verifice, la un prag de semnificație $1 - \alpha = 0.05$, ipoteza conform căreia rezistența ohmică (în $K\Omega$) a unor tronșoane de ceramică acoperite cu carbon este o variabilă normală. Verificarea se va face pe baza a 10 măsurători: 1,68; 1,74; 1,82; 1,60; 1,72; 1,90; 1,79; 1,98; 1,85; 1,93. Scrieți un program Matlab pentru aplicarea testului Massey datelor din acest exemplu.

Vom construi programul corespunzător testului Massey utilizând Matlab 7.9:

Etapa 1. Se scriu într-un fișier datele rezultate din măsurători.

Etapa 2. Se scrie funcția phi.m, corespunzătoare funcției lui Laplace.

```
function r= phi(z)
```

```
syms t
```

```
r=1/sqrt(2*pi)*int(exp(-t^2/2),t,-inf,z);
```

```
end
```

Etapa 3. Se construiește funcția cu ajutorul căreia se determină d_{critic} pentru $\alpha = 0.90$ sau $\alpha = 0.95$.

```
function dcrit=valcrit(N,al)
```

```
if al==0.95
```

```
    dcrit=0.1851-0.01064*N+0.000785*N^2;
```

```
elseif al==0.90
```

```
    dcrit=0.1408+0.00714*N-0.000769*N^2;
```

```
end
```

Etapa 4. Se construiește script-ul Massey.m, ce permite implementarea testului Massey.

```
fid=fopen('data','r');
```

```
x=fscanf(fid,'%f\t',[1,10]);
```

```

fclose(fid);
N=10;
x=sort(x);
m=mean(x);
sigma=std(x);
z=(x-m)/sigma;
for i=1:N
  phii(i)=vpa(phi(z(i)),4);
end
for i=1:N
  n(i)=length(find(z<=z(i)));
end
f=n/N;
d=abs(f-eval(phii))
dmax=max(d);
al=0.95;
dcrit=valcrit(N,al);
if dmax<dcrit
  disp('se accepta ipoteza ca esantionul de valori experimentale are repartitie
normala');
else
  disp('se respinge ipoteza ca esantionul de valori experimentale are o distributie
normala');
end

```

Modul de aplicare a testului Massey este ilustrat in tabelul urmatoar:

x_i (ordonate crescator)	z_i	$\Phi(z_i)$	F_i	d_i	d_{critic}
1,60	-1,6941	0,0451	0,1	0,0549	0,1572
1,68	-1,0198	0,1539	0,2	0,0461	
1,72	-0,6827	0,2474	0,3	0,0526	
1,74	-0,5141	0,3036	0,4	0,0964	
1,79	-0,0927	0,4631	0,5	0,0369	
1,82	0,1601	0,5636	0,6	0,0364	
1,85	0,4130	0,6602	0,7	0,0398	

1,90	0,8344	0,798	0,8	0,002	
1,93	1,0873	0,8615	0,9	0,0385	
1,98	1,5087	0,9343	1	0,0657	
$\bar{x} = 1,801$	$\sigma = 0,1186$			d_{\max}	0,0964

TESTUL χ^2

Cel mai important și mai des utilizat test de verificare a normalității repartițiilor unui sir de date experimentale este testul χ^2 .

Fie X o variabilă aleatoare și x_1, \dots, x_n o selecție de volum $n \geq 100$ asupra lui X .

Algoritmul corespunzător testului χ^2 este următorul:

Pasul 1. Valorile caracteristicii sub cercetare X se împart în k intervale de clasă:

$$(-\infty, x_1], (x_1, x_2], \dots, (x_{k-2}, x_{k-1}], (x_{k-1}, \infty)$$

unde $x_1 < x_2 < \dots < x_{k-1}$.

Conform lui Brooks și Carruthers, numărul k de intervale de clasă este

$$k < 5 \lg n$$

în timp ce formula lui Sturges este:

$$k = 1 + 3,3 \lg n,$$

n fiind volumul sondajului.

Pasul 2. Se determină frecvențele absolute n_i ale intervalelor (numărul înregistrat de valori din intervalul I) și respectiv frecvențele relative $f_i = \frac{n_i}{n}$, (\forall) $i = \overline{1, k}$, valorile funcției de repartiție empirică.

Pasul 3. Se calculează valoarea mediei aritmetice \bar{x} și respectiv abaterea standard σ a valorilor observate.

Pasul 4. Se normalizează valorile rezultate din măsurători:

$$z_i = \frac{x_i - \bar{x}}{\sigma}, \quad (\forall) i = \overline{1, k},$$

unde x_i reprezintă extremitatea dreaptă a fiecărui interval I_i .

Pasul 5. Se determină valorile $\Phi(z_i)$, adică valorile funcției de repartiție teoretică, Φ fiind funcția lui Laplace.

Pasul 6. Se calculeaza probabilitatile $p_i, (\forall) i = \overline{1, k}$ ca o observatie sa apartina clasei I , exprimate de:

$$\begin{cases} p_1 = P(-\infty < X \leq x_1) = \Phi\left(\frac{x_1 - \bar{x}}{\sigma}\right) \\ p_i = P(x_{i-1} < X \leq x_i) = \Phi\left(\frac{x_{i-1} - \bar{x}}{\sigma}\right) - \Phi\left(\frac{x_i - \bar{x}}{\sigma}\right), (\forall) i = \overline{2, k-1} \\ p_k = P(x_{k-1} < X < \infty) = 1 - \Phi\left(\frac{x_{k-1} - \bar{x}}{\sigma}\right). \end{cases}$$

Pasul 7. Se calculeaza statistica $\chi^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i}$ si se compara cu valoarea din tabele a lui $\chi_{k-s-1, 1-\alpha}^2$, unde: $s=2$ semnifica numarul de parametri estimati (s -au estimat doi parametri μ si σ^2), α este coeficientul de incredere, k este numarul intervalelor de clasa.

Exemplul 2. Sa se verifice, la un prag de semnificatie $1 - \alpha = 0.05$, ipoteza conform careia rezistenta la rupere a unor fire de bumbac este o variabila normala. Verificarea se va face pe baza a 124 masuratori:

1.97	1.69	1.65	1.69	1.72	1.81	1.70	1.77	1.84	1.81	1.96	1.81	1.87
1.77	1.85	1.77	1.78	1.69	1.68	1.74	1.71	1.69	1.92	1.74	2.03	1.88
1.61	1.75	1.77	2.01	1.74	1.9	1.87	1.76	1.77	1.82	1.86	1.88	1.99
1.61	1.68	1.73	1.95	1.86	1.83	1.61	1.77	1.94	2.02	1.83	1.84	1.82
1.63	1.78	1.93	2.02	1.92	1.77	1.98	1.86	1.62	1.73	1.81	1.7	1.62
1.76	1.93	1.95	1.86	1.9	1.85	1.94	1.97	1.69	1.72	1.9	1.97	1.74
1.8	1.98	1.91	1.82	1.74	1.85	1.74	1.93	1.8	1.79	2.01	1.64	2.02
1.64	1.65	1.81	1.72	1.86	2.03	1.7	1.73	1.84	1.91	1.96	1.99	1.73
1.86	1.85	1.67	1.78	1.8	1.68	1.89	1.71	1.99	1.63	1.92	1.94	1.67
1.67	1.75	2.01	1.87	1.9	1.93	1.76						

Scrieti un program Matlab pentru aplicarea testului χ^2 datelor din acest exemplu.

Vom construi programul corespunzator testului χ^2 utilizand Matlab 7.9:

Etapa 1. Se scriu intr-un fisier datele rezultate din măsurători.

Etapa 2. Se scrie functia phi.m, corespunzatoare functiei lui Laplace.

Etapa 3. Se construiesc script-ul hipatrat.m, ce permite implementarea testului χ^2 .

```
fid=fopen('date.txt','r');
u=fscanf(fid,'%f\t',[1,124]);
fclose(fid);
m=mean(u); sigma=std(u);
```

```

N=124; u=sort(u);
5*log10(N)
k=9;
x(1)=1.65; x(k)=inf;
for i=2:k-1
x(i)=x(1)+0.05*(i-1);
end
n(1)=length(find(u<=x(1)));
for i=2:k-1
    n(i)=length(find(u> x(i-1)& u<= x(i)));
end
n(k)=length(find(u>x(k-1)));
f=n/N; z=(x-m)/sigma;
for i=1:k
phii(i)=vpa(phi(z(i)),4);
end
p(1)=phii(1); p(k)=1-phii(k-1);
for i=2:k-1
    p(i)=eval(phii(i))-eval(phii(i-1));
end
hi_calc=sum(((n-N*eval(p)).^2)./(N*eval(p)))
hi_tabel=12.59;
if hi_calc<hi_tabel
    disp('se accepta ipoteza ca esantionul de valori experimentale are repartitie
normala');
else
    disp('se respinge ipoteza ca esantionul de valori experimentale are o
distributie normala');
end

```

Modul de aplicare a testului χ^2 este ilustrat in tabelul urmator:

Intervalul I	x_i	n_i	z_i	$\Phi(z_i)$	p_i	$\frac{(n_i - np_i)^2}{np_i}$
$(-\infty, 1.65]$	1.65	11	-1.4459	0.0741	0.0741	0.3572
$(1.65, 1.7]$	1.70	14	-1.0105	0.1561	0.082	1.4442
$(1.7, 1.75]$	1.75	17	-0.5751	0.2826	0.1265	0.1101
$(1.75, 1.8]$	1.80	14	-0.1397	0.4444	0.1618	1.8323
$(1.8, 1.85]$	1.85	16	0.2956	0.6162	0.1718	1.3202
$(1.85, 1.9]$	1.90	20	0.7310	0.7676	0.1514	0.0801
$(1.9, 1.95]$	1.95	14	1.1664	0.8783	0.1107	0.0054
$(1.95, 2]$	2	10	1.6018	0.9454	0.0671	0.3391
$(2, \infty)$	∞	8	∞	1	0.0546	0.2233
$\bar{x} = 1.801$	$\sigma = 0.1186$	$\sum_{i=1}^n n_i = 124$	$\chi_{calc}^2 = 5.7118 < \chi_{s+k-1; 1-\alpha}^2 = 12.59$			

Probleme propuse.

1. Se încercă 16 imprimante de același tip și se constată timpii de funcționare din tabel.

Nr.	T_i
1	50,5
2	60,5
3	71,6
:	:
13	212
14	260
15	285
16	304

Sa se verifice ipoteza conform căreia repartiția timpului de funcționare fără defectțiuni este normală (se considera $\alpha = 0.95$).

2. O selecție de volum $n = 11$ asupra unei caracteristici X , privind sensibilitatea la un antibiotic a condus la următoarele valori: 30, 25, 41, 30, 7, 16, 41, 70, 20, 16, 78. Verificați la un prag de semnificație $1 - \alpha = 0.05$ ipoteza conform careia caracteristica X este o variabilă normală.
3. Se caută să se pună la punct o mașină de ambalat ciment în saci de greutate nominală 50 kg. După primele reglaje, se verifică la întâmplare 500 de saci a căror greutate a fost notată în tabelul următor.

Greutatea în Kg	<45	[45-47)	[47-49)	[49-51)	[51-53)	[53-55)	[55-57)	>57	Total
Nr saci	35	53	76	100	88	78	42	28	500

Se poate considera că repartiția sacilor de ciment este normală?

Laboratorul 4. Crearea de algoritmi proprii pentru generarea variabilelor aleatoare discrete

Bibliografie:

1. I. Văduva. *Modele de simulare*, Editura Universitatii din București, 2004.
2. I. Vaduva, *Modele de simulare cu calculatorul*, Editura Tehnica, București, 1977.
3. I. Vladimirescu, *Probabilitati si statistica*, Note de curs, Facultatea de Matematica si Informatica, Universitatea din Craiova, an III, 1995-1996.
4. R. Trandafir, *Modele de simulare*, Note de curs, Facultatea de Hidrotehnica, an III, AIA, 2011-2012.
5. I. Armeanu, V. Petrehus, *Probabilitati si statistica aplicate in biologie*, MatrixRom, Bucuresti, 2006.
6. I. Iatan , *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Implementarea in Matlab a metodei inverse de simulare a unei variabile aleatoare discrete.
- 2) Construirea unor functii in Matlab pentru simularea unei variabile discrete, cu o repartitie particulara (Bernoulli, binomiala, Poisson, geometrica, hipergeometrica), utilizata in fiabilitate.

METODA INVERSA DE SIMULARE A UNEI VARIABILE ALEATOARE DISCRETE

Vom descrie o metoda de simulare a unei variabile aleatoare discrete, ce ia un numar finit de valori, numita *metoda inversa*. Conform acestei metode putem simula orice variabila aleatoare X daca cunoastem functia sa de repartitie F si putem calcula functia inversa F^{-1} .

Folosind aceasta metoda vom construi un program Matlab pentru simularea variabilei discrete X , ce are repartitia

$$X : \begin{pmatrix} a_1 & \dots & a_k & \dots & a_m \\ p_1 & \dots & p_k & \dots & p_m \end{pmatrix}, \sum_{k=1}^m p_k = 1.$$

Funcția de repartitie a acesteia este:

$$F_X(x) = \begin{cases} 0, & x \leq a_1 \\ p_1, & x \in (a_1, a_2] \\ p_1 + p_2, & x \in (a_2, a_3] \\ \vdots & \\ p_1 + p_2 + \dots + p_k, & x \in (a_k, a_{k+1}] \\ \vdots & \\ 1, & x > a_m \end{cases}$$

iar functia inversa va fi:

$$F_X^{-1}(u) = a_k, u \in (F_X(a_{k-1}), F_X(a_k)], (\forall) k = \overline{1, m},$$

unde:

$$a_0 = -\infty, \text{ iar } F_X(a_0) = 0.$$

Algoritmul pentru simularea variabilei aleatoare X consta in:

- generarea unei valori u uniform repartizate in $[0,1]$,
- determinarea indicelui k pentru care

$$F_X(a_{k-1}) < u \leq F_X(a_k) \tag{1}$$

Relatia (1) rezulta din faptul ca:

$$a_{k-1} < x \leq a_k \Rightarrow F_X(a_{k-1}) < F_X(x) \leq F_X(a_k)$$

si $F_X(x) = u$.

Vom construi programul Matlab corespunzator:

```
function x=simdiscrv(F,a,m)
```

```
u=rand;
```

```
k=1;
```

```
while (u>F(k)) & (k<m)
```

```
    k=k+1;
```

```
end
```

```
x=a(k);
```

```
end
```

Vom aplica functia Matlab precedenta pentru a genera o variabila aleatoare discreta care da numarul punctelor obtinute in experienta aruncarii unui zar o data.

Astfel

$$X : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}, \sum_{k=1}^6 p_k = 1$$

iar

$$F_X(x) = \begin{cases} 0, & x \leq 1 \\ 1/6, & x \in (1, 2] \\ 2/6, & x \in (2, 3] \\ 3/6, & x \in (3, 4] \\ 4/6, & x \in (4, 5] \\ 5/6, & x \in (5, 6] \\ 1, & x > 6 \end{cases}$$

```
>> a=1:7;
>> F=0:1/6:1;
>> x=simdiscrv(F,a,7)
```

```
u =
    0.6557
x =
     5
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE BERNOULLI

Fie X o variabila aleatoare binara astfel incat: $P(X=1)=p$, $p \in [0,1]$ si $P(X=0)=q=1-p$, adica: $X=1$ daca intr-o experienta intamplatoare, un eveniment aleator observabil A se produce cu probabilitatea p (avem de-a face cu un *succes*) si $X=0$ daca se produce evenimentul contrar \bar{A} cu probabilitatea $q=1-p$ (se realizeaza un *esec*).

Asadar X are repartitia:

$$X : \begin{pmatrix} 0 & 1 \\ q & p \end{pmatrix}, M(X)=p, \text{Var}(X)=pq=p(1-p).$$

Functia de repartitie a lui X este:

$$F(x) = P(X < x) = \begin{cases} 0, & x \leq 0 \\ q, & x \in (0, 1] \\ 1, & x > 1. \end{cases}$$

Simularea unei probe Bernoulli revine la a simula un eveniment aleator de probabilitate constanta p . Un astfel de eveniment este de exemplu $\{U < p\}$, unde U este o variabila aleatoare uniforma pe $[0,1]$. Evenimentul contrar $\{U \geq p\}$ este un *esec*.

Vom construi programul Matlab pentru simularea lui X :

```
function x=bern(p)
u=rand;
if u>=p
    x=0;
else
    x=1;
end
end
```

Observatie. Daca aplicam de n ori aceasta functie putem spune ca am simulat o selectie de volum n asupra lui X .

```
>> for i=1:100
    x(i)=bern(0.1);
end
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE BINOMIALA

Repartitia binomiala se foloseste la probleme de tipul urmator: Intr-o urna se afla bile albe si negre in proportii date. Fie p probabilitatea ca la o extragere de bila din aceasta urna sa obtinem culoarea alba si $q = 1 - p$. Extragem din aceasta urna n bile, succesiv cate una, cu revenirea bilei extrasa, de fiecare data.

Atunci probabilitatea ca din n bile extrase k sa fie albe este

$$P_n(k) = C_n^k p^k q^{n-k},$$

adica este termenul general al dezvoltarii binomului $(p + q)^n$.

Fie variabila discreta $X \sim B(n, p)$, ce are repartitie binomiala

$$X : \left(\begin{array}{cccccc} 0 & 1 & \dots & k & \dots & n \\ q^n & C_n^1 p q^{n-1} & \dots & C_n^k p^k q^{n-k} & \dots & C_n^n p^n \end{array} \right), p, q \in (0, 1), p + q = 1.$$

Functia de repartitie a acesteia este:

$$F_X(x) = \begin{cases} 0, & x \leq 0 \\ q^n, & x \in (0, 1] \\ q^n + C_n^1 p q^{n-1}, & x \in (1, 2] \\ \vdots \\ q^n + C_n^1 p q^{n-1} + \dots + C_n^k p^k q^{n-k}, & x \in (k, k+1] \\ \vdots \\ 1, & x > n. \end{cases}$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = np \\ D^2(X) = npq. \end{cases} \quad (2)$$

Pentru n suficient de mare, conform teoremei limita centrala rezulta ca variabila

$$W = \frac{X - np}{\sqrt{npq}} \sim N(0, 1).$$

Pe baza acestei relatii vom construi programul Matlab de generare a lui $X \sim B(n, p)$.

```
function x=binom(n,p)
w=rand;
q=1-p;
x=round(n*p+w*sqrt(n*p*q));
end
>> x=binom(5000,1/4)
x =
    1271
```

Observatie. Variabila aleatoare discreta $X \in \mathbb{N}$ este o variabila binomiala $B(n, p), n \in \mathbb{N}^*, 0 < p < 1$ daca X reprezinta numarul de succese in n probe Bernoulli independente, adica

$$X = \sum_{i=1}^n Z_i,$$

unde Z_i sunt n variabile aleatoare Bernoulli, independente.

Pornind de la aceasta observatie putem simula variabila discreta X cu repartitie binomiala numarand succesele in n probe Bernoulli independente.

```
function x =bern2(n,p)
for h=1:10
```

6

```
for k=1:n
xx(k)=bern(p);
end
x(h)=sum(xx);
end
end
```

```
>> x=bern2(100,0.2)
```

```
x =
```

```
Columns 1 through 9
```

```
15 22 16 18 20 22 26 17 16
```

```
Column 10
```

```
17
```

Vom folosi formulele (2) pentru validarea algoritmului precedent.

```
>> mean(x)
```

```
ans =
```

```
18.9000
```

```
>> 100*0.2
```

```
ans =
```

```
20
```

```
>> std(x)
```

```
ans =
```

```
3.5103
```

```
>> sqrt(100*0.2*0.8)
```

```
ans =
```

```
4
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE POISSON

Repartitia Poisson este o repartitie asemanatoare cu cea binomiala, deosebindu-se de aceasta prin faptul ca numarul n de extrageri din urna este foarte mare iar probabilitatea p de

extragere a unei bile albe este foarte mica. Cu alte cuvinte, repartitia Poisson este un caz limita al repartitiei binomiale, pentru $n \rightarrow \infty$ si $p \rightarrow 0$, unde produsul $np = \lambda$ este constant.

Deoarece repartitia Poisson se foloseste in cazul aparitiei foarte rare a unui eveniment se mai numeste si repartitia evenimentelor rare.

Aceasta repartitie este frecvent intalnita in studiul unor fenomene din biologie, telecomunicatii, controlul statistic al calitatii (atunci cand probabilitatea obtinerii unui defect este foarte mica).

Probabilitatea ca din cele n bile extrase, k sa fie albe este

$$\begin{aligned} P(k) &= \lim_{n \rightarrow \infty} P_n(k) = \lim_{n \rightarrow \infty} C_n^k p^k q^{n-k} = \lim_{n \rightarrow \infty} C_n^k \left(\frac{\lambda}{n}\right)^k \left(1 - \frac{\lambda}{n}\right)^{n-k} = \\ &= \frac{\lambda^k}{k!} \cdot \underbrace{\lim_{n \rightarrow \infty} \frac{n!}{n^k (n-k)!}}_{=1} \cdot \underbrace{\lim_{n \rightarrow \infty} \left(1 - \frac{\lambda}{n}\right)^{n-k}}_{e^{-\lambda}} = \frac{\lambda^k}{k!} \cdot e^{-\lambda}, \quad \lambda > 0. \end{aligned}$$

Fie variabila discreta $X \sim P(\lambda)$, ce are repartitie Poisson

$$X : \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots \\ e^{-\lambda} & e^{-\lambda} \cdot \frac{\lambda}{1!} & e^{-\lambda} \cdot \frac{\lambda^2}{2!} & & e^{-\lambda} \cdot \frac{\lambda^k}{k!} & \dots \end{pmatrix},$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = \lambda \\ D^2(X) = \lambda. \end{cases}$$

Functia Matlab urmatoare simuleaza $X \sim P(\lambda)$ pornind de la o repartitie binomiala $B(n, p)$, pentru care se noteaza $\lambda = np$ si se face presupunerea ca $n \rightarrow \infty$ si $p \rightarrow 0$, λ ramanand constant.

```
function x=poisson(la,p)
```

```
n=round(la/p)
```

```
x=binom(n,p);
```

```
end
```

```
>> x=poisson(1,0.001)
```

```
n =
```

```
1000
```

```
x =
```

```
2
```

SIMULAREA UNEI VARIABLE DISCRETE CU REPARTITIE GEOMETRICA

Fie X o variabila aleatoare care semnifica numarul de esecuri pana la aparitia unui succes intr-un sir oarecare de probe Bernoulli independente.

Deci X are repartitia:

$$X : \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots & n \\ p & pq & pq^2 & & pq^k & & pq^n \end{pmatrix}, M(X) = \frac{q}{p}, \text{Var}(X) = \frac{q}{p^2}.$$

Funcția de repartitie a lui X este:

$$F(x) = P(X < x) = \sum_{k=0}^{x-1} pq^k = 1 - q^x, x = 0, 1, 2, \dots$$

adica este o functie de repartitie discreta.

Numele de repartitie geometrica provine din faptul ca $P(X = x) = pq^x$ reprezinta termenul unei progresii geometrice.

Programul Matlab urmator simuleaza X , pe baza unui algoritm care numara esecurile produse pana la realizarea unui succes intr-un sir de probe Bernoulli independente:

```
function x=rgeom(p)
```

```
x=0;
```

```
k=0;
```

```
while(k~=1)
```

```
    u=rand;
```

```
    if u<=p
```

```
        k=k+1;
```

```
    else
```

```
        x=x+1;
```

```
    end
```

```
end
```

```
end
```

```
>>x=rgeom(0.2)
```

```
x=
```

```
7
```

Observatie. Simularea variabilei aleatoare X , ce are repartitie geometrica se poate realiza si prin metoda inversa cu formula:

$$X = \left\lceil \frac{\log(U)}{\log(q)} \right\rceil,$$

unde :

- ✓ $[a]$ este partea intreaga a lui a ,
- ✓ U este o variabila aleatoare uniform repartizata in $[0,1]$.

Vom construi functia Matlab ce simuleaza $X \sim \text{Geom}(p)$ cu metoda inversa.

```
function x=rgeom2(p)
```

```
q=1-p;
```

```
u=rand;
```

```
x=round(log(u)/log(q));
```

```
end
```

```
>> x=rgeom2(0.3)
```

```
x =
```

```
2
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE HIPERGEOMETRICA

Vom considera experimentul cu urna descris in legatura cu repartitia binomiala, cu deosebire ca acum cele n bile se extrag din urna, ce contine a bile albe si b bile negre, fara revenirea bilei extrasa inaintea extragerii urmatoare.

Probabilitatea ca din cele n bile extrase, k sa fie albe este

$$P_n(k) = \frac{C_a^k C_b^{n-k}}{C_{a+b}^n}.$$

Daca $X \sim H(n, a, b)$, adica X are repartitie hipergeometrica de parametri n, a, b atunci

$$X : \left(\begin{array}{ccc} 0 & 1 & \dots & k \\ \frac{C_a^0 C_b^n}{C_{a+b}^n} & \frac{C_a^1 C_b^{n-1}}{C_{a+b}^n} & \dots & \frac{C_a^k C_b^{n-k}}{C_{a+b}^n} \end{array} \right),$$

pentru orice $\max(0, n-b) \leq k \leq \min(n, a)$.

Functia de repartitie a lui X este:

$$F(x) = \sum_{k=\max(0,n-b)}^{\min(n,a)} \frac{C_a^k C_b^{n-k}}{C_{a+b}^n}.$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = np \\ D^2(X) = np(1-p) \frac{a+b-n}{a+b-1}. \end{cases} \quad (3)$$

Se introduc urmatoarele notatii:

- $N_i = N_{i-1} - 1$, $(\forall) i = \overline{1, n}$ semnifica numarul bilelor din urna dupa extractia de rang i (unde $N_0 = a + b$),
- $p_i = \frac{N_{i-1} p_{i-1} - S}{N_{i-1} - 1}$, $(\forall) i = \overline{1, n}$ reprezinta probabilitatea de a extrage o bila alba la extragerea i , unde $S = 1$ daca la extractia $i - 1$ a fost gasita o bila alba si $S = 0$ in caz contrar.

Functia Matlab urmatoare permite simularea variabilei hipergeometrice X .

```
function x=hipergeom(n,a,b,p)
i=0;
x=0;
N=a+b;
while (i<n)
u=rand;
i=i+1;
if u<=p
S=1;
else
S=0;
end
x=x+S;
p=(N*p-S)/(N-1);
N=N-1;
end
end
```

Vom simula o selectie de volum 20 asupra lui X si vom folosi formulele (3) pentru validarea algoritmului precedent.

```
>>for k=1:20
```

```
    x(k)=hipergeom(10,15,13,0.9);
```

```
end
```

```
>> mean(x)
```

```
ans =
```

```
    8.9000
```

```
>> 10*0.9
```

```
ans =
```

```
    9
```

```
>> std(x)
```

```
ans =
```

```
    0.7182
```

```
>> 10*0.9*0.1*18/27
```

```
ans =
```

```
    0.6000
```

Laboratorul 5. Crearea de algoritmi proprii pentru generarea variabilelor aleatoare continue

Bibliografie:

1. I. Văduva. *Modele de simulare*, Editura Universitatii din București, 2004.
2. I. Vaduva, *Modele de simulare cu calculatorul*, Editura Tehnica, București, 1977.
3. I. Vladimirescu, *Probabilitati si statistica*, Note de curs, Facultatea de Matematica si Informatica, Universitatea din Craiova, an III, 1995-1996.
4. R. Trandafir, *Modele de simulare*, Note de curs, Facultatea de Hidrotehnica, an III, AIA, 2011-2012.
5. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Construirea in Matlab a unor functii pentru simularea unei variabile continue normale si a variabilelor inrudite.
- 2) Construirea unor functii in Matlab pentru simularea unei variabile continue cu o repartitie: exponentiala, Weibull, gamma.

SIMULAREA UNEI VARIABILE CU REPARTITIE NORMALA

O variabila aleatoare X are o repartitie unidimensionala $N(\mu, \sigma)$ daca admite o functie densitate de probabilitate de forma:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

unde

$$\begin{cases} M(X) = \mu \\ D^2(X) = \sigma^2. \end{cases}$$

Funcția de repartitie a variabilei aleatoare X nu are o forma analitica explicita, avand expresia urmatoare:

$$F_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt.$$

Simularea lui X se reduce la simularea variabilei normale redusa asociata acesteia:

$$Z = \frac{X - \mu}{\sigma} \sim N(0,1).$$

Relatia precedenta poate fi justificata in urmatoorul mod:

$$\begin{aligned} F_Z(z) &= P(Z < z) = P\left(\frac{X - \mu}{\sigma} < z\right) = P(X < \mu + z\sigma) = F_X(\mu + z\sigma) = \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\mu + z\sigma} e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \stackrel{w=(t-\mu)/\sigma}{=} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{w^2}{2}} dw = \Phi(z). \end{aligned}$$

Intre variabilele aleatoare X si Z exista relatia:

$$X = \mu + Z\sigma.$$

Vom construi in Matlab functia repnorm.m, pentru simularea lui X :

```
function [m1,s1]=repnorm(m,sig)
```

```
z=randn(100,1);
```

```
x=m+sig*z;
```

```
m1=mean(x);
```

```
s1=std(x);
```

```
end
```

```
>>[m,s]=repnorm(1,2)
```

```
m=
```

```
0.9155
```

```
s=
```

```
2.0831
```

Observatie. Daca aplicam functia predefinita in Matlab **$z=\text{randn}(n,1)$** putem spune ca am simulat o selectie de volum n asupra variabilei aleatoare $Z \sim N(0,1)$.

SIMULAREA UNEI VARIABILE ALEATOARE χ^2

Fie $Z_i, 1 \leq i \leq \gamma$ variabile normale $N(0,1)$ independente. O variabila aleatoare χ^2 cu γ grade de libertate este o variabila de forma:

$$\chi_\gamma^2 = \sum_{i=1}^{\gamma} Z_i^2, \gamma \in N^*. \quad (1)$$

O variabila aleatoare χ_γ^2 este continua si admite densitatea de probabilitate:

$$f(x) = \frac{1}{2^{\gamma/2} \Gamma\left(\frac{\gamma}{2}\right)} \cdot x^{\frac{\gamma}{2}-1} e^{-\frac{x}{2}}, x > 0,$$

unde

$$\Gamma(\gamma) = \int_0^{\infty} x^{\gamma-1} e^{-x} dx$$

semnifica functia Gamma a lui Euler, $\Gamma : (0, \infty) \rightarrow \mathfrak{R}$ si are proprietatile:

$$\begin{cases} \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \\ \Gamma(1) = 1 \\ \Gamma(a+1) = a\Gamma(a), (\forall) a > 0 \\ \Gamma(n+1) = n!, (\forall) n \in \mathbb{N} \end{cases}$$

iar

$$\begin{cases} M(\chi_{\gamma}^2) = \gamma \\ D^2(\chi_{\gamma}^2) = 2\gamma. \end{cases}$$

Pentru simularea in Matlab a unei variabile aleatoare χ_{γ}^2 vom folosi formula (1).

```
function x=hip(n)
z=randn(n,1);
x=sum(z.^2);
end
>> for i=1:200
x(i)=hip(10);
end
>> mean(x)
ans =
9.7490
>> std(x)^2
ans =
18.5520
```

SIMULAREA UNEI VARIABILE ALEATOARE STUDENT

Fie Z o variabila $N(0,1)$ si χ_{γ}^2 o variabila χ^2 cu γ grade de libertate. Presupunem ca Z si χ_{γ}^2 sunt independente. Numim variabila Student cu γ grade de libertate, variabila aleatoare

$$t_\gamma = \frac{Z}{\sqrt{\frac{\chi_\gamma^2}{\gamma}}}$$

Funcția densitatea de probabilitate a variabilei aleatoare t_γ este:

$$g(x) = \frac{1}{\sqrt{\pi\gamma}} \cdot \frac{\Gamma\left(\frac{\gamma+1}{2}\right)}{\Gamma\left(\frac{\gamma}{2}\right)} \left(1 + \frac{x^2}{\gamma}\right)^{-\frac{\gamma+1}{2}}$$

Vom construi funcția Matlab care ne permite simularea variabilei aleatoare t_γ .

```
function t = stud(n)
```

```
z=randn(1,1);
```

```
h=hip(n);
```

```
t=z/sqrt(h/n);
```

```
end
```

```
>>stud(4)
```

```
ans=
```

```
0.2128
```

SIMULAREA UNEI VARIABILE ALEATOARE SNEDECOR

Fie $\chi_{\gamma_1}^2$ și $\chi_{\gamma_2}^2$ două variabile aleatoare χ^2 independente. Numim variabila F a lui Snedecor cu (γ_1, γ_2) grade de libertate, variabila de forma:

$$F_{\gamma_1, \gamma_2} = \frac{\gamma_2}{\gamma_1} \cdot \frac{\chi_{\gamma_1}^2}{\chi_{\gamma_2}^2}$$

Densitatea de probabilitate a variabilei F este:

$$f(x) = \left(\frac{\gamma_1}{\gamma_2}\right)^{\frac{\gamma_1}{2}} \cdot \frac{\Gamma\left(\frac{\gamma_1 + \gamma_2}{2}\right)}{\Gamma\left(\frac{\gamma_1}{2}\right)\Gamma\left(\frac{\gamma_2}{2}\right)} \cdot x^{\frac{\gamma_1}{2}-1} \left(1 + \frac{\gamma_1}{\gamma_2}x\right)^{-\frac{\gamma_1 + \gamma_2}{2}},$$

iar

$$\begin{cases} M(F_{\gamma_1, \gamma_2}) = \frac{\gamma_2}{\gamma_2 - 2}, \gamma_2 > 2 \\ D^2(F_{\gamma_1, \gamma_2}) = \frac{2\gamma_2^2(\gamma_1 + \gamma_2 - 2)}{\gamma_1(\gamma_2 - 2)^2(\gamma_2 - 4)}, \gamma_2 \neq 2, \gamma_2 \neq 4. \end{cases}$$

In Matlab vom avea:

```
function [m,s] = snedec(n1,n2)
for i=1:100
h1=hip(n1);
h2=hip(n2);
F(i)=(n2/n1)*(h1/h2);
end
m=mean(F);
s=std(F)^2;
if n2>2
    mm=n2/(n2-2)
end
if n2~=2 & n2~=4
    ss=(2*n2^2*(n1+n2-2))/(n1*(n2-2)^2*(n2-4))
end
end
>> [m,s]=snedec(6,9)
mm =

    1.2857
ss =
    1.4327
m =
    1.3769
s =
    1.2140
```

SIMULAREA UNEI VARIABILE LOG- NORMALE

O variabila aleatoare X este o variabila log- normala daca $Y = \ln X$ este normala, adica are functia densitate de probabilitate:

$$f(y) = \frac{1}{\sigma_y \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{y - \mu_y}{\sigma_y} \right)^2},$$

unde :

$$\begin{cases} \mu_y = M(Y) = \ln \mu_x - \frac{1}{2} \ln \left[\frac{\sigma_x^2}{\mu_x^2} + 1 \right] \\ \sigma_y^2 = \ln \left[\frac{\sigma_x^2}{\mu_x^2} + 1 \right]. \end{cases}$$

Funcția Matlab urmatoare simuleaza o variabila log- normala.

```
function v=lognor(mx,sx)
spy=log(sx^2/mx^2+1);
my=log(mx)-1/2*log(sx^2/mx^2+1);
z=norm(0,1);
y=my+sqrt(spy)*z;
v=exp(y);
end
```

SIMULAREA UNEI VARIABILE CU REPARTITIE EXPONENTIALA

O variabila exponentiala $X \sim \text{Exp}(\lambda)$ are functia densitatea de probabilitate:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0 \end{cases}, \lambda \in \mathfrak{R},$$

functia de repartitie:

$$F_X(x) = \int_{-\infty}^x f(t) dt = \int_0^x f(t) dt = 1 - e^{-\lambda x}, \quad x > 0,$$

iar

$$\begin{cases} M(X) = \frac{1}{\lambda} \\ D^2(X) = \frac{1}{\lambda^2}. \end{cases}$$

Pentru simularea unei variabile aleatoare X , ce are repartitie exponentiala vom folosi *metoda inversa*, conform careia putem simula orice variabila aleatoare X daca cunoastem functia sa de repartitie F si putem calcula functia inversa F^{-1} .

Algoritmul pentru simularea variabilei aleatoare X consta in:

- generarea unei valori u uniform repartizate in $[0,1]$,
- determinarea lui $X = F^{-1}(u) = -\frac{1}{\lambda} \ln(1-u)$.

Funcția **exponential.m**, pe care am realizat-o in Matlab permite simularea lui X cu metoda inversa:

```
function x=exponential(la)
```

```
u=rand();
```

```
x=(-1/la)*log(1-u);
```

```
end
```

```
>>for i=1:10
```

```
x(i)=exponential(1);
```

```
end
```

```
>>mean(x)
```

```
ans=
```

```
0.9955
```

```
>>std(x)^2
```

```
ans=
```

```
1.0115
```

SIMULAREA UNEI VARIABILE ALEATOARE CU REPARTITIE WEIBULL

O variabila aleatoare inrudita cu variabila exponentiala este variabila Weibull (notata $W(\alpha, \lambda, \gamma)$), a carei densitate de probabilitate este :

$$f(x) = \begin{cases} \gamma \lambda (x - \alpha)^{\gamma-1} e^{-\lambda(x-\alpha)^\gamma}, & x > \alpha, \alpha \in \mathfrak{R}, \lambda, \gamma > 0. \\ 0, & x \leq \alpha \end{cases}$$

Daca $X \sim Exp(1)$, atunci variabila Weibull se genereaza cu formula:

$$W = \alpha + \left(\frac{X}{\lambda}\right)^{\frac{1}{\gamma}}.$$

Intr-adevar, avem:

$$P(W < w) = P\left(X < \lambda(w - \alpha)^{\gamma}\right) = \int_{-\infty}^{\lambda(w - \alpha)^{\gamma}} \lambda(x - \alpha)^{\gamma-1} e^{-\lambda(x - \alpha)^{\gamma}} dx = \int_{-\infty}^{u = \alpha + \left(\frac{t}{\lambda}\right)^{\frac{1}{\gamma}}} \gamma \lambda (w - \alpha)^{\gamma-1} e^{-\lambda(w - \alpha)^{\gamma}} du.$$

Variabila Weibull se utilizeaza in fiabilitate, ea reprezentand durata de exploatare fara caderi a unui echipament sau produs industrial.

SIMULAREA UNEI VARIABILE ALEATOARE CU REPARTITIE GAMMA

O variabila aleatoare X are repartitia $G(\alpha, \lambda, \gamma)$ daca are functia densitate de probabilitate

$$f(x) = \begin{cases} \frac{\lambda^{\gamma}}{\Gamma(\gamma)} (x - \alpha)^{\gamma-1} e^{-\lambda(x - \alpha)}, & x > \alpha \\ 0, & x \leq \alpha \end{cases}$$

unde: $\alpha \in \mathfrak{R}$, $\lambda, \gamma > 0$ sunt respectiv parametrii de locatie, de scala si de forma ai variabilei.

Se observa ca o variabila exponentiala este variabila gamma $G(0, \lambda, 1)$ iar χ_{γ}^2 este o variabila gamma $G\left(0, \frac{1}{2}, \frac{\gamma}{2}\right)$.

Daca $Y \sim G\left(\alpha, \lambda, \frac{\gamma}{2}\right)$ iar $Z \sim G\left(0, \frac{1}{2}, \frac{\gamma}{2}\right)$ atunci avem:

$$Y = \alpha + \frac{Z}{2\lambda}. \tag{2}$$

Relatia (2) poate fi justificata in felul urmator:

$$\begin{aligned}
F_Z(z) &= P(Z < z) = P(2\lambda(Y - \alpha) < z) = P\left(Y < \alpha + \frac{z}{2\lambda}\right) = F_Y\left(\alpha + \frac{z}{2\lambda}\right) = \\
&= \int_{-\infty}^{\alpha + \frac{z}{2\lambda}} \frac{\lambda^{\gamma/2}}{\Gamma\left(\frac{\gamma}{2}\right)} (t - \alpha)^{\gamma/2 - 1} e^{-\lambda(t - \alpha)} dt \stackrel{w=2\lambda(t-\alpha)}{=} \int_{-\infty}^z \frac{\lambda^{\gamma/2}}{\Gamma\left(\frac{\gamma}{2}\right)} \left(\frac{w}{2\lambda}\right)^{\gamma/2 - 1} e^{-\lambda \cdot w/2\lambda} \frac{dw}{2} \\
&= \int_{-\infty}^z \frac{\lambda^{\gamma/2}}{2^{\gamma/2} \Gamma\left(\frac{\gamma}{2}\right)} \cdot w^{\gamma/2 - 1} e^{-w/2} dw.
\end{aligned}$$

Pentru simularea in Matlab a unei variabile aleatoare Y , ce are repartitia $G\left(\alpha, \lambda, \frac{\gamma}{2}\right)$

vom proceda in felul urmat: se genereaza $Z = \chi_{\gamma}^2$; se determina Y , utilizand (2).

```

function y = gam(al, la, n)
z = hip(n);
y = al + z / (2 * la);
end

```

Laboratorul 6. AJUSTAREA MATEMATICĂ A DATELOR EXPERIMENTALE

Bibliografie:

1. G. Groza, *Analiza numerica*, Ed. Matrix Rom, București, 2005.
2. I. Toma, I. Iatan, *Analiză numerică. Curs, aplicații, algoritmi în pseudocod și programe de calcul*, Ed. Matrix Rom, București, 2005.
3. R. Trandafir, *Modele și algoritmi de optimizare*, Editura Agir, București, 2004.
4. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, București, 2009.

Scopuri:

- 1) Ajustarea clasică. Metoda celor mai mici pătrate
- 2) Ajustare pe rețele echidistante
- 3) Alegerea curbei de ajustare
- 4) Ajustarea datelor experimentale utilizând Matlab și CurveExpert

Dependența funcțională a unei mărimi fizice, reprezentată prin variabila y , de o altă mărime fizică, reprezentată prin variabila x , poate fi studiată și empiric, măsurându-l pe y pentru cât mai multe valori ale lui x . Se obțin astfel seturile numerice (x_k, y_k) , $k = \overline{1, N}$. Aceste rezultate se pot înscrie într-un tabel și/sau reprezenta sub forma unui grafic. Problema constă, în acest caz, în găsirea unei funcții – mai precis, a unei formule, care să descrie cât mai exact posibil rezultatele experimentale. Altfel spus, graficul funcției căutate poate să conțină sau nu punctele (x_k, y_k) , datorită zgomotului, dar trebuie să fie cât mai aproape de acestea. Cu cât este mai bogată informația pe care o avem despre corespondența $x \rightarrow y(x)$, cu atât este mai sigură eliminarea zgomotului. Vom exprima deci dependența funcțională $x \rightarrow y(x)$ sub forma

$$y = f(x, c_0, c_1, \dots, c_n), \quad (1)$$

unde c_k , $k = \overline{0, n}$, sunt parametri care trebuie determinați astfel încât graficul lui y să fie suficient de aproape de graficul rezultatelor experimentale.

unde σ_k și τ_k sunt definiți astfel:

$$\sigma_m = \sum_{k=1}^N x_k^m p_k, k = \overline{0, 2n}, \tau_m = \sum_{k=1}^N y_k x_k^m p_k, m = \overline{0, n}. \quad (7)$$

Observatie. Soluția sistemului (6) există, este unică și reprezintă minimum-ul expresiei (2).

Exemplul 1.

Presupunem că am realizat câteva teste asupra unor piese de probă, din oțel, obținând astfel următorul tabel (în unități relative):

tensiunea x_i	2	4	5	7	8
deplasarea y_i	4	8	11	14	18

Intenționăm să găsim expresia analitică (empirică), ce caracterizează acest experiment, care să ne permită să estimăm cele mai bune deplasări, corespunzătoare tensiunilor care nu apar în tabelul anterior, dar sunt cuprinse între 2 și 8 unități relative.

Soluție.

Dacă am aplica interpolarea pe rețeaua $x_0 = 2, x_1 = 4, x_2 = 5, x_3 = 7, x_4 = 8$, polinomul de interpolare obținut ar avea gradul 4. În afara efortului de calcul, prin interpolare nu am înlăturat zgomotul prezent la măsurători.

Din aceste motive, vom încerca să aplicăm metoda celor mai mici pătrate. La început vom reprezenta punctele (x_i, y_i) într-un sistem de coordonate xOy (vezi figura 1).

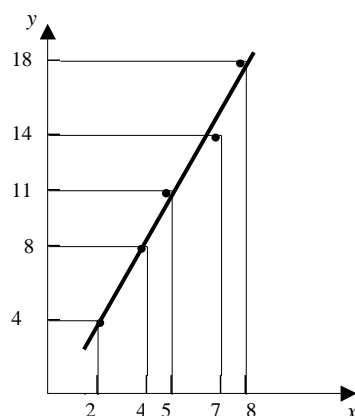


Figura 1. Reprezentarea grafică a punctelor (x_i, y_i) și dreapta de ajustare.

Se observă că punctele experimentale sunt distribuite în imediata vecinătate a unei drepte. Vom încerca să determinăm dreapta situată cel mai aproape de punctele graficului

anterior, impunând condiția ca distanța (2) dintre funcția deplasare $y(x)$ și dreapta $y = Ax + B$ să fie minimă, adică

$$\sum_{i=0}^4 (y_i - (Ax_i + B))^2 \equiv \varphi(A, B). \quad (8)$$

Soluție.

Sistemul (4) este în acest caz

$$\begin{cases} \frac{\partial \varphi}{\partial A} = -2 \sum_{i=0}^4 x_i (y_i - Ax_i - B) = 0 \\ \frac{\partial \varphi}{\partial B} = -2 \sum_{i=0}^4 (y_i - Ax_i - B) = 0. \end{cases} \quad (9)$$

După calcule simple, rezultă că avem de rezolvat următorul sistem algebric de două ecuații cu două necunoscute:

$$\begin{cases} A \sum_{i=0}^4 x_i^2 + B \sum_{i=0}^4 x_i = \sum_{i=0}^4 x_i y_i \\ A \sum_{i=0}^4 x_i + 5B = \sum_{i=0}^4 y_i. \end{cases} \quad (10)$$

Folosind notațiile (7), pentru $N = 5$, $p_j = 1$, $j = \overline{0, 4}$, sistemul anterior se scrie sub forma

$$\begin{cases} A\sigma_2 + B\sigma_1 = \tau_1 \\ A\sigma_1 + 5B = \tau_0 \end{cases} \quad (11)$$

și înlocuind x_i , y_i cu valorile din tabel, obținem sistemul

$$\begin{cases} 158A + 26B = 337 \\ 26A + 5B = 55, \end{cases} \quad (12)$$

a cărui soluție unică este $A = 2.237$, $B = -0.63$.

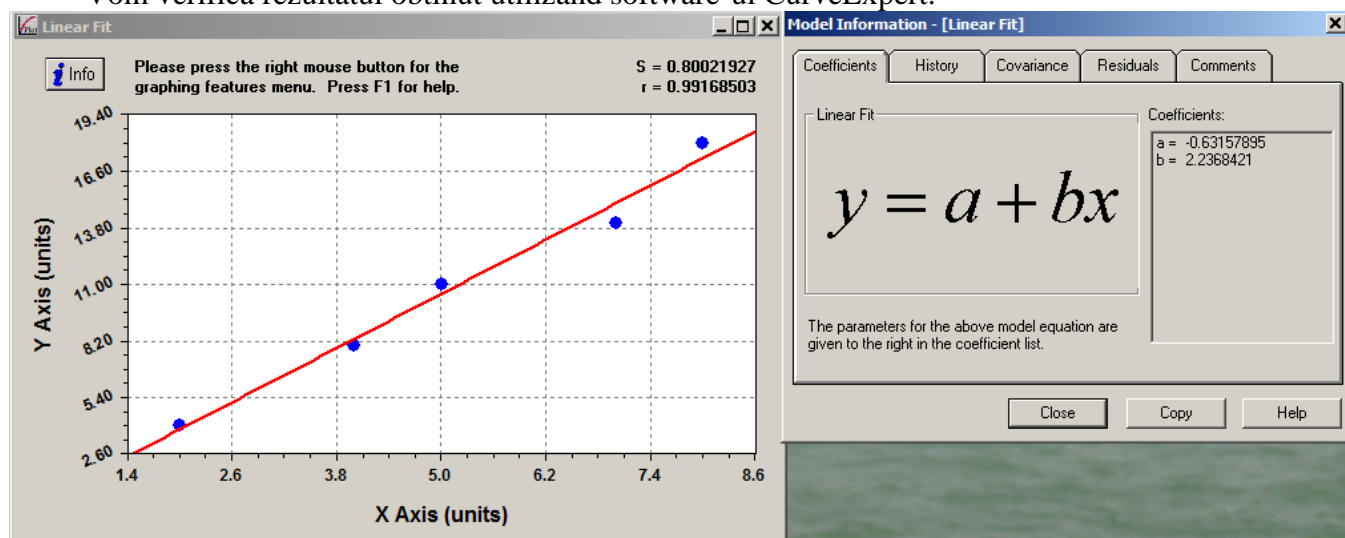
Ecuția dreptei căutate este $y = 2.237x - 0.63$. Dacă dorim să determinăm deplasarea pentru o tensiune de 3 unități relative, obținem cu aproximație $y = 2.237 \cdot 3 - 0.63 = 6.4$.

Secvența Matlab următoare ne permite ajustarea cu o dreapta (în sensul celor mai mici patrate) a setului de date.

```
>> x=[2 4 5 7 8];
>> y=[4 8 11 14 18];
>> coef=polyfit(x,y,1);
```

```
>> A=coef(1);
>> B=coef(2);
>> y1=A*x+B;
>> plot(x,y1,x,y,'o')
```

Funcția *polyfit* aproximează un set de date cu un polinom de grad n .
Vom verifica rezultatul obținut utilizând software-ul CurveExpert.



Exemplul 2. Să se ajusteze printr-o parabolă, de ecuație $y = ax^2 + bx + c$ datele tablei următoare:

x_i	-1	0	1	2
y_i	3	0	2	5

Soluție.

Coeficienții a , b , c se obțin din condiția ca funcția

$$E(a,b,c) = \sum_{i=1}^N \left(y_i - (ax_i^2 + bx_i + c) \right)^2$$

să admită un minimum, adică:

$$\begin{cases} \frac{\partial E}{\partial a} = 0, \\ \frac{\partial E}{\partial b} = 0, \\ \frac{\partial E}{\partial c} = 0. \end{cases}$$

Deoarece valorile x_i sunt echidistante, calculele pot fi simplificate prin schimbarea de variabilă $u = \frac{x - \bar{x}}{h}$.

Avem

$$h = 1; \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \Rightarrow \bar{x} = \frac{1}{2}; N = 4.$$

x_i	-1	0	1	2
u_i	$-\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{2}$
y_i	3	0	2	5

Etapa I. Determinăm parabola de ecuație $y = a_1 u^2 + b_1 u + c_1$.

Folosind notațiile $\sigma_k = \sum_{i=1}^4 u_i^k$ și $\tau_k = \sum_{i=1}^4 u_i^k \cdot y_i$, cum $\sigma_0 = 4$, sistemul din care găsim

coeficienții a_1 , b_1 și c_1 este:

$$\begin{cases} a_1 \sigma_4 + b_1 \sigma_3 + c_1 \sigma_2 = \tau_2, \\ a_1 \sigma_3 + b_1 \sigma_2 + c_1 \sigma_1 = \tau_1, \\ a_1 \sigma_2 + b_1 \sigma_1 + c_1 \sigma_0 = \tau_0. \end{cases}$$

1) Calculăm σ_k și τ_k , $k \geq 0$.

$$\sigma_1 = \sum_{i=1}^4 u_i = 0; \sigma_2 = \sum_{i=1}^4 u_i^2 = 2 \cdot \sum_{i=1}^2 u_i^2 = 2 \left(\frac{1}{4} + \frac{9}{4} \right) = 5;$$

$$\sigma_3 = \sum_{i=1}^4 u_i^3 = 0; \sigma_4 = \sum_{i=1}^4 u_i^4 = 2 \cdot \sum_{i=1}^2 u_i^4 = 2 \left(\frac{1}{16} + \frac{81}{16} \right);$$

$$\tau_0 = \sum_{i=1}^4 y_i = 10; \tau_2 = \sum_{i=1}^4 u_i^2 \cdot y_i = \frac{37}{2}; \tau_1 = \sum_{i=1}^4 u_i^1 \cdot y_i = 4.$$

Observăm că, în noua variabilă u , întotdeauna se anulează sumele σ de indice impar, iar cele de ordin par se calculează dublând rezultatul obținut pentru valorile pozitive ale lui u .

2) Se rezolvă sistemul anterior, care se decuplează, în virtutea observației anterioare:

$$\begin{cases} \frac{41}{4} a_1 + 5c_1 = \frac{37}{2}, \\ 5b_1 = 4, \\ 5a_1 + 4c_1 = 10. \end{cases}$$

Calculule se simplifică lucrând cu variabila u . Într-adevăr, în loc să rezolvăm un sistem de trei ecuații cu trei necunoscute, vom rezolva separat un sistem de numai 2 ecuații și 2 necunoscute. Rezultă $a_1 = 1.5$, $b_1 = 0.8$, $c_1 = 0.625$.

Etapa II. Revenim la variabila x și determinăm parabola de ecuație $y = ax^2 + bx + c$.

Avem

$$\begin{aligned} 1.5u^2 + 0.8u + 0.625 &= 1.5\left(x - \frac{1}{2}\right)^2 + 0.8\left(x - \frac{1}{2}\right) + 0.625 = \\ &= 1.5x^2 + -1.5x + \frac{1.5}{4} + 0.8x - 0.4 + 0.625. \end{aligned}$$

Rezultă parabola de ecuație: $y = 1.5x^2 - 0.7x + 0.6$, înfățișată în figura 2.

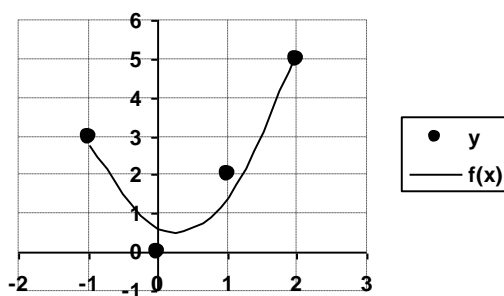


Figura 2. Ajustarea cu o parabolă

```
function r=sigma(x,n,N)
for m=1:2*n+1
r(m)=0;
for k=1:N
r(m)=r(m)+x(k)^(m-1);
end
end
end
```

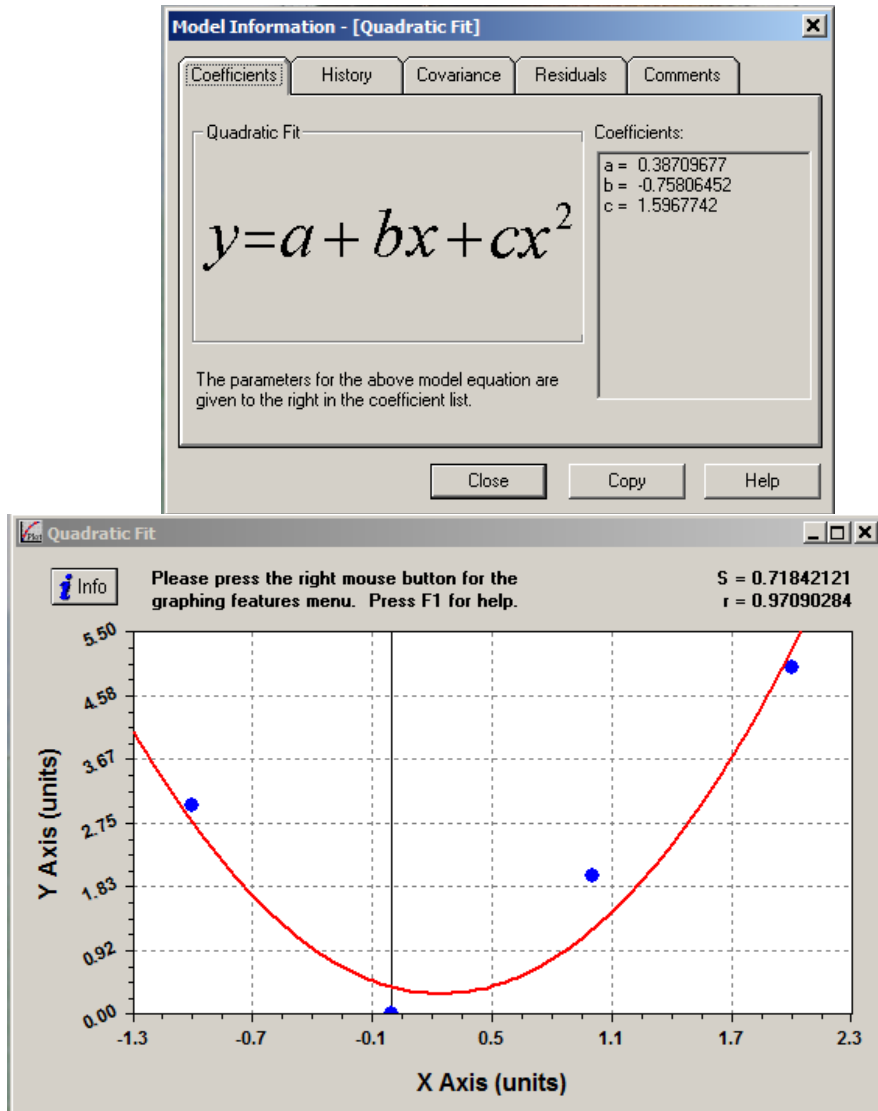
```
function r=tau(x,y,n,N)
for m=1:n+1
r(m)=0;
for k=1:N
r(m)=r(m)+y(k)*x(k)^(m-1);
end
end
end
```

```
>> x=-1:2;
>> u=x-mean(x);
>> y=[3 0 2 5];
>> N=4;
>> syms w a b c
>> r=sigma(u,2,N);
>> t=tau(u,y,2,N);
>> A=[r(3) r(2) r(1);r(4) r(3) r(2); r(5) r(4) r(3)];
>> B=[t(1) t(2) t(3)]';
```

8

```
>> q=linsolve(A,B);  
>> f=@(a,b,c,w) a*w.^2+b*w+c;  
>> plot(x,y,'db',x,f(q(1),q(2),q(3),x-mean(x)))
```

Utilizand CurveExpert vom obtine:



Problema alegerii curbei de ajustare este destul de subiectiva, depinzand de observatorul care decide asupra alurii graficului datelor experimentale. Se pune o problema fireasca: cum discernem intre doua sau mai multe forme care, la prima vedere, par la fel de potrivite pentru acelasi set de date?

Indicatorul de performanta este abaterea medie patratica, adica poate fi determinat cu

$$\text{formula } E = \sqrt{\frac{\sum_{i=1}^N (y_i - y(x_i))^2}{N - 1}},$$

in care $y_i, i=1, \overline{N}$ sunt datele, iar $y(x)$ este ecuația curbei de ajustare. Ecuația pentru care E este minim va reprezenta alegerea mai potrivită.

Exemplul 3. Să se ajusteze printr-o hiperbolă, de ecuație $y = \frac{a}{x} + b$ datele tabelului următor

x_i	7	8	9	10	11	12	13
y_i	7.4	8.4	9.1	9.4	9.5	9.5	9.4

Soluție.

Hiperbola de ajustare se obține din condiția ca funcția

$$E(a, b) = \sum_{i=1}^N \left(y_i - \left(\frac{a}{x_i} + b \right) \right)^2$$

să admită un minim, adică $\begin{cases} \frac{\partial E}{\partial a} = 0 \\ \frac{\partial E}{\partial b} = 0. \end{cases}$

Coefficienții a și b se obțin rezolvând sistemul:

$$\begin{cases} a \sum_{i=1}^7 \frac{1}{x_i^2} + b \sum_{i=1}^7 \frac{1}{x_i} = \sum_{i=1}^7 \frac{y_i}{x_i} \\ a \sum_{i=1}^7 \frac{1}{x_i} + 7b = \sum_{i=1}^7 y_i \end{cases} \Leftrightarrow \begin{cases} 0.08a + 0.73b = 6.437 & \cdot (-0.73) \\ 0.73a + 7b = 62.7 & \cdot 0.08 \end{cases}$$

$$\Leftrightarrow b = \frac{-0.73 \cdot 6.437 + 0.08 \cdot 62.7}{-0.73^2 + 7 \cdot 0.08} = 11.697$$

$$a = \frac{62.7 - 7 \cdot 11.697}{0.73} \Rightarrow a = -26.273$$

Ecuația hiperbolei este: $y = -\frac{26.273}{x} + 11.697$ și este ilustrată în figura 3.

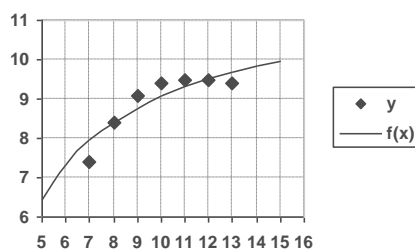


Figura 3. Ajustarea cu o hiperbolă.

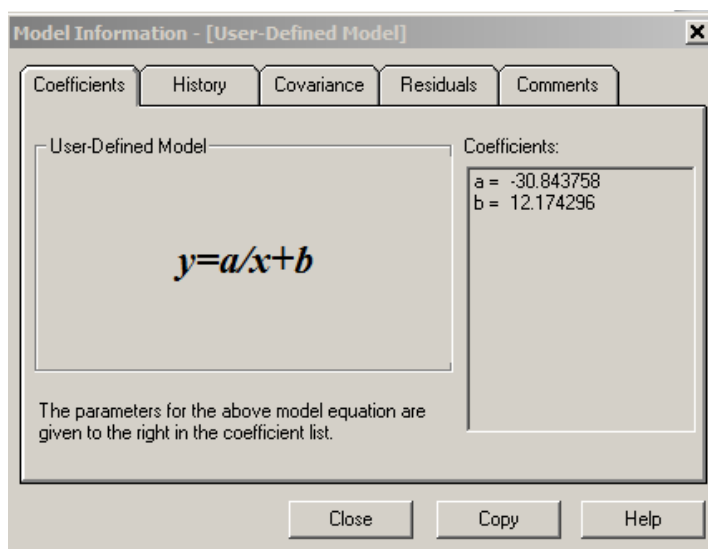
```

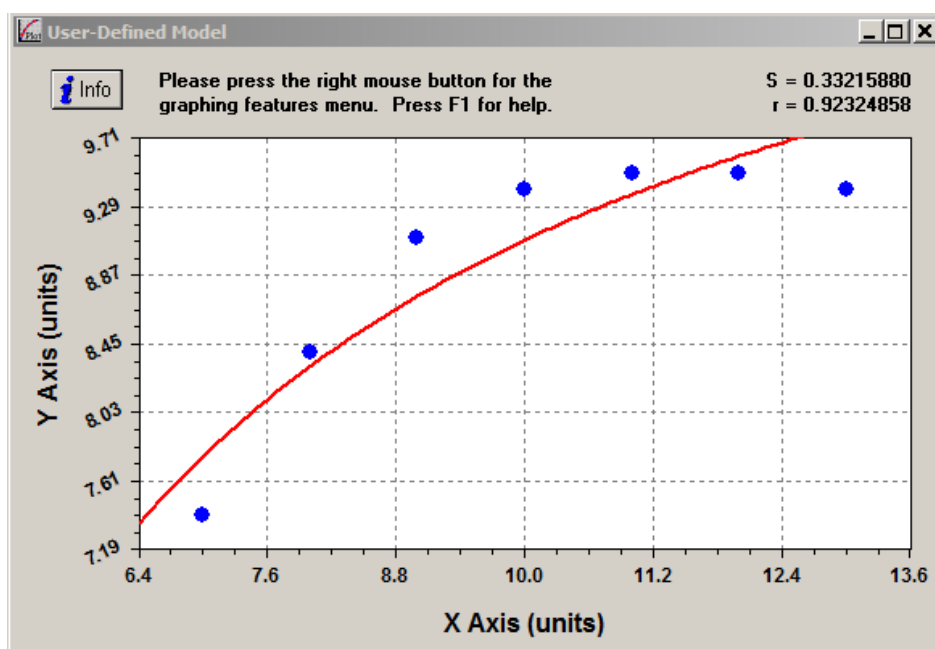
function r=sigma(x,n,N)
for m=1:n+2
r(m)=0;
for k=1:N
    r(m)=r(m)+x(k)^(-m+1);
end
end
end

function r=tau(x,y,n,N)
for m=1:2*n+1
r(m)=0;
for k=1:N
    r(m)=r(m)+y(k)/(x(k)^(m-1));
end
end
end
>>N=7;
>>syms w a b
>>x=7:13;
>>y=[7.4 8.4 9.1 9.4 9.5 9.5 9.4];
>>r=sigma(x,1,N);
>>t=tau(x,y,1,N);
>>A=[r(2) r(1);r(3) r(2)];
>>B=[t(1) t(2)]';
>>u=linsolve(A,B);
>>f=@(a,b,w) a./w+b ;
>>plot(x,y,'db',x,f(u(1),u(2),x))

```

Acelasi rezultat poate fi obtinut utilizand CurveExpert:





Exemplul 4. Să se ajusteze printr-o funcție de forma

$$y(x) = c_0 + c_1 \sin \pi x + c_2 \cos \pi x$$

datele tabelii următoare

x_i	0.2	0.4	0.6	0.8	1	1.2
y_i	1	2	3	5	5	3

Vom avea de rezolvat sistemul:

$$\begin{cases} \sum_{k=1}^6 (y_k - c_0 - c_1 \sin \pi x_k - c_2 \cos \pi x_k) = 0 \\ \sum_{k=1}^6 (y_k - c_0 - c_1 \sin \pi x_k - c_2 \cos \pi x_k) \sin \pi x_k = 0 \\ \sum_{k=1}^6 (y_k - c_0 - c_1 \sin \pi x_k - c_2 \cos \pi x_k) \cos \pi x_k = 0 \end{cases}$$

```
>> x=0.2:0.2:1.2;
>> y=[1 2 3 5 5 3];
>> tau=[sum(y),sum(y.*sin(pi*x)),sum(y.*cos(pi*x))] ;
>> sig(1,1)=6;
>> sig(1,2)=sum(sin(pi*x));
>> sig(1,3)=sum(cos(pi*x));
>> sig(2,2)=sum(sin(pi*x).^2);
>> sig(2,3)=sum(sin(pi*x).*cos(pi*x));
>> sig(3,3)=sum(cos(pi*x).^2);
>> sig(2,1)=sig(1,2);sig(3,1)=sig(1,3);sig(3,2)=sig(2,3);
>> linsolve(sig,tau')
ans =
```

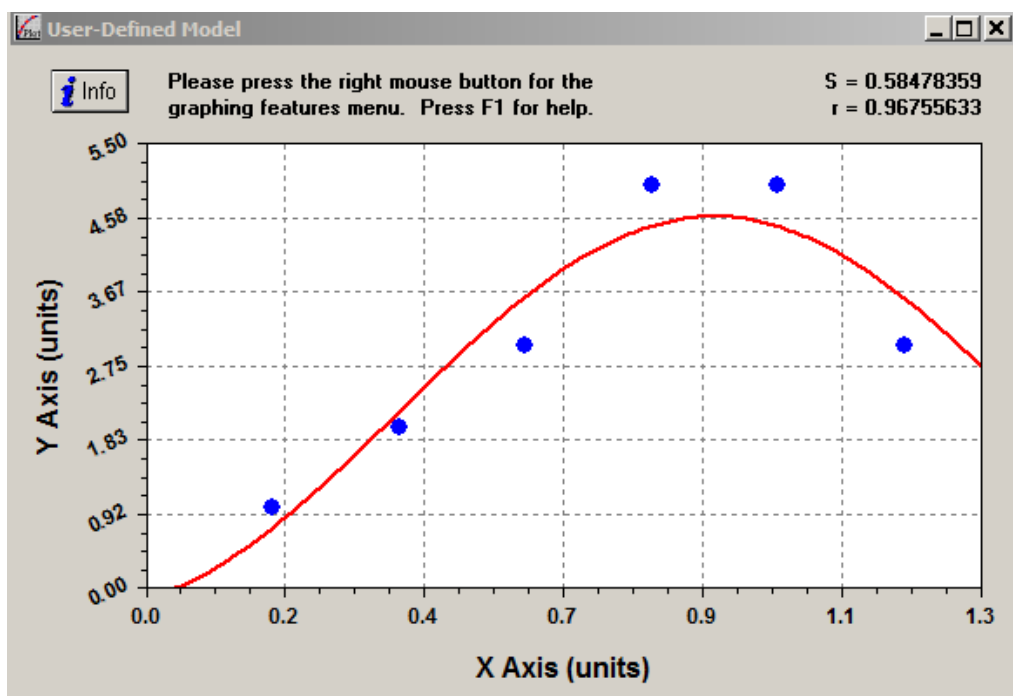
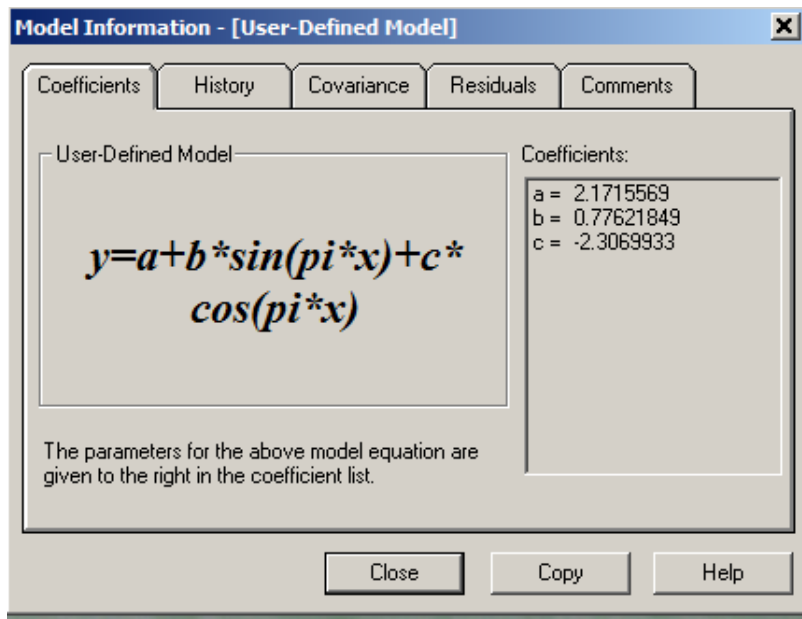
12

2.0660
0.8887
-2.4274

Deci,

$$y(x) = 2.0660 + 0.8887 \sin \pi x - 2.4274 \cos \pi x.$$

Folosind CurveExpert va rezulta:



Laboratorul 7. Validarea generatorilor

Bibliografie:

1. I. Văduva. *Modele de simulare*, Editura Universitatii din București, 2004.
2. I. Vaduva, *Modele de simulare cu calculatorul*, Editura Tehnica, București, 1977.
3. I. Vladimirescu, *Probabilitati si statistica*, Note de curs, Facultatea de Matematica si Informatica, Universitatea din Craiova, an III, 1995-1996.
4. R. Trandafir, *Modele de simulare*, Note de curs, Facultatea de Hidrotehnica, an III, AIA, 2011-2012.
5. M. Ghinea, V. Firețeanu, *Matlab: Calcul numeric- Grafică-Aplicații*, ed. Teora, București, 1998.

Scopuri:

- 1) Validarea in Matlab a algoritmilor pentru simuleaza unor variabile aleatoare continue.
- 2) Validarea in Matlab a algoritmilor pentru simuleaza unor variabile aleatoare discrete.

Validarea generatorilor se refera atat la verificarea corectitudinii formale a programelor cat si la verificarea ipotezei statistice de concordanta

$$H: X \sim F(x), \quad (1)$$

cu privire la functia de repartitie $F(x)$ a variabilei aleatoare X asupra careia s-a efectuat selectia simulata X_1, X_2, \dots, X_n de volum n suficient de mare.

Validarea generatorilor presupune urmatoarele doua etape:

- A. Construirea grafica a histogramei si compararea acesteia cu densitatea de probabilitate a lui X .
- B. Aplicarea testului de concordanta χ^2 pentru verificarea ipotezei (1).

Constructia histogramei se realizeaza utilizand urmatorul algoritim:

Pasul 1. Simulam un numar $n_1 \ll n$ de valori de selectie X_1, X_2, \dots, X_{n_1} , pe care le memoram.

Pasul 2. Alegem un numar k , care semnifica numarul intervalelor histogramei: I_1, I_2, \dots, I_k .

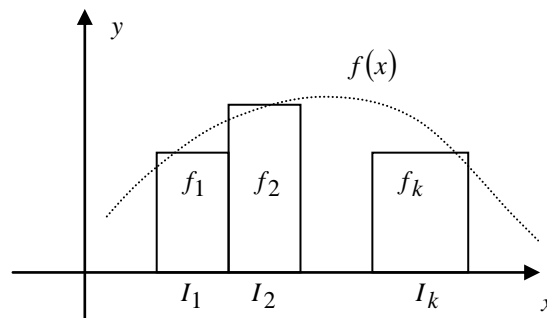
Pasul 3. Determinam pe baza selectiei urmatoarele limite ale intervalelor histogramei: $a_2 = \min\{X_1, X_2, \dots, X_{n_1}\}$ si $a_k = \max\{X_1, X_2, \dots, X_{n_1}\}$. Apoi formam intervalele: $I_i = (a_i, a_{i+1}]$, $(\forall) i = \overline{2, k-1}$ unde $a_i = a_2 + (i-2)h$, $(\forall) i = \overline{3, k-1}$.

Pasul 4. Determinam frecventele relative $f_i = n_i/n$, $(\forall) i = \overline{2, k-1}$, unde n_i reprezinta frecventele absolute, adica numarul valorilor de selectie, ce apartin intervalului I_i . Vom iface initializarile: $f_1 = f_k = 0$, $a_1 = a_2$, $a_{k+1} = a_k$.

Pasul 5. Simulam pe rand, celelalte $n - n_1$ valori de selectie si pentru fiecare X astfel simulat, vom realiza urmatoarele operatii:

- daca $X \leq a_2$ atunci setam: $a_1 = \min\{a(1), X\}$ si $f_1 = f_1 + 1$;
- daca $X > a_k$ atunci setam: $a_{k+1} = \max\{a(k+1), X\}$ si $f_k = f_k + 1$;
- daca $a_2 < X \leq a_k$ atunci setam: $p = \left[\frac{X - a_2}{h} \right] + 2$ si $f_{p+1} = f_{p+1} + 1$;

Pasul 6. Reprezentam grafic histograma selectiei X_1, X_2, \dots, X_n astfel: luam pe abscisa intervalele I_i si construim dreptunghiuri care au ca baza aceste intervale si ca inaltime, frecventele relative f_i .



Linia punctata sugereaza forma densitatii de probabilitate a variabilei X .

Observatie. Pentru o variabila aleatoare discreta X , ce ia valorile $a_1, a_2, a_3, \dots, a_m$ cu probabilitatile $p_1, p_2, p_3, \dots, p_m$ functia de probabilitate $f(x)$ se defineste prin:

$$f(x) = \begin{cases} p_i, & \text{daca } x = x_i, i = 1, 2, 3, \dots, m \\ 0, & \text{in rest} \end{cases}$$

iar functia de repartitie:

$$F_X(x) = P(X < x) = \begin{cases} 0, & x \leq a_1 \\ p_1, & x \in (a_1, a_2] \\ p_1 + p_2, & x \in (a_2, a_3] \\ \vdots & \\ p_1 + p_2 + \dots + p_k, & x \in (a_k, a_{k+1}] \\ \vdots & \\ 1, & x > a_m \end{cases}$$

Odata construita histograma putem aplica testul χ^2 pentru verificarea ipotezei (1).

Se calculeaza statistica $\chi^2 = \sum_{i=1}^k \frac{(n_i - np_i)^2}{np_i}$, care are o distributie χ^2 cu $k-1$ grade de

libertate (vezi teorema lui Karl Pearson) unde:

- k este numarul de intervale ale histogramei,
- n_i reprezinta frecventele absolute,
- $p_i, (\forall) i = \overline{1, k}$ constituie probabilitatile ca o observatie sa apartina intervalului

$I_i, (\forall) i = \overline{1, k}$ si sunt exprimate de:

$$\begin{cases} p_1 = P(a_1 < X \leq a_2) = F(a_2) \\ p_i = P(a_i < X \leq a_{i+1}) = F(a_{i+1}) - F(a_i), (\forall) i = \overline{2, k-1} \\ p_k = P(a_k < X < a_{k+1}) = 1 - F(a_k). \end{cases}$$

Ipoteza H se accepta daca $\chi^2 \leq \chi_{k-s-1, \alpha}^2$ si se respinge in caz contrar, α fiind probabilitatea erorii de genul I (se mai numeste nivel de semnificatie sau risc sau probabilitate de transgresiune) iar s semnificand numarul de parametri estimati.

VALIDAREA ALGORITMULUI PENTRU SIMULAREA UNEI VARIABILE ALEATOARE CU REPARTITIE NORMALA

Etapă I. Functia Matlab urmatoare permite simularea unei variabile aleatoare X, ce are o repartitie unidimensionala $N(\mu, \sigma)$, adica admite o functie densitate de probabilitate de forma:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (2)$$

function x=repnorm(m,sig,n)
z=randn(n,1);

```
x=m+sig*z;
end
```

Etapa II. Construim functia Matlab corespunzatoare functiei densitate de probabilitate din (2).

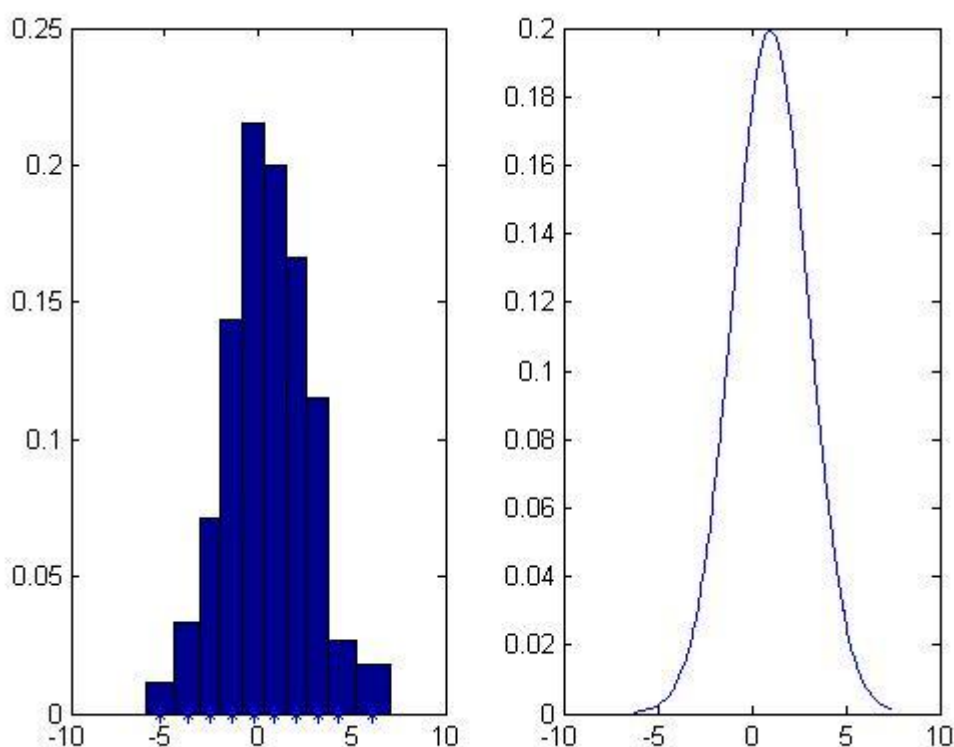
```
function f=dnorm(x,m,sig)
f=1/(sig*sqrt(2*pi))*exp(-(x-m).^2/(2*sig^2));
end
```

Etapa III. Construim functia Matlab, pe baza careia vom determina elementele histogramei.

```
function [x,a,ff,I]=normhist(n,n1,k)
f=zeros(1,k);
a=zeros(1,k+1);
x=repnorm(1,2,n1);
a(2)=min(x);
a(k)=max(x);
a(1)=a(2);
a(k+1)=a(k);
h=(a(k)-a(2))/(k-2);
for i=3:k-1
    a(i)=a(2)+(i-2)*h;
end
for i=3:k
    aa=a(i-1);bb=a(i);
    f(i-1)=length(find(x>aa& x<=bb));
end
for i=1:n-1
    xx=repnorm(1,2,1);
    x(n1+i)=xx;
    if xx<=a(2)
        a(1)=min(a(1),xx);
        f(1)=f(1)+1;
    elseif xx>a(k)
        a(k+1)=max(a(k+1),xx);
        f(k)=f(k)+1;
    else
        if a(2)<xx & xx<=a(k)
            p=round((xx-a(2))/h);
            f(p+1)=f(p+1)+1;
        end
    end
end
end
for i=1:k
    I(i)=(a(i+1)+a(i))/2;
end
ff=f/n;
end
```

Etapa IV. Scriem secventa Matlab utilizata pentru reprezentarea grafica a histogramei si respectiv a densitatii de probabilitate.

```
>>[x,a,f,I]=normhist(100,20,10)
>>subplot(1,2,1);
>>bar(I,f,'hist')
>>subplot(1,2,2);
>>x=sort(x);
>>plot(x,dnorm(x,1,2),'-b')
```



Etapa V. Scriem functia Matlab ce ne ajuta la definirea functiei de repartitie.

```
function r=frepnorm(z,m,sig)
syms t
r=(1/(sig*sqrt(2*pi)))*int(exp((-t-m)^2)/(2*sig^2)),t,-inf,z);
end
```

Etapa VI. Construim functia Matlab, ce ne permite aplicarea testului de concordanta χ^2 pentru verificarea ipotezei (1), pentru $\alpha = 0.005$.

```
function test_hist(n,k,m,sig)
[x,a,f,I]=normhist(n,20,k);
f=f*n;
p(1)=frepnorm(a(2),m,sig);
for i=2:k-1
    p(i)=eval(frepnorm(a(i+1),m,sig))-eval(frepnorm(a(i),m,sig));
```

```

end
p(k)=1-eval(frepnorm(a(k),m,sig));
eval(p)
hi_calc=sum(((f-n*eval(p)).^2)./(n*eval(p)))
if hi_calc<=20.28
    disp('se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica');
else
    disp('se respinge ipoteza ca repartitia empirica se aseamana cu cea teoretica');
end
end
>> test_hist(100,10,1,2)
hi_calc =
    10.7200
se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica

```

**VALIDAREA ALGORITMULUI PENTRU SIMULAREA UNEI VARIABILE ALEATOARE CU
REPARTITIE EXPONENTIALA**

O variabila exponentiala $X \sim Exp(\lambda)$ are functia densitatea de probabilitate:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0 \end{cases}, \lambda \in \mathfrak{R}$$

si functia de repartitie:

$$F_X(x) = \int_{-\infty}^x f(t) dt = \int_0^x f(t) dt = 1 - e^{-\lambda x}, \quad x > 0.$$

Vom rationa precum in cazul precedent, fiind necesare urmatoarele modificari:

Etapa I. Se defineste functia Matlab:

```

function x=exponential(la)
u=rand();
x=(-1/la)*log(1-u);
end

```

Etapa II. Construim functia Matlab:

```

function r=dexp(x,la)
if x>0
    r=la*exp(-la*x);
else
    r=0;
end
end

```

Etapa III. Functia Matlab normhist.m se modifica astfel:

Instructiunea

```

x=repnorm(1,2,n1);
se inlocuieste cu:
for i=1:n1
  x(i)=exponential(1);
end

```

iar instructiunea

```

xx=repnorm(1,2,1);
devine:
xx=exponential(1a);

```

Etapa IV. Instructiunea

```

>>plot(x,dnorm(x,1,2),'-b')

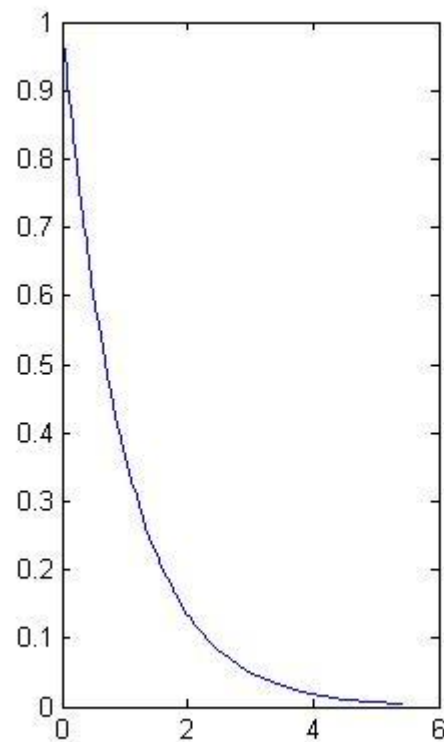
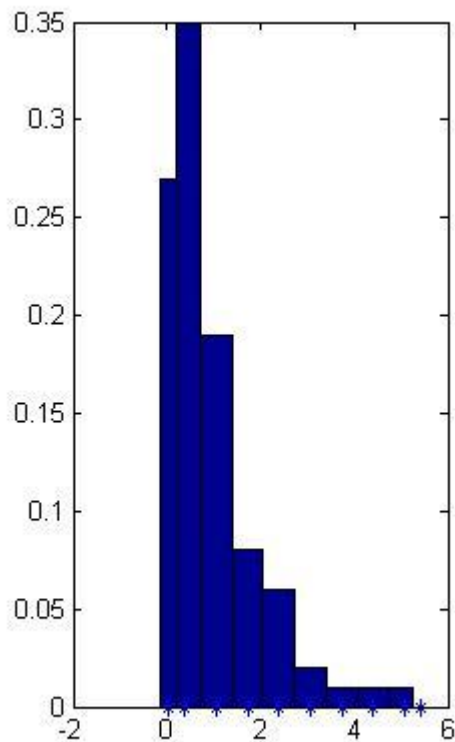
```

devine

```

plot(x,dexp(x,1),'-b')
si obtinem graficul

```



Etapa V. Vom avea functia:

```

function r=frepexp(x,la)
if x>0
r=1-exp(-la*x);
end
end

```

Etapa VI. Functia test_hist.m se modifica astfel:

```

function test_hist(n,k,la)
[x,a,f,I]=normhist(n,20,k);
f=f*n;
p(1)=frexp(a(2),la);
for i=2:k-1
    p(i)=frexp(a(i+1),la)-frexp(a(i),la);
end
p(k)=1-frexp(a(k),la);
p
hi_calc=sum(((f-n*p).^2)./(n*p))
if hi_calc<21.96
    disp('se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica');
else
    disp('se respinge ipoteza ca repartitia empirica se aseamana cu cea teoretica');
end
end
>> test_hist(100,10,1)
hi_calc =
    5.8196
se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica

```

**VALIDAREA ALGORITMULUI PENTRU SIMULAREA UNEI VARIABILE ALEATOARE CU
REPARTITIE GEOMETRICA**

Fie X o variabila aleatoare care semnifica numarul de esecuri pana la aparitia unui success intr-un sir oarecare de probe Bernoulli independente.

$$\text{Deci } X \text{ are repartitia: } X : \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots & n \\ p & pq & pq^2 & & pq^k & & pq^n \end{pmatrix},$$

Etapa I. Definim functia care simuleaza variabila X (vezi Lab. 4).

```

function x=rgeom2(p)
q=1-p;
u=rand;
x=round(log(u)/log(q));
end
>> x=rgeom2(0.3);

```

Etapa II. Construim functia Matlab:

```

function f=dgeom(x,t,xx,p)
q=1-p;
u=find(x(t)==xx);
if length(u)~=0
    f=p*q^(u-1);
else
    f=0;
end

```

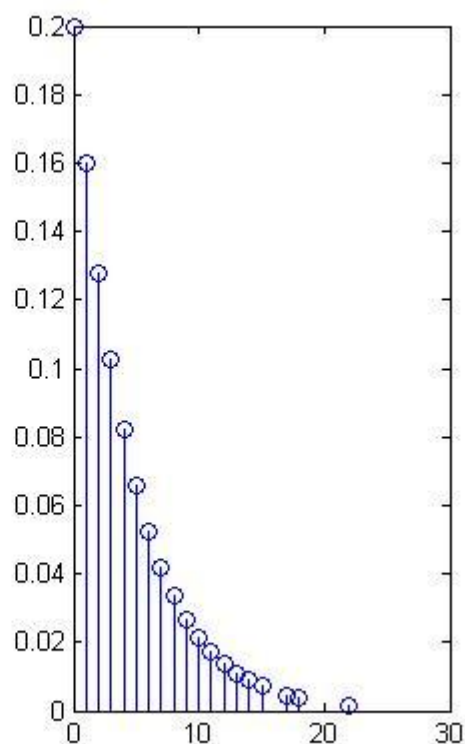
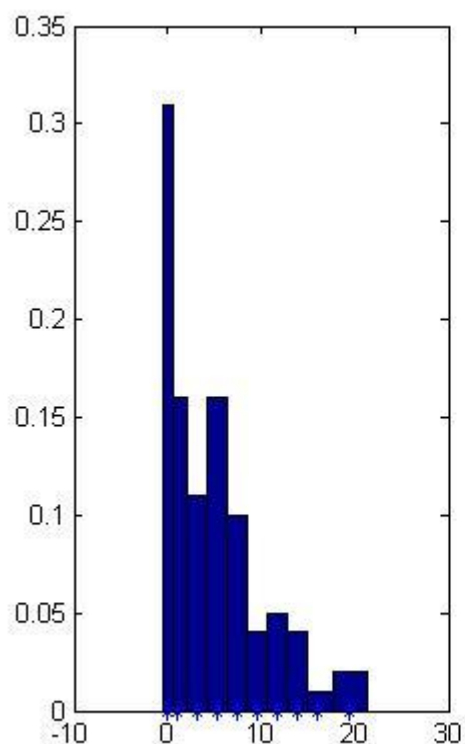
end

Etapa III. Functia Matlab normhist.m se modifica astfel:

```
for i=1:n1
    x(i)=rgeom2(0.2);
end
xx=rgeom2(0.2);
```

Etapa IV. Reprezentam histograma.

```
[x,a,f,I]=normhist(100,20,10)
subplot(1,2,1);
bar(I,f,'hist')
subplot(1,2,2);
xx=0:99;
for t=1:100
    fg(t)=dgeom(x,t,xx,0.2);
end
stem(x,fg)
```



Etapa V. Definim functiile:

```
function r=calcrep(x,t,xx,p)
n=length(xx);
s=frepgeom(xx,p);
if x(t)<=xx(1)
    r=0;
```

10

```
elseif x(t)>xx(n)
    r=1;
else
    for i=2:n
        if x(t)>xx(i-1)& x(t)<=xx(i)
            r=s(i);
        end
    end
end
end
```

```
function r=frepgeom(xx,p)
q=1-p;
s(1)=0;
n=length(xx);
for i=1:n-1
    s(i+1)=s(i)+p*q^(i-1);
end
s(n)=1;
r=s;
end
```

Etapa VI. Functia test_hist.m se modifica astfel:

```
function test_hist1(n,k,pp,xx)
[x,a,f,I]=normhist1(n,20,k);
f=f*n;
p(1)=calcrep(a,2,xx,pp);
for i=2:k-1
    p(i)=calcrep(a,i+1,xx,pp)-calcrep(a,i,xx,pp);
end
p(k)=1-calcrep(a,k,xx,pp);
p
hi_calc=sum(((f-n*p).^2)./(n*p))
if hi_calc<23.59
    disp('se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica');
else
    disp('se respinge ipoteza ca repartitia empirica se aseamana cu cea teoretica');
end
end
>>test_hist1(100,10,0.2,xx)
hi_calc =
    13.3864
se accepta ipoteza ca repartitia empirica se aseamana cu cea teoretica
```

Laborator 9. Elemente de fiabilitate

Bibliografie:

1. F. Baicu, *Elemente de fiabilitate*, ed. Victor, Bucuresti, 2005.
2. M. Ghinea, V. Fireșteanu, *Matlab: Calcul numeric- Grafică-Aplicații*, ed. Teora, București, 1998.
3. B. Gheorghe, *Modele de simulare cu aplicatii in fiabilitate*, ed Tehnica, București, 1992.
4. Al. Isaic- Maniu, V. Gh. Voda, *Fiabilitatea- sansa si risc*, ed. Tehnica, Bucuresti, 1986.
5. Gh. Mihoc, A. Muja, E. Diatcu, *Bazele matematice ale teoriei fiabilitatii*, ed. Dacia, Cluj- Napoca, 1976.
6. I. Vaduva, *Fiabilitatea programelor*, ed. Universitatii din Bucuresti, 2003.

Scopuri:

- 1) Insusirea notiunilor privind principalii indicatori de fiabilitate (marimi care exprima fiabilitatea unui sistem).
- 2) Estimarea fiabilității utilizând media timpului rămas de funcționare
- 3) Studiul fiabilitatii sistemelor pe baza schemelor logice de fiabilitate.

Exemplul 1. Un esantion de $n=10$ supape ale unor sisteme hidraulice speciale sunt supuse unei testari de fiabilitate, timpul fixat fiind $T=2000$ ore. Dupa epuizarea perioadei T au ramas in stare de buna functionare doar doua supape. In ipoteza unui model exponential, calculati:

- a) durata medie de functionare a elementelor considerate
- b) rata de defectare
- c) probabilitatea ca supapele sa functioneze fara defectiuni cel putin 500 ore.

Supapa	Timpul de buna functionare ai celor 8 supape
1	100
2	170
3	250
4	400
5	520
6	680
7	1200
8	1500

Rezolvare

O variabila exponentiala $X \sim Exp(\lambda)$ are functia densitatea de probabilitate:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0 \end{cases}, \lambda \in \mathfrak{R},$$

functia de repartitie:

$$F_X(x) = 1 - e^{-\lambda x}, \quad x > 0.$$

- a) Daca presupunem ca supunem unei incercari de fiabilitate n elemente de acelasi tip, inregistrand timpii de buna functionare t_1, \dots, t_n pana la caderea fiecarui element si fixam un timp de testare T_0 mai mic decat viata reala a produselor, in care cad $m \leq n$ elemente (nu este obligatoriu sa se defecteze toate cele n elemente supuse incercarii) atunci durata medie de functionare este data de formula:

$$M[T] = \frac{1}{m} \left[\sum_{i=1}^m t_i + (n-m)T_0 \right] = \frac{100 + 170 + \dots + 1500 + (10-8) \cdot 2000}{8} = \frac{8820}{8} = 1102.5 \cong 1103 \text{ ore}$$

- b) Rata de defectare $h(t)$ este probabilitatea de defectare în jurul unui moment dat t , conditionata de buna functionare a produsului pâna în acel moment:

$$h(t) = \frac{f(t)}{F(t)} = \frac{\lambda e^{-\lambda t}}{1 - F(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda = \frac{1}{M[T]} = \frac{8}{8820} \approx 0.0009 \text{ caderi/ ora}$$

- c) *Probabilitatea de supraviețuire* se notează $\bar{F}(t)$ sau $R(t)$ și se definește prin:

$$\bar{F}(t) = P(T > t) = 1 - F(t),$$

unde $F(t)$ și $f(t)$ sunt funcția de repartiție, respectiv, densitatea de repartiție a variabilei aleatoare T . $R(t)$ semnifica probabilitatea ca un sistem sau o componenta a acestuia sa fie in stare de functionare la momentul t sau probabilitatea sa functioneze fara sa se defecteze un timp mai mare decat t .

Deci:

$$\bar{F}(500) = P(T > 500) = 1 - F(500) = e^{-500\lambda} = e^{-500 \cdot 0.0009} = 0.64$$

Exemplul 2. S-au observat un numar de 74 autovehicule pe durata a 25000 km, cat reprezinta perioada de garantie. Datele, in ordinea producerii iesirilor din functionare se prezinta in tabelul urmator:

Autovehicul	t_i	Autovehicul	t_i
1	130	38	10410
2	200	39	11091
3	228	40	11300
4	412	41	11600
5	430	42	12010
6	540	43	12450
7	750	44	12450
8	750	45	12450
9	830	46	12600
10	965	47	12800
11	1000	48	13600
12	1434	49	14080
13	1635	50	15190
14	1981	51	15700
15	2040	52	15700
16	2040	53	15870
17	2280	54	15910
18	2400	55	15980
19	2830	56	16102
20	2901	57	16540
21	2901	58	16600
22	4380	59	17090
23	5140	60	17100
24	5400	61	17120
25	5611	62	17600
26	5900	63	17600
27	6012	64	19130
28	7200	65	19300
29	7200	66	19400
30	7510	67	19400
31	8200	68	21320
32	8430	69	22040
33	9000	70	22040
34	9060	71	22800
35	9111	72	23130
36	9613	73	23130
37	9710	74	23900

Sa se estimeze fiabilitatea acestui sistem utilizând media duratei rămase de funcționare.

Rezolvare.

Pasul 1. Determina o estimatie a funcției media timpului rămas de funcționare $L(t)$ evaluată la momentul $t = t_i$, pe baza datelor observate este dată de:

$$\hat{L}(t_i) = \hat{\mu}_{n-i} - t_i, \quad i = 0, 1, 2, \dots, (n-1).$$

function miu=TimpRamas(n,t,i)

s=0;

for j=i+2:n

 s=s+t(j);

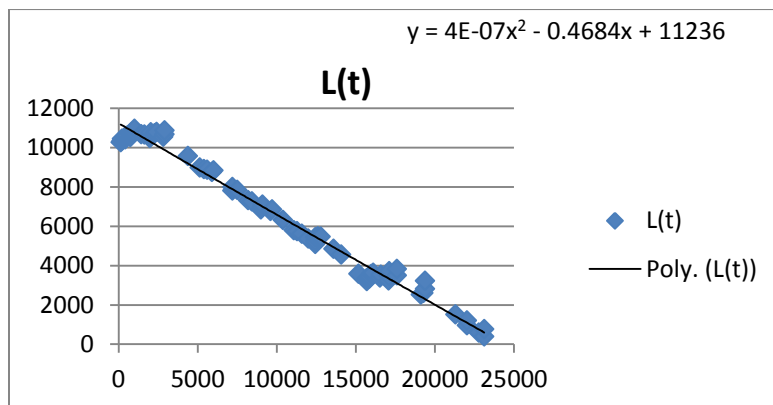
```

end
miu=s/(n-i-1);
end

fid=fopen('datta.txt','r')
t=fscanf(fid,'%f\t',[1,74]);
fclose(fid);
n=73;
for i=0:n-1
    miu(i+1)= TimpRamas(n+1,t,i); L(i+1)= miu(i+1)-t(i+1);
end

```

Pasul 2. Pentru perechile $(t_i, \hat{L}(t_i))$, $i = \overline{1, n-1}$, se determină expresia analitică a dependenței cu metoda celor mai mici pătrate (folosind Excel-ul)



Pasul 3. Se estimează fiabilitatea după $t^* = 2400$ km, cu relația $\bar{F}(t^*) = e^{-\int_0^{t^*} \frac{1+L'(x)}{L(x)} dx}$.

$$l(x) := 0.0000004x^2 - 0.4684x + 11236$$

$$h(x) := \frac{1 + \frac{d}{dx}l(x)}{l(x)}$$

$$e^{-\int_0^{2400} h(x) dx} = 0.887$$

Observatie. Probabilitatea de buna functionare a unui sistem este cu atat mai mare, cu cat durata de functionare este mai mica.

Exemplul 3. Fie un sistem serie alcatuit din 4 componente, pentru care ratele de defectare sunt: $\lambda_1 = 0.1 \cdot 10^{-6}$, $\lambda_2 = 0.2 \cdot 10^{-6}$, $\lambda_3 = \lambda_4 = 0.5 \cdot 10^{-6}$, distributia fiind considerata exponentiala. Calculati probabilitatea de buna functionare (deci fiabilitatea) a sistemului la momentul $t = 100000$.

Rezolvare

Sistemele de tip serie se caracterizeaza prin aceea ca defectarea unui element component determina iesirea din functiune a intregului sistem, adica sistemul funcționează dacă și numai dacă toate elementele sistemului funcționează.

Probabilitate de buna functionare a unui sistem serie se calculeaza conform formulei:

$$R_s(t) = R_1(t) \cdot R_2(t) \cdot \dots \cdot R_n(t) = \prod_{i=1}^n R_i(t).$$

Pentru cazul exponentialitatii, formula precedenta devine:

$$R_s(t) = \prod_{i=1}^n \lambda_i e^{-\lambda_i t} = e^{-\sum_{i=1}^n \lambda_i t} = e^{-1.3 \cdot 10^{-6} t} = 0.8781.$$

Exemplul 4. Se considera un sistem paralel, a carui prima componenta, cu fiabilitatea $R_1(t) = 0.9$ la defectare este inlocuita cu rezerva sa, a doua componenta, identica cu prima (deci cu fiabilitatea $R_2(t) = 0.9$).

Rezolvare

Sistemele de tip paralel se caracterizeaza prin aceea ca defectarea unui element component nu determina defectarea sistemului, intrucat la defectarea unei componente intra in functiune componenta legata in paralel cu cea defectata, numita componenta de rezerva. Doarece sistemul poate functiona cu cel putin un element conduce la un grad ridicat de fiabilitate.

Probabilitate de buna functionare a unui sistem paralel se calculeaza conform formulei:

$$R_p(t) = 1 - F_s(t) = 1 - \prod_{i=1}^n (1 - R_i(t)).$$

In cazul sistemului considerat fiabilitatea este:

$$R_p(t) = 1 - F_s(t) = 1 - (1 - 0.9)^2 \cdot 0.99.$$

Exemplul 5. Sa se scrie un program Matlab care sa analizeze comportarea unui sistem serie cu 4 componente, avand fiabilitatatile 0.7, 0.85, 0.9, 0.92.

```
N=4000; p1=0.7; p2=0.85; p3=0.9; p4=0.92; F1=rand(n,1); F2=rand(n,1);
F3=rand(n,1); F4=rand(n,1);
F=find(F1<=p1 & F2<=p2 & F3<=p3 & F4<=p4 );
N=length(F);
F=N/n;
```

Laborator 10. Serii de timp

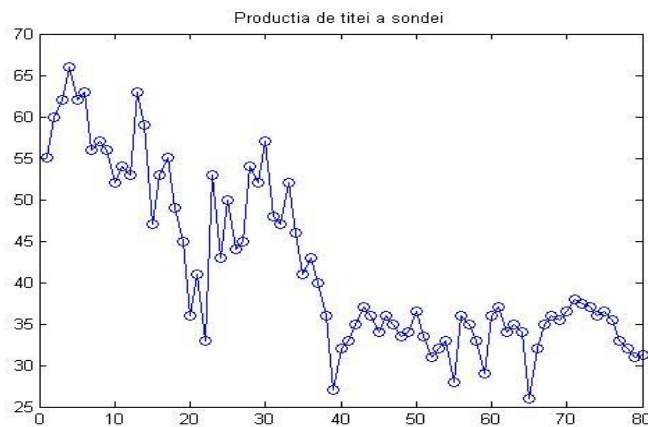
Bibliografie:

1. T. Popescu, *Serii de timp. Aplicatii in analiza sistemelor*, ed. Tehnica, Bucuresti, 2000.
2. M. Ghinea, V. Fireșteanu, *Matlab: Calcul numeric- Grafică-Aplicații*, ed. Teora, București, 1998.
3. P. J. Brockwell, R. A. Davis, *Time Series: Theory and Methods*, Springer-Verlag, 1987.
4. A Colojoara, *Serii de timp*, ed. Universitatii din Bucuresti, 2007.

Scopuri:

Realizarea unui exemplu practic de constructie a modelelor seriilor de timp utilizand metodologia de modelare Box-Jenkins

Exemplul . Se considera o serie de timp, care reprezinta valoarea productiei lunare de titei ale unei sonde, inregistrata intr-o anumita perioada de timp (reprezentata in figura).



Se cere:

- a) calculati media si dispersia corespunzatoare seriei de timp;
- b) reprezentati grafic functia functiile de autocorelatie si respectiv de autocorelatie partiala estimate ale seriei;
- c) modelati seria de timp respectiva;
- d) realizati predictia valorilor viitoare ale acestei serii.

Rezolvare

a) Analizand reprezentarea grafica a datelor seriei de timp considerata se poate observa:

- o tendinta descrescatoare, ce poate fi justificata prin exploatarea indelungata a zacamantului;
- aparitia unor variatii aproximativ periodice, in jurul tendintei, care marcheaza prezenta unui pattern sezonier in date. Factorii care influenteaza aparitia acestui pattern sunt: variatia vascozitatii titeiului si a capacitatii de extractie cu temperatura mediului.

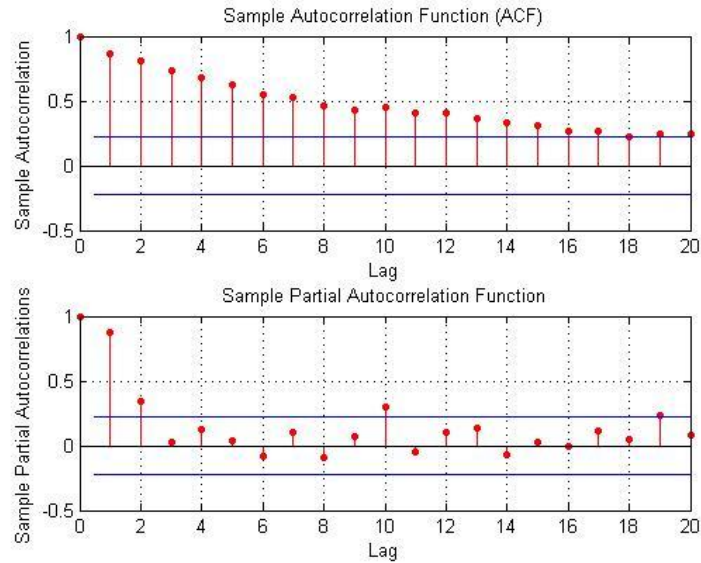
Se obtine: media=41.90375 iar dispersia=107.2626

b) Functia de autocorelatie estimata este:

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^n (x_t - \bar{x})^2}, k = 0, 1, 2, \dots$$

iar functia de autocorelatie partiala estimata se calculeaza folosind ecuatiile recursive:

$$\left\{ \begin{array}{l} \hat{\phi}_{11} = \hat{\rho}_1 \\ \hat{\phi}_{kk} = \frac{\hat{\rho}_k - \sum_{j=1}^{k-1} \hat{\phi}_{j,k-1} \cdot \hat{\rho}_{k-j}}{1 - \sum_{j=1}^{k-1} \hat{\phi}_{j,k-1} \cdot \hat{\rho}_j}, k = 2, 3, \dots \\ \hat{\phi}_{kj} = \hat{\phi}_{k-1,j} - \hat{\phi}_{kk} \hat{\phi}_{k-1,k-j}, j = 1, 2, \dots, k-1, k = 3, 4, \dots \end{array} \right.$$



Deoarece functia de autocorelatie estimata nu se amortizeaza tinzand catre 0, ea reflecta nestationaritatea (in medie) datelor seriei.

c) Tendinta inregistrata in cadrul seriei de timp si respectiv pattern-ul sezonier al datelor pot fi eliminate utilizand diferentierea.

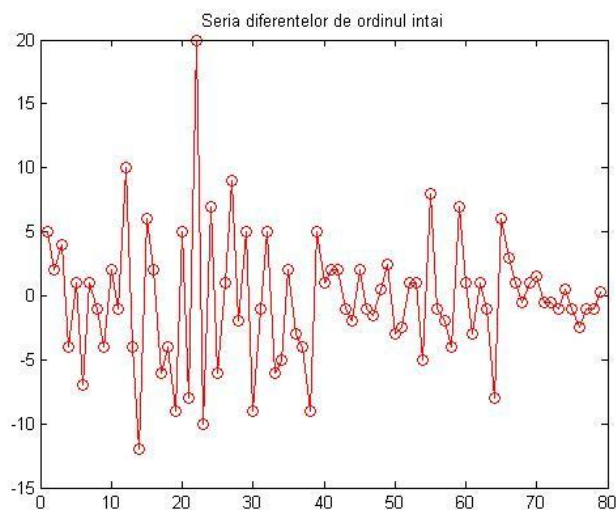
Seria rezultata in urma diferentierii de ordinul intai a datelor originale este:

$$\nabla z_t = z_t - z_{t-1} = (1 - B)z_t,$$

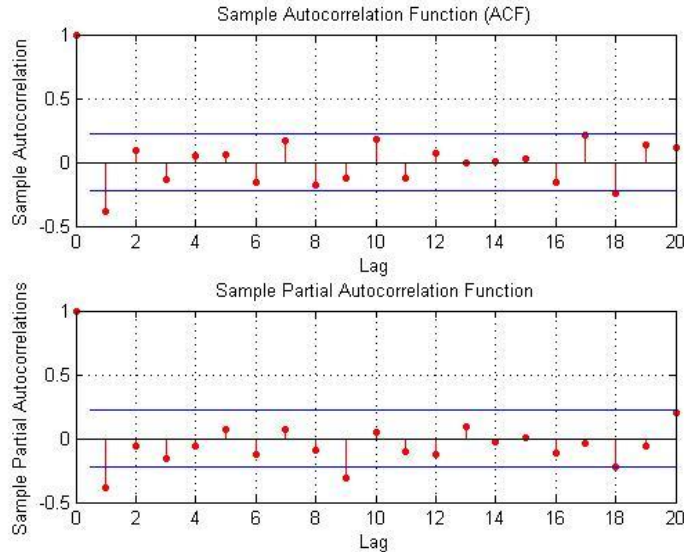
unde B este operatorul de intarziere:

$$Bz_t = z_{t-1}$$

si este reprezentata in fig. de mai jos:



Funcția de autocorelație estimată și respectiv funcția de autocorelație parțială estimată corespunzătoare seriei diferentelor de ordinul întâi se amortizează tinzând la zero:



Modelul seriei de timp considerată poate fi modelul ARIMA (auto regresiv integrat și de medie alunecătoare):

$$(1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p) \nabla z_t = a_t,$$

adică:

$$(1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_p B^p)(1 - B)z_t = a_t.$$

Deoarece valorile cele mai mari ale funcției de autocorelație estimată a seriei diferentelor corespund întârzierilor 1 și 18 rezultă modelul ARIMA:

$$(1 - \varphi_1 B - \varphi_{18} B^{18})(1 - B)z_t = a_t$$

deoarece trebuie amortizate aceste două valori.

d) Modelul ARIMA poate fi scris sub forma echivalentă:

$$z_t - \varphi_1 z_{t-1} - \varphi_{18} z_{t-18} = \frac{a_t}{1 - B};$$

sistemul Yule-Walker va deveni:

$$\begin{cases} \hat{\rho}_1 = \varphi_1 + \varphi_{18} \hat{\rho}_{17} \\ \hat{\rho}_{18} = \varphi_1 \hat{\rho}_{17} + \varphi_{18} \end{cases}$$

adică

$$\begin{cases} 1 = \varphi_1 + 0.2685 \cdot \varphi_{18} \\ 0.2661 = 0.2685 \cdot \hat{\rho}_{17} + \varphi_{18} \end{cases} \Rightarrow \begin{cases} \varphi_1 = 1.0007 \\ \varphi_{18} = -0.0026 \end{cases}$$

Deci, modelul ARIMA va fi:

$$(1 - 1.0007B + 0.0026B^{18})(1 - B)z_t = a_t$$

Testarea stationaritatii modelului pe baza functie de autocorelatie se realizeaza nu numai vizual ci si teoretic, utilizand testul Q (Box- Pierce), care presupune verificarea ipotezei:

$$H_0: \hat{\rho}_1 = \hat{\rho}_2 = \dots = \hat{\rho}_k = 0$$

pe baza statisticii:

$$Q(K) = n \sum_{k=1}^K \hat{\rho}_k^2 \approx \chi_K^2,$$

n fiind numarul datelor seriei disponibile dupa diferentiere.

Daca la un prag de semnificatie α :

$$Q(K) > \chi_{\alpha, K}^2$$

atunci ipoteza se respinge in sensul ca cel putin una dintre valorile coeficientilor de autocorelatie difera semnificativ de 0. Altfel, ipoteza se accepta iar seria este considerata stationara.

Programul Matlab asociat exemplului este urmatorul:

```
fid=fopen('datele.txt','r');
y=fscanf(fid,'%f\t',[1,80]);
fclose(fid);
n=80;
m=mean(y);v=var(y);
t=1:80;
figure(1);
plot(t,y,'-ob');
title('Productia de titei a sondei');
figure(2);
subplot(2,1,1);
autocorr(y,20);
```

```

subplot(2,1,2);
parcorr(y,20);
for i=2:n
yy(i-1)=y(i)-y(i-1);
end
figure(3)
t1=1:79;
figure(3);
plot(t1,yy,'-or');
title('Seria diferentelor de ordinul intai');
figure(4);
subplot(2,1,1);
autocorr(yy,20);
subplot(2,1,2);
parcorr(yy,20);
u=autocorr(yy,20)
u =
Columns 1 through 6

    1.0000   -0.3827    0.0966   -0.1399    0.0538    0.0596

Columns 7 through 12

   -0.1551    0.1641   -0.1751   -0.1253    0.1773   -0.1239

Columns 13 through 18

    0.0654   -0.0055    0.0081    0.0292   -0.1564    0.2157

Columns 19 through 21

   -0.2466    0.1375    0.1133
A=[1 u(17); u(17) 1];
b=[u(1) u(18)]';
phi=inv(A)*b

```

Laborator 11. Modelarea cu ajutorul rețelelor Petri

Bibliografie:

1. C. Girault, R. Valk, Petri Nets for Systems Engineering. A Guide to Modelling, Verification, and Applications, Springer-Verlag, 2001.
2. R. Trandafir, I. Iatan, *Modelare Simulare. Noțiuni teoretice și Aplicații*, format multimedia, 2012.

Scopuri: Realizarea unui exemplu practic de modelare a unei probleme practice cu ajutorul rețelelor Petri

Exemplul. În exemplul nostru, este pornita o cursă între două mașini A și B. Atunci când starterul primește semnul de „gata” de la toate mașinile, el dă semnalul de pornire, iar mașinile încep cursa.

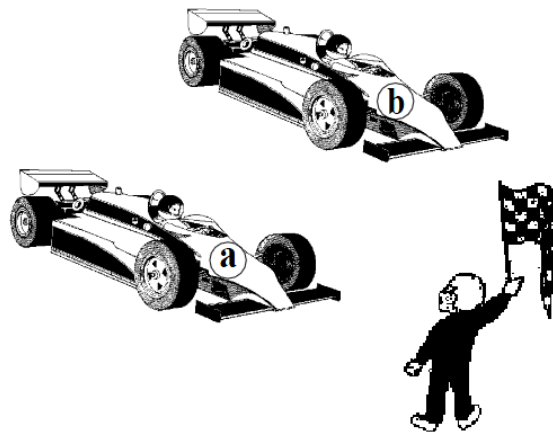


Fig. 1. Pornirea unei curse cu 2 mașini

Modelarea problemei

Alocarea locurilor:

- P_1 respectiv P_{10} - mașina A și respectiv B este pregătită pentru start;
- P_2 respectiv P_{11} - mașina A și respectiv B așteaptă începerea cursei;
- P_3 respectiv P_{12} - cursele corespunzătoare mașinilor A și respectiv B;
- P_4 respectiv P_8 - semnul de „gata” corespunzător mașinilor A și respectiv B;
- P_5 respectiv P_9 - semnul de start corespunzător mașinilor A și respectiv B;
- P_6 - starterul așteaptă semnul de „gata” (de la mașini);
- P_7 - semnul de start al starterului pentru începerea cursei.

Tranzițiile reprezintă trecerea dintr-o stare în alta:

- T_1 respectiv T_4 - transmiterea semnelor de „gata” către starter de către mașinile A și respectiv B;
- T_2 respectiv T_5 - începerea cursei de către mașinile A și B;
- T_3 - starterul dă semnul de începere a cursei.

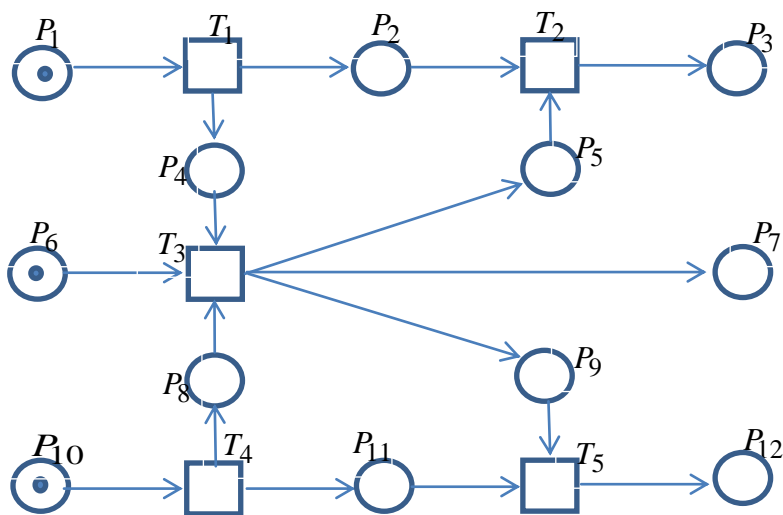


Fig.2. Reteaua Petri asociata exemplului considerat

În starea inițială se află câte un jeton în locurile P_1 , P_6 și P_{10} iar tranzițiile T_1, T_3, T_4 sunt activabile.

Vom presupune ca evaluarea fiecărui arc este 1. Prin execuția tranziției T_1 se iau jetoanele din P_1 și se pune unul în P_4 , iar celalalt în P_2 , în timp ce prin execuția tranziției T_4 se iau jetoanele din P_{10} și se pune unul în P_8 , iar celalalt în P_{11} (Fig. 3).

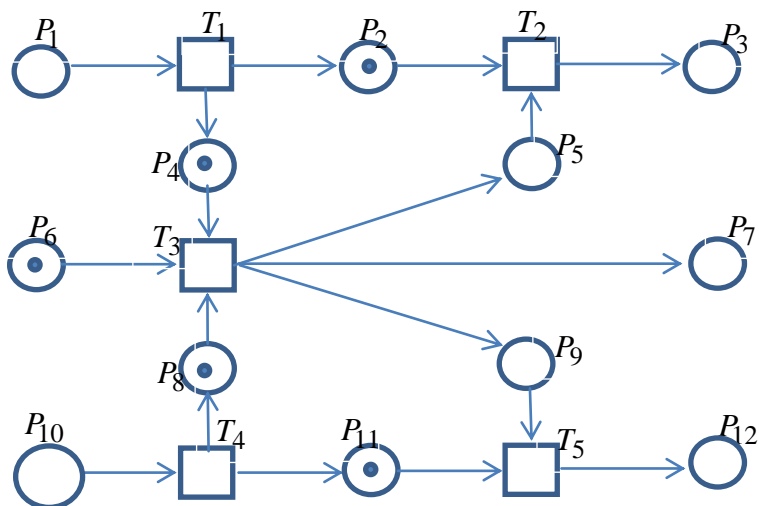


Fig. 3. Rezultatul execuției tranzițiilor T_1 și T_4

Atunci când se execută tranziția T_3 se ia jetonul din P_6 și se pune un jeton în P_7 , jetonul din P_4 se pune în P_5 , iar jetonul din P_8 se pune în P_9 (Fig. 4).

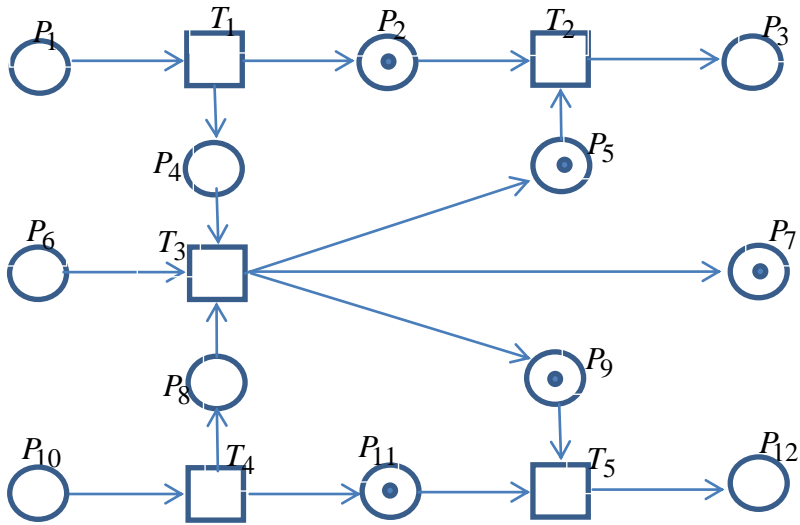


Fig. 4. Rezultatul executiei tranziției T_3

Executia tranziției T_2 determina scoaterea jetoanelor din P_2 si respectiv P_5 si punerea unui jeton in P_3 ; similar, executia tranziției T_5 determina scoaterea jetoanelor din P_9 si respectiv P_{11} si punerea unui jeton in P_{12} .

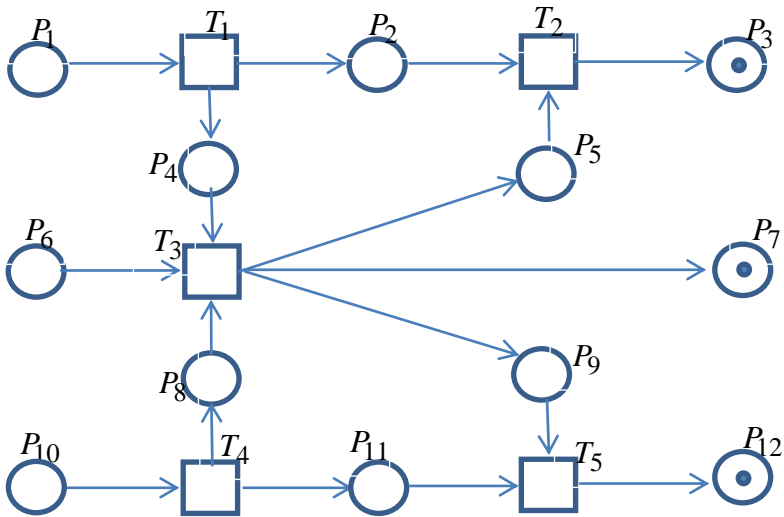


Fig. 5. Rezultatul executiei tranzițiilor T_2 si T_5

Exemplu prezentat poate fi rezolvat folosind Visual Simnet, in felul urmator:

Pasul 1. Proiectarea rețelei Petri (vezi Fig. 6)

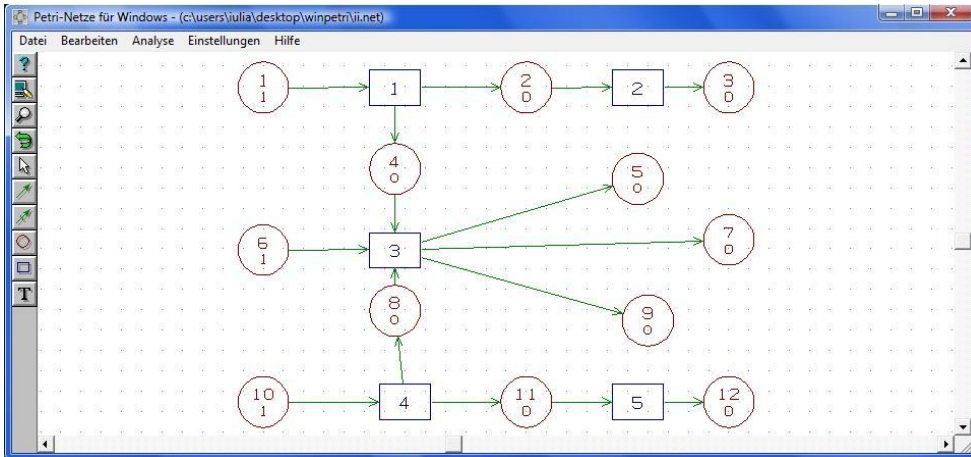


Fig. 6. Proiectarea rețelei Petri

Pasul 2. Simularea modelului pe care aceasta îl reprezintă (vezi Fig. 7)

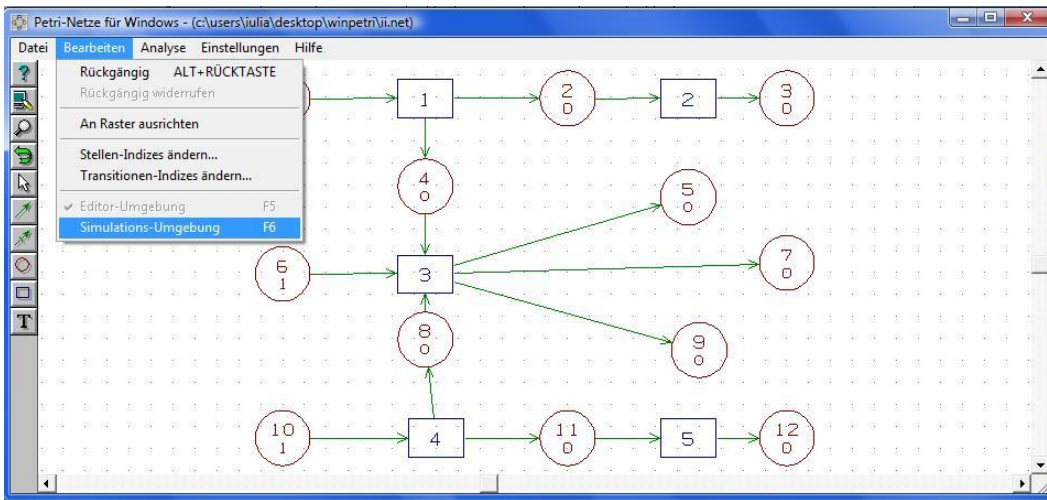


Fig. 7. Simularea modelului pe care îl reprezintă rețeaua Petri

In final rezulta:

